# R&S®RTP
# High-Performance Oscilloscope
# User Manual

1803648402
Version 03

ROHDE&SCHWARZ
Make ideas real

This manual describes the following R&S®RTP models with firmware version 5.30:

- R&S®RTP044B (1803.7000K04)
- R&S®RTP064B (1803.7000K06)
- R&S®RTP084B (1803.7000K08)
- R&S®RTP134B (1803.7000K13)
- R&S®RTP164B (1803.7000K16)

© 2023 Rohde & Schwarz GmbH & Co. KG
Muehldorfstr. 15, 81671 Muenchen, Germany
Phone: +49 89 41 29 - 0
Email: info@rohde-schwarz.com
Internet: www.rohde-schwarz.com
Subject to change – data without tolerance limits is not binding.
R&S® is a registered trademark of Rohde & Schwarz GmbH & Co. KG.
The terms HDMI, HDMI High-Definition Multimedia Interface, HDMI trade dress and the HDMI Logos are trademarks or registered trademarks of HDMI Licensing Administrator, Inc.
TEKTRONIX, TEK, and TEKPROBE are registered trademarks of Tektronix, Inc.
All other trademarks are the properties of their respective owners.

1803.6484.02 | Version 03 | R&S®RTP

Throughout this manual, products from Rohde & Schwarz are indicated without the ® symbol, e.g. R&S®RTP is indicated as R&S RTP.

# Contents


1 Safety and regulatory information....11
1.1 Safety instructions....11
1.2 Labels on the product....16
1.3 Warning messages in the documentation....16
1.4 Where to find key documents on Rohde & Schwarz....17
1.5 Korea certification class A....17

2 Preface....18
2.1 Key features....18
2.2 Documentation overview....18
2.3 Options described in this document....20
2.4 Conventions used in the documentation....22

3 Getting started....24
3.1 Preparing for use....24
3.2 Instrument tour....31
3.3 Operating the instrument....43

4 Instrument setup....70
4.1 System settings....70
4.2 Option settings....74
4.3 Appearance settings....77
4.4 Display settings....93
4.5 Frontpanel settings....97
4.6 Preset setup....101
4.7 Maintenance settings....103
4.8 Save / recall....107

5 Acquisition and waveform setup....112
5.1 Basics....112
5.2 Horizontal....122
5.3 Horizontal accuracy....125
5.4 Acquire settings....128
5.5 Vertical setup....140

5.6 Probes....146
5.7 R&S RT-ZVC probe....174
5.8 Differential signals....184
5.9 Setting up the waveform....188
6 Triggers....193
6.1 Basics of triggering....193
6.2 Setting up the trigger....194
6.3 Trigger types....196
6.4 Mode / Holdoff....225
6.5 Conditioning....228
6.6 Actions on trigger....229
6.7 Qualification....231
6.8 Sequence....234
6.9 Zone trigger....237
6.10 High speed serial pattern trigger (option R&S RTP-K140/K141)....241
6.11 External trigger input....256
7 Waveform analysis....261
7.1 Zoom....261
7.2 Reference waveforms....271
7.3 Mathematics....279
7.4 History....298
7.5 XY-diagram....304
8 Measurements....308
8.1 Cursor measurements....308
8.2 Automatic measurements....319
8.3 Quick measurements....396
9 Spectrum analysis....399
9.1 FFT analysis....399
9.2 Spectrum analysis (option R&S RTP-K37)....419
10 Mask testing....423
10.1 About mask testing....423
10.2 Mask test settings....425

10.3 Mask testing on eye diagrams....437
10.4 Working with masks....441

11 Search functions....449
11.1 Overview: search definition and results....449
11.2 Search setup....450
11.3 Search gate....465
11.4 Search display presentation....468

12 Data and file management....473
12.1 Save and recall user settings....473
12.2 Save and recall waveform data and results....482
12.3 Screenshots....507
12.4 Reports....511
12.5 File selection dialog....513

13 Protocol analysis....515
13.1 Basics of protocol analysis....515
13.2 I²C (option R&S RTP-K1)....525
13.3 SPI bus (option R&S RTP-K1)....539
13.4 UART/RS-232/RS-422/RS-485 (option R&S RTP-K2)....550
13.5 CAN and CAN FD (option R&S RTP-K3/K9)....559
13.6 LIN (option R&S RTP-K3)....585
13.7 MIL-1553 (option R&S RTP-K6)....599
13.8 ARINC 429 (option R&S RTP-K7)....616
13.9 Ethernet 10BASE-T and 100BASE-TX (option R&S RTP-K8)....629
13.10 Ethernet 100BASE-T1 (option R&S RTP-57)....644
13.11 Ethernet 1000BASE-T1 (option R&S RTP-K58)....663
13.12 RFFE (option R&S RTP-K40)....681
13.13 D-PHY (option R&S RTP-K42)....702
13.14 M-PHY and USB SSIC (option R&S RTP-K44 and K64)....718
13.15 Custom: Manchester / NRZ (option R&S RTP-K50)....748
13.16 8b/10b (option R&S RTP-K52)....781
13.17 MDIO (option R&S RTP-K55)....794
13.18 USB (option R&S RTP-K60)....807

13.19 USB 3.1 (option R&S RTP-K61/K62)....834
13.20 USBPD (option R&S RTP-K63)....856
13.21 SpaceWire (option R&S RTP-K65)....869
13.22 PCIe (option R&S RTP-K72/K73)....881
13.23 DDR (option R&S RTP-K91/K93 )....916

14 Mixed signal option (MSO, R&S RTP-B1)....938
14.1 Digital channels and parallel buses....938
14.2 Display....947
14.3 Trigger....949
14.4 Measurements on digital channels....962
14.5 Data export....963
14.6 Mathematics....964
14.7 Search....964

15 Waveform generator (option R&S RTP-B6)....966
15.1 Setup of the waveform generator....966
15.2 Setup of the pattern generator....985
15.3 Output....988
15.4 Coupling settings....991
15.5 Sync settings....992
15.6 Configuring the waveform generator....993

16 Pulse source and TDR/TDT analysis (options R&S RTP-B7/K130)....997
16.1 Pulse source (option R&S RTP-B7)....997
16.2 TDR/TDT analysis (option R&S RTP-K130)....999

17 I/Q software interface (option R&S RTP-K11)....1014
17.1 I/Q mode....1014
17.2 I/Q sampling setup....1014
17.3 I/Q data output....1020
17.4 Deembedding in I/Q mode....1021

18 Jitter analysis and clock data recovery....1023
18.1 Jitter measurements (option R&S RTP-K12)....1024
18.2 Advanced jitter and noise (option R&S RTP-K133/K134)....1033

18.3 Clock data recovery....1067

19 Advanced eye diagram analysis (option R&S RTP-K136/K137).. 1072

19.1 Calculation and display of advanced eye diagrams....1072

19.2 Eye diagram setup....1073

19.3 Measurements and zoom on eye diagrams....1085

19.4 Mask Testing on Eye Diagrams....1086

20 Deembedding (option R&S RTP-K121/-K122)....1087

20.1 Configuring the measurement setup for deembedding....1087

20.2 Deembedding settings....1088

20.3 Responses....1097

20.4 Proven probe, proven cable....1098

21 External frontend control (option R&S RTP-K553)....1103

21.1 External frontend basics....1103

21.2 Test setups....1104

21.3 Settings for frontend control....1107

21.4 Using frontends to get I/Q data....1114

21.5 Using frontends to get raw data....1115

21.6 Troubleshooting frontend control....1117

22 Compliance tests....1121

22.1 USB compliance test....1122

22.2 Ethernet compliance tests (options R&S RTP-K22/K23/K25/K86)....1124

22.3 D-PHY compliance tests (option R&S RTP-K26)....1126

22.4 eMMC compliance tests (option R&S RTP-K92)....1127

22.5 PCIe compliance tests (option R&S RTP-K81)....1128

22.6 DDR3 compliance tests (option R&S RTP-K91)....1129

22.7 DDR4 compliance tests (option R&S RTP-K93)....1130

22.8 C-PHY compliance tests (option R&S RTP-K28)....1130

23 Network operation and remote control....1132

23.1 Operating system....1132

23.2 Setting up a network (LAN) connection....1137

23.3 Web interface....1142

23.4 Remote Desktop connection....1147

23.5 Remote control....1149

24 Remote control commands....1165

24.1 Conventions used in remote command description....1171

24.2 Finding the appropriate command....1172

24.3 Programming examples....1172

24.4 Frequently used parameters and suffixes....1188

24.5 Common commands....1196

24.6 General remote settings....1201

24.7 Instrument setup....1205

24.8 Acquisition and setup....1234

24.9 Trigger....1304

24.10 Waveform analysis....1378

24.11 Cursor measurements....1413

24.12 Automatic measurements....1426

24.13 Spectrum analysis....1524

24.14 Mask testing....1542

24.15 Search....1565

24.16 Data management....1604

24.17 Protocols....1636

24.18 Mixed signal option (MSO, R&S RTP-B1)....2309

24.19 Waveform generator (option R&S RTP-B6)....2334

24.20 Pulse source and TDR/TDT analysis (options R&S RTP-B7/K130)....2357

24.21 I/Q software interface (option R&S RTP-K11)....2366

24.22 Jitter analysis and clock data recovery....2373

24.23 Eye diagram analysis....2412

24.24 Deembedding (option R&S RTP-K121/-K122)....2432

24.25 External frontend control (option R&S RTP-K553)....2444

24.26 Maintenance....2460

24.27 Status reporting....2463

24.28 Remote trace....2469

24.29 Deprecated commands....2473

25 Maintenance and support....2475

25.1 Cleaning....2475

25.2 Power cables for R&S RTP....2475
25.3 Contacting customer support....2476
25.4 Information for technical support....2476
25.5 Data security....2477
25.6 Transporting....2477
25.7 Storage....2477
25.8 Disposal....2478
List of commands....2479
Index....2552

# 1 Safety and regulatory information

The product documentation helps you to use the product safely and efficiently. Follow the instructions provided here and in the Chapter 1.1, "Safety instructions", on page 11.

**Intended use**

The R&S RTP oscilloscope is designed for measurements on circuits that are only indirectly connected to the mains or not connected at all. It is not rated for any measurement category.

The product is intended for the development, production and verification of electronic components and devices in industrial, administrative, and laboratory environments. Use the product only for its designated purpose. Observe the operating conditions and performance limits stated in the data sheet.

**Where do I find safety information?**

Safety information is part of the product documentation. It warns you of potential dangers and gives instructions on how to prevent personal injury or damage caused by dangerous situations. Safety information is provided as follows:

- In Chapter 1.1, "Safety instructions", on page 11. The same information is provided in many languages as printed "Safety Instructions". The printed "Safety Instructions" are delivered with the product.
- Throughout the documentation, safety instructions are provided when you need to take care during setup or operation.

## 1.1 Safety instructions

Products from the Rohde & Schwarz group of companies are manufactured according to the highest technical standards. To use the products safely, follow the instructions provided here and in the product documentation. Keep the product documentation nearby and offer it to other users.

Use the product only for its intended use and within its performance limits. Intended use and limits are described in the product documentation such as the data sheet, manuals and the printed "Safety Instructions for Oscilloscopes and Accessories" document. If you are unsure about the appropriate use, contact Rohde & Schwarz customer service.

Using the product requires specialists or specially trained personnel. These users also need sound knowledge of at least one of the languages in which the user interfaces and the product documentation are available.

Reconfigure or adjust the product only as described in the product documentation or the data sheet. Any other modifications can affect safety and are not permitted.

Never open the casing of the product. Only service personnel authorized by Rohde & Schwarz are allowed to repair the product. If any part of the product is dam-

aged or broken, stop using the product. Contact Rohde & Schwarz customer service at https://www.rohde-schwarz.com/support.

In these safety instructions, the term "product" covers instruments (oscilloscopes), probes and their accessories.

### Lifting and carrying the instrument

Check the data sheet for the maximum weight of the instrument. A single person can only carry a maximum of 18 kg safely depending on age, gender and physical condition. If your instrument is heavier than 18 kg, do not move or carry it by yourself.

Use the instrument handles to move or carry the instrument. Do not use the mounted accessories instead of the handles. Accessories are not designed to carry the weight of the instrument.

To move the instrument safely, you can use lifting or transporting equipment such as lift trucks and forklifts. Follow the instructions provided by the equipment manufacturer.

### Choosing the operating site

Only use the product indoors. The product casing is not waterproof. Water that enters can electrically connect the casing with live parts, which can lead to electric shock, serious personal injury or death if you touch the casing. If Rohde & Schwarz provides accessories designed for outdoor use of your product, e.g. a protective cover, you can use the product outdoors.

Unless otherwise specified in the data sheet, you can operate the product up to an altitude of 2000 m above sea level.

The product is suitable for pollution degree 2 environments where nonconductive contamination can occur. For more information on environmental conditions such as ambient temperature and humidity, see the data sheet.

### Setting up the product

Always place the product on a stable, flat and level surface with the bottom of the product facing down. If the product is designed for different positions, secure the product so that it cannot fall over.

If the product has foldable feet, always fold the feet completely in or out to ensure stability. The feet can collapse if they are not folded out completely or if the product is moved without lifting it. The foldable feet are designed to carry the weight of the product, but not an extra load.

If stacking is possible, keep in mind that a stack of products can fall over and cause injury.

If you mount products in a rack, ensure that the rack has sufficient load capacity and stability. Observe the specifications of the rack manufacturer. Always install the products from the bottom shelf to the top shelf so that the rack stands securely. Secure the product so that it cannot fall off the rack.

### Connecting to power and grounding

The mains power supply input of the instrument complies with overvoltage category II. It has to be connected to a fixed installation used to supply energy-consuming equipment such as household appliances and similar loads. Be aware that electrically powered products have risks, such as electric shock, fire, personal injury or even death.

Take the following measures for your safety:

- Do not use an isolating transformer to connect the instrument to the mains power supply.
- Before switching on the product, ensure that the voltage and frequency indicated on the product match the available power source. If the power adapter does not adjust automatically, set the correct value and check the rating of the fuse.
- Only use the power cable delivered with the product. It complies with country-specific safety requirements. Only insert the plug into an outlet with protective conductor terminal.
- If a product has an exchangeable fuse, its type and characteristics are indicated next to the fuse holder. Before changing the fuse, switch off the instrument and disconnect it from the power source. How to change the fuse is described in the product documentation.
- Only use intact cables and route them carefully so that they cannot be damaged. Check the power cables regularly to ensure that they are undamaged. Also ensure that nobody can trip over loose cables.
- If the product needs an external power supply, use the power supply that is delivered with the product or that is recommended in the product documentation or a power supply that conforms to the country-specific regulations.
- Ensure that you can disconnect the product from the power source at any time. Pull the power plug to disconnect the product. The power plug must be easily accessible. If the product is integrated into a system that does not meet these requirements, provide an easily accessible circuit breaker at the system level.
- Replace parts that are relevant to safety only by original parts, e.g. power cables or fuses.

### Performing measurements

Take the following measures for your safety:

- To ascertain voltage-free state, use an appropriate voltage tester. Any measurement setup including an oscilloscope is not suitable for this purpose.
- The maximum input voltage on channel inputs and the external trigger input must not exceed the value specified in the data sheet.
- Observe all voltage and current ratings of the instrument, the probes, and the accessories. Exceeding the allowed voltages can lead to an electric shock. Limits and ratings are marked on the products and listed in the data sheets. Consider that the rated voltage depends on the frequency. The voltage limitation curves or values are provided in the data sheet.
- Never cause any short circuits when measuring sources with high output currents.
- Use only probes and accessories that comply with the measurement category (CAT) of your measurement task. If the product is rated for any measurement cate-

gory, the permitted category is indicated on the product and in the data sheet. If you use other than Rohde & Schwarz accessories, make sure that they are suitable for the instrument and the measurement task.

- Set the correct attenuation factor on the instrument according to the probe being used. Otherwise, the measurement results do not reflect the actual voltage level, and you might misjudge the actual risk.
- When working with high voltages and current probes, observe the additional operating conditions specified in these safety instructions.
- The probe pins are extremely pointed and can easily penetrate clothes and the skin. Handle the probe pins with great care. To exchange a probe pin, use tweezers or pliers to avoid injuries. When transporting the accessories, always use the box supplied with the probe.
- Prevent the probe from receiving mechanical shock. Avoid putting excessive strain on the probe cable or exposing it to sharp bends. Touching a broken cable during measurements can cause injuries.
- Set up all probe connections to the instrument before applying power.

**Working with hazardous voltages**

Voltages higher than 30 V RMS, or 42 V peak, or 60 V DC are regarded as hazardous contact voltages. Direct contact with them can cause serious injuries.

Make sure that only electrically skilled persons use the products for measurements on hazardous contact voltages. These working conditions require special education and experience to perceive risks and to avoid hazards which electricity can create.

When working with hazardous contact voltages, use protective measures to preclude direct contact with the measurement setup:

- Do not touch exposed connections and components when power is applied.
- Switch off the test circuit while connecting and disconnecting probe leads.
- Use only insulated voltage probes, test leads and adapters.
- Make sure that the input leads fulfill the safety requirements for your measurement. The delivered input leads might have a jacket wear indicator that indicates a worn jacket by different jacket color. In this case, do not use the input lead. Replace it with a new one.
- When connecting to the DUT, keep your fingers behind finger guard. Remove jewelry, watches, and other metallic objects. Only use 4 mm safety banana plugs.

**Working with current probes**

When working with current probes, you can measure high-frequency currents or currents that contain high-frequency components.

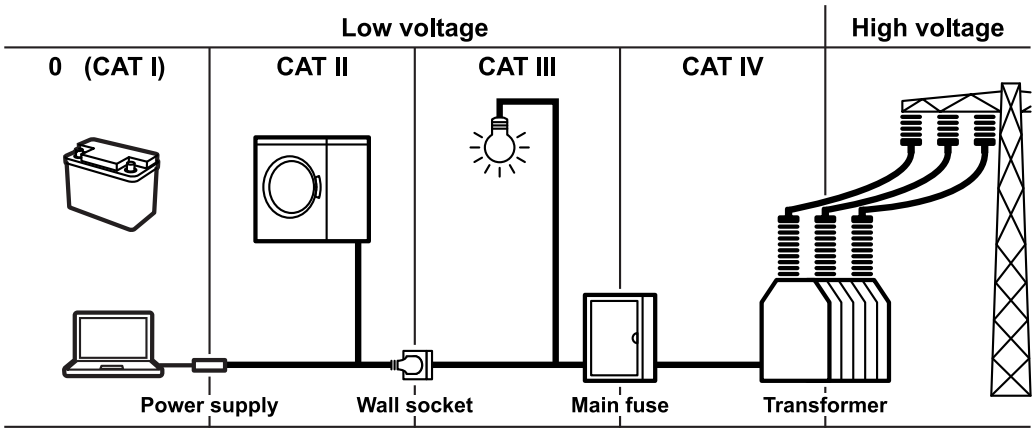
- Switch off the test circuit while connecting the probe.
- Do not attach the clamp to bare unisolated conductors. To avoid injury from a short circuit, measure at a location on an insulated wire where the insulation is sufficient for the circuit voltage.
- Connect the probe only to the secondary side of a breaker. With this measure, you avoid injury, if a short circuit occurs.

- The following effects can cause burns and fire or damage to the measurement site:
  – Eddy current loss can cause heating of the sensor head.
  – Dielectric heating can cause heating of cord insulation and other materials.
- When measuring current that includes a high-frequency component, consider the derating characteristics of the probe. Do not measure any current that exceeds the rated current.
- Using the probes with high frequencies or strong magnetic fields may cause the device to become abnormally hot, resulting in fire, equipment damage, or burns.

### Measurement categories

IEC 61010-2-030 defines measurement categories that rate instruments on their ability to resist short transient overvoltages that occur in addition to the working voltage. Use the measurement setup only in electrical environments for which they are rated.

- 0 - Instruments without rated measurement category
  For measurements performed on circuits not directly connected to mains, for example, electronics, circuits powered by batteries, and specially protected secondary circuits. This measurement category is also known as CAT I.
- CAT II:
  For measurements performed on circuits directly connected to the low-voltage installation by a standard socket outlet, for example, household appliances and portable tools.
- CAT III:
  For measurements performed in the building installation, such as junction boxes, circuit breakers, distribution boards, and equipment with permanent connection to the fixed installation.
- CAT IV:
  For measurements performed at the source of the low-voltage installation, such as electricity meters and primary overcurrent protection devices.




### Cleaning the product

Use a dry, lint-free cloth to clean the product. When cleaning, keep in mind that the casing is not waterproof. Do not use liquid cleaning agents.

**Meaning of safety labels**

Safety labels on the product warn against potential hazards.

| | |
|---|---|
| | Potential hazard<br>Read the product documentation to avoid personal injury or product damage. |
| | Heavy product<br>Be careful when lifting, moving or carrying the product. Carrying the product requires a sufficient number of persons or transport equipment. |
| | Electrical hazard<br>Indicates live parts. Risk of electric shock, fire, personal injury or even death. |
| | Protective conductor terminal<br>Connect this terminal to a grounded external conductor or to protective ground. This connection protects you against electric shock if an electric problem occurs. |

## 1.2 Labels on the product

Labels on the casing inform about:

- Personal safety, see "Meaning of safety labels" on page 16
- Product and environment safety, see Table 1-1
- Identification of the product

***Table 1-1: Labels regarding product and environment safety***

| | |
|---|---|
| | Chassis grounding terminal |
| | Take care when handling electrostatic sensitive devices. |
| | Labeling in line with EN 50419 for disposal of electrical and electronic equipment after the product has come to the end of its service life.<br>For more information, see "Disposing of electrical and electronic equipment" on page 2478. |

## 1.3 Warning messages in the documentation

A warning message points out a risk or danger that you need to be aware of. The signal word indicates the severity of the safety hazard and how likely it will occur if you do not follow the safety precautions.

**WARNING**

Potentially hazardous situation. Could result in death or serious injury if not avoided.

**CAUTION**

Potentially hazardous situation. Could result in minor or moderate injury if not avoided.

**NOTICE**

Potential risks of damage. Could result in damage to the supported product or to other property.

## 1.4 Where to find key documents on Rohde & Schwarz

Certificates issued to Rohde & Schwarz that are relevant for your country are provided at www.rohde-schwarz.com/key-documents, e.g. concerning:

- Quality management
- Environmental management
- Information security management
- Accreditations

## 1.5 Korea certification class A

이 기기는 업무용(A급) 전자파 적합기기로서 판매자 또는 사용자는 이 점을 주의하시기 바라며, 가정외의 지역에서 사용하는 것을 목적으로 합니다.

# 2 Preface

## 2.1 Key features

The R&S RTP high-performance oscilloscopes combine superior signal integrity with fastest acquisition rate. It combines multiple instrument capabilities for time-correlated debugging in one box. Outstanding key features are:

Real-time signal integrity

- Superior frontend performance
- Real-time de-embedding and advanced triggering
- Hardware accelerated processing for faster results
- Comprehensive analysis tools, for example, powerful protocol analysis, jitter analysis functions, automatic compliance tests, spectrograms, and precise and wideband RF analysis

Multi-instrument capability

- Time-correlated analysis of different signal types
- Integrated arbitrary waveform and pattern generator that can be used, for example, for protocol-based programming
- Easy-to-use serial protocol analysis for various protocols used in embedded design, mobile communications and more
- General-purpose mixed signal option
- Multi-channel power consumption analysis with up to 8 additional voltage and 8 additional current measurement channels and very high dynamic range
- 16 GHz pulse source

Compact and configurable

- Most compact high-end instrument platform
- Highest configuration flexibility
- Advanced user interface

For a detailed specification, refer to the data sheet.

## 2.2 Documentation overview

This section provides an overview of the R&S RTP user documentation.

### 2.2.1 Manuals and instrument help

You find the manuals on the product page at:

www.rohde-schwarz.com/manual/rtp

### Getting started manual

Introduces the R&S RTP and describes how to set up and start working with the instrument, and describes basic operations. A printed English version is included in the delivery. Editions in other languages are available on the product website.

### Instrument help

The help offers quick, context-sensitive access to the complete information for the firmware basic functionality and applications.

### User manual (Instrument)

Describes all instrument functions in detail. It also provides an introduction to remote control, a complete description of the remote control commands with programming examples, and information on maintenance and instrument interfaces. Includes the contents of the getting started manual.

The *online version* of the user manual provides the complete contents for immediate display on the internet.

### Manuals for compliance test options

For compliance test options, extra test procedure manuals are available. Test fixtures are described in printed manuals, which are delivered with the fixture.

- The following test procedure manuals are available:
  - USB 2.0/3.2 Compliance Test Procedures
  - Ethernet Compliance Tests Procedures
  - MIPI D-PHY Compliance Tests Procedures
  - eMMC Compliance Tests Procedures
  - PCIe Compliance Tests Procedures
  - DDR3 Compliance Tests Procedures
  - DDR4 Compliance Tests Procedures
  - HDMI Tests Procedures
  - ScopeSuite Automation
- The following test fixture manuals are available:
  - R&S RT-ZF1 USB 2.0 Compliance Test Fixture Set
  - R&S RT-ZF2 Ethernet Compliance Test Fixture Set
  - R&S RT-ZF3 Frequency Converter Board (100BASE-T1)
  - R&S RT-ZF4 10BASE-Te Test Fixture
  - R&S RT-ZF5 Ethernet Probing Fixture
  - R&S RT-ZF6 Frequency Converter Board (1000BASE-T1)
  - R&S RT-ZF7 Automotive Ethernet T&D Fixture
  - R&S RT-ZF7A and R&S RT-ZF7P SMA Adaptors
  - R&S RT-ZF8 Automotive Ethernet Compliance Test Fixture

**Safety instructions**

Provides safety information in many languages. The printed document is delivered with the product.

**Instrument security procedures manual**

Deals with security issues when working with the R&S RTP in secure areas. It is available for download on the internet.

**Service Manual**

Describes the performance test for checking the rated specifications, module replacement, firmware update, troubleshooting and fault elimination, and contains mechanical drawings and spare part lists. The service manual is available for registered users on the global Rohde & Schwarz information system (GLORIS, https://gloris.rohde-schwarz.com).

### 2.2.2 Data sheet and brochure

The data sheet contains the technical specifications of the R&S RTP. It also lists the firmware applications and their order numbers, and optional accessories. The brochure provides an overview of the instrument and deals with the specific characteristics.

See www.rohde-schwarz.com/brochure-datasheet/rtp

### 2.2.3 Release notes, open source acknowledgment

The release notes list new features, improvements and known issues of the current firmware version, and describe the firmware installation. The open source acknowledgment document provides verbatim license texts of the used open source software. It can also be read directly on the instrument.

See www.rohde-schwarz.com/firmware/rtp.

### 2.2.4 Application notes, application cards, videos

These documents deal with special applications or background information on particular topics.

- Application notes
- Media center
- Application videos

## 2.3 Options described in this document

In addition to the base unit, the following options are described in this documentation:

| Type | Designation | Order No. |
|---|---|---|
| R&S RTP-B1 | MSO | 1333.2424.02 |
| R&S RTP-B6 | Waveform and pattern generator | 1333.2418.02 |
| R&S RTP-B7 | Pulse source | 1333.2030.02 |
| R&S RTP-K1 | I²C and SPI serial triggering and decoding | 1337.8604.02 |
| R&S RTP-K2 | UART/RS-232/RS-422/RS-485 serial triggering and decoding | 1337.8610.02 |
| R&S RTP-K3 | CAN and LIN serial triggering and decoding | 1337.8627.02 |
| R&S RTP-K6 | MIL-STD-1553 serial triggering and decoding | 1800.6654.02 |
| R&S RTP-K7 | ARINC 429 serial triggering and decoding | 1800.6660.02 |
| R&S RTP-K8 | Ethernet serial triggering and decoding | 1337.8633.02 |
| R&S RTP-K9 | CAN-FD serial triggering and decoding | 1337.8640.02 |
| R&S RTP-K11 | I/Q software interface | 1800.6683.02 |
| R&S RTP-K12 | Jitter analysis | 1337.8656.02 |
| R&S RTP-K133 | Advanced jitter analysis | 1800.6860.02 |
| R&S RTP-K134 | Advanced noise analysis | 1800.6977.02 |
| R&S RTP-K19 | Zone trigger | 1337.8879.02 |
| R&S RTP-K37 | Spectrum analysis | 1338.1110.02 |
| R&S RTP-K35 | Bus analysis for I2C, SPI, RS232/UART, CAN, CAN-FD, LIN, SENT, 100BASE-Tx and 100BASE-T1. Requires corresponding protocol decoding option. | 1800.6648.02 |
| R&S RTP-K39 | Scripting interface | |
| R&S RTP-K40 | MIPI RFFE serial triggering and decoding | 1337.8733.02 |
| R&S RTP-K42 | MIPI D-PHY serial triggering and decoding | 1337.8740.02 |
| R&S RTP-K44 | MIPI M-PHY serial triggering and decoding | 1337.8756.02 |
| R&S RTP-K50 | Custom Manchester and NRZ serial triggering and decoding | 1337.8762.02 |
| R&S RTP-K52 | 8b10b serial triggering and decoding | 1337.8779.02 |
| R&S RTP-K55 | MDIO serial triggering and decoding | 1337.8785.02 |
| R&S RTP-57 | 100BASE-T1 serial triggering and decoding | 1800.6548.02 |
| R&S RTP-K58 | 1000BASE-T1 serial triggering and decoding | 1800.6702.02 |
| R&S RTP-K60 | USB 1.0/1.1/2.0/HSIC serial triggering and decoding | 1337.8791.02 |
| R&S RTP-K61 | USB 3.1 Gen 1 serial triggering and decoding | 1337.8804.02 |
| R&S RTP-K62 | USB 3.1 Gen 2serial triggering and decoding | 1337.9100.02 |
| R&S RTP-K63 | USB-PD serial triggering and decoding | 1337.8810.02 |
| R&S RTP-K64 | USB SSIC serial triggering and decoding | 1337.9117.02 |

| Type | Designation | Order No. |
|---|---|---|
| R&S RTP-K65 | SpaceWire serial triggering and decoding | 1800.6677.02 |
| R&S RTP-K72 | PCI Express 1.x/2.x serial triggering and decoding | 1337.8827.02 |
| K73 | PCI Express 3 serial triggering and decoding | 1800.6960.02 |
| R&S RTP-K91 | DDR3/DDR3L/LPDDR3 debug & compliance test. Decode and debug is described in this user manual. The compliance test is described in a separate manual. | 1337.8840.02 |
| R&S RTP-K93 | DDR4/LPDDR4 debug & compliance test. Decode and debug is described in this user manual. The compliance test is described in a separate manual. | 1801.3671.02 |
| R&S RTP-K121 | Deembedding base option | 1326.3064.02 |
| R&S RTP-K122 | Realtime deembedding extension | 1326.3070.02 |
| R&S RTP-K130 | TDR/TDT analysis (requires pulse source option R&S RTP-B7) | 1326.3093.02 |
| R&S RTP-K136 | Advanced eye analysis (8 Gb/s) | 1803.6561.02 |
| R&S RTP-K137 | Advanced eye analysis (16 Gb/s) | 1800.6983.02 |
| R&S RTP-K140 | High-speed serial pattern trigger (8 Gb/s) | 1326.4554.02 |
| R&S RTP-K141 | High-speed serial pattern trigger (16 Gb/s) | 1326.4560.02 |
| R&S RTP-K553 | External frontend support | |

## 2.4 Conventions used in the documentation

### 2.4.1 Typographical conventions

The following text markers are used throughout this documentation:

| Convention | Description |
|---|---|
| "Graphical user interface elements" | All names of graphical user interface elements on the screen, such as dialog boxes, menus, options, buttons, and softkeys are enclosed by quotation marks. |
| [Keys] | Key and knob names are enclosed by square brackets. |
| `Filenames, commands, program code` | Filenames, commands, coding samples and screen output are distinguished by their font. |
| *Input* | Input to be entered by the user is displayed in italics. |
| Links | Links that you can click are displayed in blue font. |
| "References" | References to other parts of the documentation are enclosed by quotation marks. |

### 2.4.2 Conventions for procedure descriptions

When operating the instrument, several alternative methods may be available to perform the same task. In this case, the procedure using the touchscreen is described. Any elements that can be activated by touching can also be clicked using an additionally connected mouse. The alternative procedure using the keys on the instrument or the on-screen keyboard is only described if it deviates from the standard operating procedures.

The term "select" may refer to any of the described methods, i.e. using a finger on the touchscreen, a mouse pointer in the display, or a key on the instrument or on a keyboard.

# 3 Getting started

Note: the following chapters are identical to those in the R&S RTP Getting Started manual for R&S RTP instruments.


- Preparing for use.....24
- Instrument tour.....31
- Operating the instrument.....43


## 3.1 Preparing for use

Here, you can find basic information about setting up the instrument for the first time or when changing the operating site.

### 3.1.1 Lifting and carrying

See: "Lifting and carrying the instrument" on page 12.

### 3.1.2 Unpacking and checking

1. Unpack the product carefully.
2. Retain the original packing material. Use it when transporting or shipping the product later.
3. Using the delivery notes, check the equipment for completeness.
4. Check the equipment for damage.

   If the delivery is incomplete or equipment is damaged, contact Rohde & Schwarz.

### 3.1.3 Choosing the operating site

Specific operating conditions ensure proper operation and avoid damage to the product and connected devices. For information on environmental conditions such as ambient temperature and humidity, see the data sheet.

See also "Choosing the operating site" on page 12.

**Electromagnetic compatibility classes**

The electromagnetic compatibility (EMC) class indicates where you can operate the product. The EMC class of the product is given in the data sheet.

- Class B equipment is suitable for use in:
  - Residential environments

  - Environments that are directly connected to a low-voltage supply network that supplies residential buildings
- Class A equipment is intended for use in industrial environments. It can cause radio disturbances in residential environments due to possible conducted and radiated disturbances. It is therefore not suitable for class B environments.
  If class A equipment causes radio disturbances, take appropriate measures to eliminate them.

### 3.1.4 Setting up the product

When setting up the instrument, follow the safety instructions:

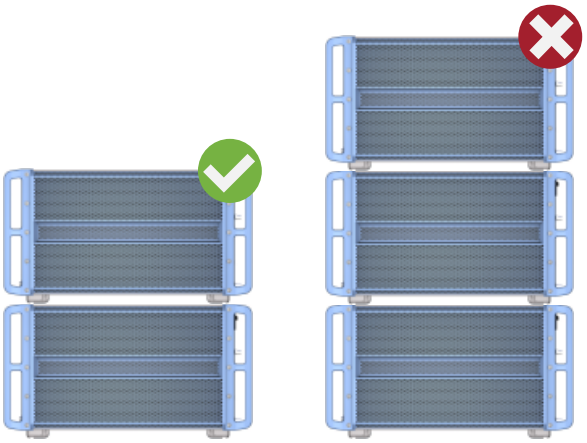
- "Setting up the product" on page 12
- "Intended use" on page 11

#### 3.1.4.1 Placing the product on a bench top

For standalone operation, place the instrument on a horizontal bench with even, flat surface. The instrument can be used in horizontal position, standing on its feet, or with the support feet on the bottom extended.

**To place the product on a bench top**

1. Place the product on a stable, flat and level surface. Ensure that the surface can support the weight of the product. For information on the weight, see the data sheet.
2. **CAUTION!** Foldable feet can collapse. See "Setting up the product" on page 12.
   Always fold the feet completely in or out. With folded-out feet, do not place anything on top or underneath.
3. **WARNING!** A stack of products can fall over and cause injury. Never stack more than two products. Otherwise, mount them in a rack.
   Stack as follows:
   - All products must have the same dimensions (width and length).
   - Do not exceed a total load of 25 kg placed on the product at the bottom of the stack.
   - Fold in all foldable feet completely.



Left = Stacked correctly
Right = Stacked incorrectly, too many products

4. **NOTICE!** Overheating can damage the product.
   Prevent overheating as follows:
   - Keep a minimum distance of 10 cm between the fan openings of the product and any object in the vicinity to provide sufficient airflow and ventilation.
   - Do not place the product next to heat-generating equipment such as radiators or other products.

### 3.1.4.2 Mounting the product in a rack

The instrument can be installed in a rack using a rack adapter kit. The order number is given in the data sheet. The installation instructions are part of the adapter kit.

**To prepare the rack**

1. Observe the requirements and instructions in "Setting up the product" on page 12.
2. **NOTICE!** Insufficient airflow can cause overheating and damage the product.
   Design and implement an efficient ventilation concept for the rack.

**To mount the R&S RTP in a rack**

1. Use an adapter kit that fits the dimensions of the R&S RTP to prepare the instrument for rack mounting. For information on the dimensions, see data sheet.
   a) Order the rack adapter kit designed for the R&S RTP. For the order number, see data sheet.
   b) Mount the adapter kit. Follow the assembly instructions provided with the adapter kit.
2. Push the product onto the shelf until the rack brackets fit closely to the rack.
3. Tighten all screws at the rack brackets with a tightening torque of 1.2 Nm to secure the product at the rack.

**To unmount the product from a rack**

1. Loosen the screws at the rack brackets.
2. Remove the product from the rack.
3. If placing the product on a bench top again, unmount the adapter kit from the product. Follow the instructions provided with the adapter kit.

### 3.1.5 Considerations for test setup

Observe safety instructions, see "Performing measurements" on page 13.

**Cable selection and electromagnetic interference (EMI)**

Electromagnetic interference (EMI) can affect the measurement results.

To suppress electromagnetic radiation during operation:

- Use high-quality shielded cables, for example, double-shielded RF and LAN cables.
- Always terminate open cable ends.
- Ensure that connected external devices comply with EMC regulations.

**Signal input and output levels**

Information on signal levels is provided in the data sheet. Keep the signal levels within the specified ranges to avoid damage to the product and connected devices.

**Preventing electrostatic discharge (ESD)**

Electrostatic discharge is most likely to occur when you connect or disconnect a DUT.

► **NOTICE!** Electrostatic discharge can damage the electronic components of the product and the device under test (DUT).

Ground yourself to prevent electrostatic discharge damage:

a) Use a wrist strap and cord to connect yourself to ground.
b) Use a conductive floor mat and heel strap combination.

During operation, if the firmware observes a serious unexpected disturbance (e.g. due to ESD), it resets some hardware components and initiates a new alignment to ensure proper instrument functioning. Then it restores the user settings to the state before the disturbance.

### 3.1.6 Connecting to power

For safety information, see "Connecting to power and grounding" on page 13.

The R&S RTP can be used with different AC power voltages and adapts itself automatically to it.

The instrument has a C20 connector. A list of appropriate power cables is given in the user manual. Refer to the data sheet for the requirements of voltage and frequency.

The nominal ranges are:

- 100 V to 240 V AC at 50 Hz to 60 Hz, and 100 V to 130 V at 400 Hz, with maximal 10% voltage fluctuation on line
- max. 13 A to 4.7 A

1. Plug the AC power cable into the AC power connector on the rear panel of the product. Only use the AC power cable that is delivered with the product.
2. Plug the AC power cable into a power outlet with ground contact.

   The required ratings are listed next to the AC power connector and in the data sheet.

### 3.1.7 Switching on or off

The instrument is switched on or off with the power switch and the [Power] key. The [Power] key is located in the bottom left corner of the front panel. The power switch is located at the rear panel of the instrument.

***Table 3-1: Overview of power states***

| Status | LED | Power switch |
|---|---|---|
| Off | (unlighted) | [0] (off) |
| Standby | orange | [I] (on) |
| Ready | green | [I] (on) |

**To switch on the product**

The product is off but connected to power.

1. Set the switch on the power supply to position [I].

   The power key on the front panel lights up.
2. Press the [Power] key on the front panel.

   The instrument performs a system check, boots the operating system, and then starts the R&S RTP firmware.
   The [Power] key turns green and the illuminated keys on the front panel light up. If the previous session was terminated regularly, the oscilloscope uses the last settings.

Before you start measurements, be sure to comply with the warm-up phase specified in the data sheet.

**To shut down the product**

The product is in the ready state.

► Press the [Power] key.

All current settings are saved, and the software shuts down. The [Power] key turns orange. The standby power supplies only the power switch circuits.

The "Power" icon in the "Menu" shuts down only the firmware application. To shut down the instrument completely, use the [Power] key.

**To disconnect from power**

The product is in the standby state.

1. **NOTICE!** Risk of data loss. If you disconnect the product from power when it is in the ready state, you can lose settings and data. Shut it down first.

   Set the switch on the power supply to position [0].

   The LED of the [Power] key is switched off.

2. Disconnect the product from the power source.

## 3.1.8 Connecting external devices

The following interfaces for external devices are provided:

- USB connectors at the front and rear panel of the instrument
- Monitor connectors HDMI and DisplayPort at the rear panel of the instrument

### 3.1.8.1 Connecting USB devices

You can connect USB flash drives to save screenshots and measurement results, and keyboard and/or mouse to simplify the operation and the entry of data. You can connect or disconnect all USB devices during operation of the instrument. The number of USB connectors can be increased by using USB hubs.

USB devices are plug&play, the operating system automatically uses the suitable device driver. If the operating system does not find a suitable driver, it prompts you to specify a directory that contains the driver software. If the driver software is on a storage media, connect the appropriate drive to the instrument before proceeding. If the instrument is integrated in a network, you can also install driver data stored in a network directory.

When a USB device is disconnected from the R&S RTP, Windows immediately detects the change in hardware configuration and deactivates the corresponding driver.

The properties of external USB devices are configured in the operating system, not in the instrument's firmware. We recommended using mouse and keyboard to access and modify the settings of the Windows operating system.

To access Windows, press the Windows key on the external keyboard, or select "Menu" > "Minimize Application" on the R&S RTP menu.

**Connecting a keyboard or mouse**

1. Connect the keyboard or mouse.

   The device is detected automatically. The default input language of the keyboard is English – US.

2. To configure the keyboard properties, press the Windows key on the external keyboard.

3. Tap the "Find" icon (magnifier) on the Windows taskbar.

4. To configure the keyboard:

   a) Type *keybord*.
   b) Select "Edit language and keyboard options".

5. To configure the mouse:

   a) Type *mouse*.
   b) Select "Mouse settings".

### 3.1.8.2 Connecting external monitors

You can connect one or two external monitors or projectors to the R&S RTP. The following connectors are available:

- "HDMI" on page 35
- "DisplayPort" on page 35

1. Ensure that the monitors and the R&S RTP are connected to a ground contact. Otherwise the instrument can be damaged.

2. Check the input type of the monitors. Make sure to select the correct cable. If the connectors do not match, use an appropriate adapter.

3. You can configure the displays directly in the instrument's setup. Open "Menu" > "Settings" > "Display".

4. Select the "Monitors" tab.

5. Select how to display the screen. Select "Extend" to arrange dialogs and result boxes on different displays.

6. To move the main firmware window to an external monitor, press F11 on the keyboard.

   Now the firmware shows a soft frontpanel together with the usual display in a window. You can move this window to an external monitor.

7. To access Windows display settings, tap "Additional settings".

The touchscreen of the R&S RTP has a screen resolution of 1920x1080 pixel. Many external monitors have a higher screen resolution. If the screen resolution of the monitor is set higher than the instrument's resolution, the firmware window uses a

1920x1080 area of the monitor display. For full screen display, adjust the monitor's screen resolution using "Additional display settings".

## 3.2 Instrument tour

This chapter describes the front and rear panels of the instrument including all function keys and connectors.

### 3.2.1 Front panel

This section provides an overview of the R&S RTP front panel and explains the various connectors and the option slots at the front.

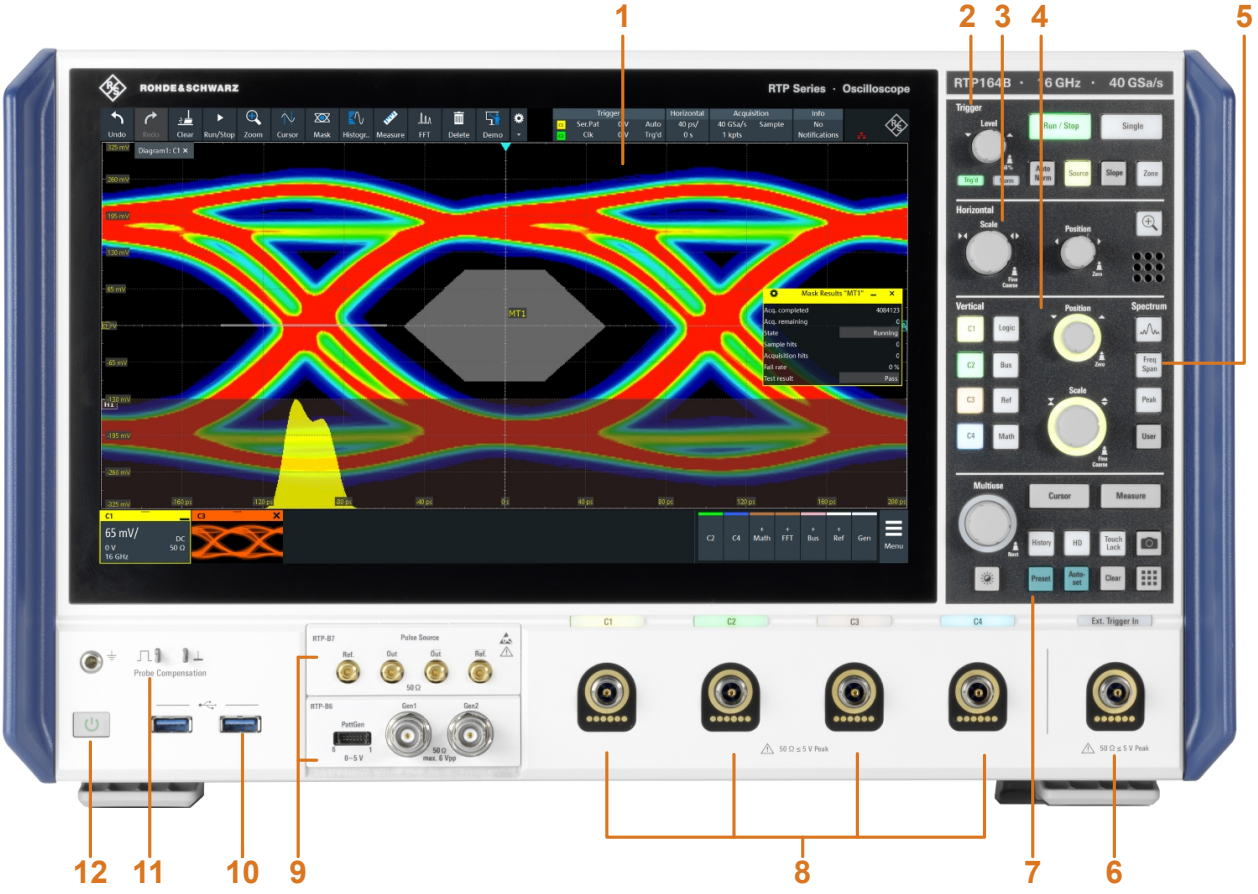


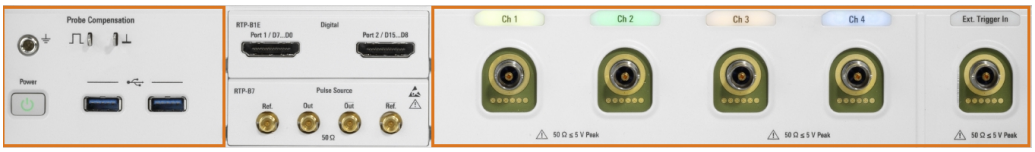

*Figure 3-1: Front panel of R&S RTP*

1 = Touchscreen
2 = Trigger controls
3 = Horizontal controls
4 = Vertical controls
5 = Spectrum keys
6 = External trigger input

7 = Analysis keys
8 = Input channels
9 = Two option slots for R&S RTP-B1 (MSO) or R&S RTP-B1E (for R&S RT-ZVC) , R&S RTP-B6 (waveform generator), R&S RTP-B7 (pulse source)
10 = USB connectors
11 = Connectors for probe compensation and grounding
12 = [Power] key

The keys and controls are described in Chapter 3.2.3, "Keys and controls", on page 36.

### 3.2.1.1 Connectors




**Channel inputs and external trigger input**
The R&S RTP has four channel inputs to connect the input signals, and an external trigger input to control the measurement by an external signal.

The input connectors are provided with a special Rohde & Schwarz active probe interface, and they are BNC compatible. Thus, the instrument can automatically detect active probes that have the Rohde & Schwarz probe interface.

The maximum input voltage on channel inputs and the external trigger input is 5 V peak. The input impedance of the inputs is 50 Ω.

**USB**
Two USB type A connectors that comply with standard USB 3.1 gen 1. They are used to connect devices like keyboard, mouse and USB flash drive.

**Note:** Electromagnetic interference (EMI) can affect the measurement results. To avoid any impact, do not use USB connecting cables exceeding 1 m.

**Probe Comp.**
Probe compensation terminal to support adjustment of passive probes to the oscilloscope channel.

⏚ Protective earth conductor, 4 mm banana jack to connect ground of DUT and test fixtures, and wrist strap.

⎍ Square wave signal for probe compensation, 1 kHz and 1 $V_{pp}$.

⊥ Ground connector for probes.

### 3.2.1.2 Option slots

Three options R&S RTP-B1/B6/B7 can be installed in one R&S RTP - two at the front panel, and one at the rear panel. Each option can be installed only once, and the slot assignment depends on the available options. To ensure correct installation and calibration, installation is done only at Rohde & Schwarz service centers.

See also: "Option slot" on page 36.

**Mixed signal option R&S RTP-B1 / digital extension port R&S RTP-B1E**
The mixed signal option R&S RTP-B1 provides logic analizer functionality and 16 digital channels. It also can connect the R&S RT-ZVC multi-channel power probe. The connectors are used to connect two logical probes with 8 digital channels each (D0 to D7 and D8 to D15), or two flat interface cables of R&S RT-ZVC.

The digital extension port R&S RTP-B1E provides the connectors for the R&S RT-ZVC multi-channel power probe without MSO. The connectors are used to connect two flat interface cables of R&S RT-ZVC.

One of the options can be installed in one of option slots at the front panel, or in the option slot at the rear panel.

The maximum input voltage is 40 V peak at 100 kΩ input impedance. The maximum input frequency for a signal with the minimum input voltage swing of 500 mV ($V_{pp}$) is 400 MHz. For detailed specifications, refer to the data sheet.

**Waveform generator option R&S RTP-B6**
The waveform generator generates various function and arbitrary waveforms, sweeps, and parallel patterns. For detailed specifications, refer to the data sheet.

The option can be installed in one of option slots at the front panel.

The module provides the following connectors:

[Gen1, Gen2] BNC connectors

[PattGen] Connector for the pattern generator

**Pulse source option R&S RTP-B7**
The pulse source outputs a symmetrical differential pulse signal. For detailed specifications, refer to the data sheet.

The option can be installed in one of option slots at the front panel.

The module has four connectors.
- Out, $\overline{\text{Out}}$: 2.92 mm connectors (K type) for pulse signal output
- Ref, $\overline{\text{Ref}}$: 2.92 mm connectors (K type) for reference signal output

### 3.2.2 Rear panel

This section explains the rear panel of the R&S RTP, including the various connectors and the option slot.

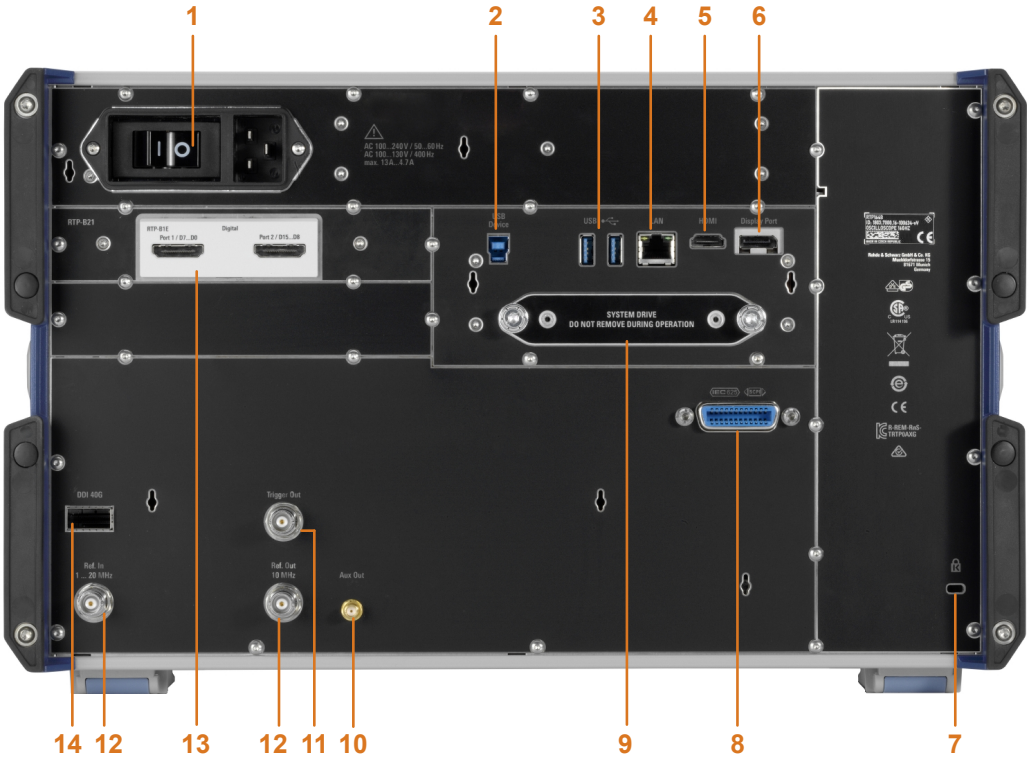



*Figure 3-2: Rear panel of R&S RTP*

1 = AC power supply connector and main power switch
2 = USB device connector, type B
3 = USB connectors, type A
4 = LAN connector
5 = HDMI connector for external monitor
6 = DisplayPort connector
7 = Kensington lock slot to secure the instrument against theft
8 = GPIB connector
9 = Exchangeable solid state disk (SSD, option R&S RTP-B19)
10 = Aux Out connector
11 = External trigger output
12 = Ref In, Ref Out: reference input and output of the OCXO reference signal
13 = Option slot
14 = Digital data interface (DDI, optional)

### AC power supply connector and main power switch

Connection to the AC power line. The R&S RTP can be used with different AC power voltages and adapts itself automatically to it. The nominal voltage and frequency ranges are displayed on the rear panel and quoted in the data sheet.

If grounding is *not* ensured by the mains system, ground the oscilloscope using the protective earth conductor on the front panel and an appropriate cable.

The AC main power switch also interrupts the power supply of the OCXO.

When you power up the instrument, be sure to comply with the warm-up phase specified in the data sheet before you start measurements.

See also: Chapter 3.1.6, "Connecting to power", on page 27.

**USB Device**
USB 3.0 interface of type B (device USB), to be used for remote control of the instrument.

**USB**
Two USB type A connectors that comply with standard USB 3.0. They are used to connect devices like keyboard, mouse and flash drive to store and reload instrument settings and measurement data.

**Note:** Electromagnetic interference (EMI) can affect the measurement results. To avoid any impact, do not use USB connecting cables exceeding 1 m.

**LAN**
8-pin RJ-45 connector used to connect the instrument to a Local Area Network (LAN). It supports up to 1000 Mbit/s (10/100/1000BASE-T Ethernet).

**HDMI**
HDMI Version 2.0 connector for an external monitor or projector. The monitor shows the complete content of the instrument's screen.

See also: Chapter 3.1.8.2, "Connecting external monitors", on page 30.

**DisplayPort**
DisplayPort connector for an external monitor or projector. It supports DisplayPort version 1.1a.

**GBIP**
Connector with GBIP interface. For detailed specifications, refer to the data sheet.

**Aux Out**
Output of the internal calibration signal, if the signal is configured to external destination.

**Trigger Out**
The BNC connector for external trigger output is used to provide the internal trigger signal of the oscilloscope to trigger other instruments for synchronized measurements.

When a trigger occurs, the R&S RTP creates a pulse of 5 V with a source impedance of 50 Ω and delivers it to the external trigger output. The instrument can also send the pulse on mask test violation or violation of measurement limits and margins.

If the connector is terminated with 50 Ω, the signal level is 2.5 V (50 mA). With 1 MΩ termination, the level is 5 V. A short-circuit of the connector to ground creates current of 100 mA.

To enable the trigger out signal, select the "Trigger out" action for the trigger or test. You can adjust polarity and length of the pulse. The default is a positive pulse of 100 ns. The delay is 800 ns.

**Ref In, Ref Out**
BNC connectors for input of a reference signal, and for output of the OCXO reference signal.

The input frequency ranges from 1 MHz to 20 MHz in 1 MHz steps. The input impedance is 50 Ω.

The output frequency of the OCXO is 10 MHz, the impedance is 50 Ω. For detailed specifications, refer to the data sheet.

**DDI 40G**
Optional digital data interface for input and output of digital data.

**Option slot**
Slot to install one of the options R&S RTP-B1 (MSO) or R&S RTP-B1E (for R&S RT-ZVC) if both slots at the front panel are used for other options. To install the option, the R&S RTP-B21 slot adapter is required. The options and their connectors are briefly described in Chapter 3.2.1, "Front panel", on page 31.

## 3.2.3 Keys and controls

For an overview of the front panel keys, see Figure 3-1.

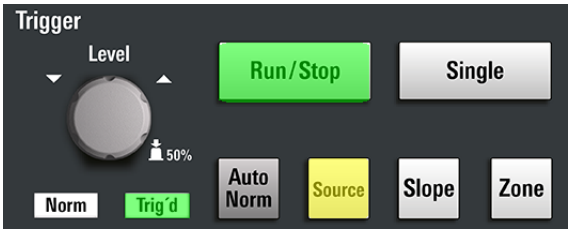
### 3.2.3.1 Power key

The [Power] key is located on the lower left corner of the front panel. It starts up and shuts down the instrument's software.

The light of the key shows the instrument state, see Chapter 3.1.7, "Switching on or off", on page 28.

### 3.2.3.2 Trigger controls

The keys and knob in the Trigger functional block adjust the trigger and start or stop acquisition.




**[Level]**
The rotary knob sets the trigger level for all trigger types. Turn clockwise to move up the trigger level. Press the knob to set the trigger level to 50% of the signal amplitude. If the selected trigger type requires two trigger levels - upper and lower level - press the knob to toggle between the two levels.

Remote command:
`TRIGger<m>:LEVel<n>[:VALue]` on page 1306

**[Run / Stop]**
Starts and stops the continuous acquisition. The [Run / Stop] key lights green when the acquisition is running. When the acquisition is stopped, the key lights red.

Remote command:
RUN on page 1235
STOP on page 1235

**[Single]**
Starts a defined number of acquisitions. The [Single] key lights green when the acquisition is running. When the acquisition is stopped, the key lights red.

To set the number of acquisitions, select "Menu" > "Acquire" > "Setup" tab, and set "N-single count".

Remote command:
SINGle on page 1235

**[Auto Norm]**
Toggles the trigger mode between "Auto" and "Normal". The current setting is shown on the trigger label.

| | |
|---|---|
| Auto | The instrument triggers repeatedly after a time interval if the trigger conditions are not fulfilled. |
| Norm | The instrument acquires a waveform only if a trigger occurs. The "Norm" LED lights up in green. |

Remote command:
TRIGger<m>:MODE on page 1370

**[Source]**
Opens a dialog box where you can select the trigger source. Press the key again to switch the source. The key lights up in the color of the selected trigger source.

Remote command:
TRIGger<m>:SOURce[:SELect] on page 1305

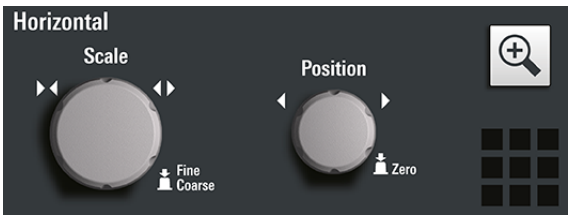
**[Slope]**
Toggles the trigger slope or trigger polarity, dependent on the trigger type. The current setting is shown on the trigger label.

**[Zone]**
Opens the overlay menu to configure the zone trigger. A zone trigger combines the trigger condition with the intersection or non-intersection of one or more zones or masks.

Remote command:
TRIGger<m>:ZONE:EXPRession[:DEFine] on page 1374

#### 3.2.3.3 Horizontal controls

The rotary knobs in the Horizontal functional block adjust the horizontal parameters. These settings are effective for all channel waveforms. In addition, a [Zoom] is available.




**[Scale]**
The rotary knob adjusts the time scale for all signals. The time scale is also known as timebase.

Turn clockwise to stretch the waveforms. Doing so, the scale value *time/div* decreases.

Press the knob to toggle between coarse and fine scale adjustment.

**[Position]**
The rotary knob changes the horizontal position of the waveform or the position of the reference point on the screen. To set the value to zero, press the knob. The current value is shown in the input box in the upper right corner of the screen.

You can select if the knob changes the position or the reference point in "Menu" > "Settings" > "Frontpanel" > "Knobs".

"Horizontal position" defines the time distance of the reference point from the zero point of the diagram. Turn clockwise to move the waveform to the right.

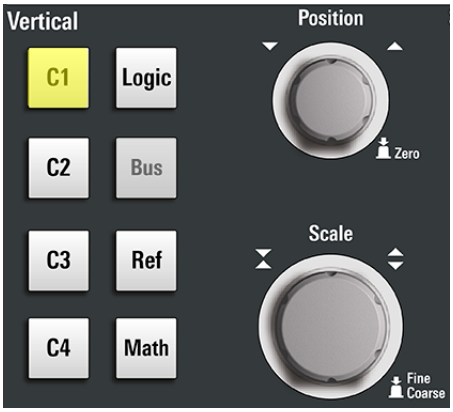
"Reference point" defines the position of the reference point on the screen. Turn clockwise to move it to the right. The reference point marks the rescaling center of the time scale. If you modify the time scale, the reference point remains fixed on the screen, and the scale is stretched or compressed to both sides of the reference point.

Remote command:
`TIMebase:HORizontal:POSition` on page 1237
`TIMebase:REFerence` on page 1237

**[Zoom]**
Displays a zoom diagram for the active diagram. The key is illuminated if at least one zoom is active. If you press the key while the zoom function is on, the "Zoom" dialog box opens, where you can configure several zoom areas for detailed signal observation.

### 3.2.3.4 Vertical controls

The keys and knobs in the Vertical functional block select a signal and adjust the vertical scale and position of the selected signal.




**[C<n>]**
Turns on and selects a channel. If the channel is active, the key lights up in the corresponding channel color.

The effect of the keypress depends on state of the channel:
- If channel is off: turns on the channel and selects it.
- If the channel is on and in focus (selected): opens the corresponding channel dialog.
- If the channel is on, but not in focus (not selected): selects the channel waveform.
- If the channel is selected, and the dialog is open: turns off the channel, and closes the dialog.

The vertical rotary knobs are focused on the selected waveform. They are illuminated in the color of the selected waveform.

Remote command:
`CHANnel<m>:STATe` on page 1248

**[Logic]**
Opens the "Logic" dialog to configure and enable the logic buses L1 to L4. A logic bus (or parallel bus) has up to 16 logic (digital) channels. The key lights up if at least one logic bus is enabled.

**[Bus]**
Opens and closes the "Protocol" tab in the "Apps" dialog, where you can select the required serial protocol and proceed to the protocol setup. The key lights up if the decoding of a serial bus is active.

**[Ref]**
Opens the "Reference" dialog box, where you can configure and display reference waveforms. Press the key repeatedly to switch the reference waveform.

If a reference waveform is selected, the vertical rotary knobs are illuminated in white or light gray (default colors), depending on the selected waveform.

**[Math]**
Opens the "Math" dialog, where you can configure the calculation of mathematical waveforms using various mathematic operations on other waveforms. Press the key repeatedly to toggle the subtabs of the math waveforms. After the last subtab, the key closes the dialog.

If a math waveform is selected, the vertical rotary knobs are illuminated in brown (default color), the brightness of the color depends on the selected waveform.

**[Position]**
The vertical [Position] knob changes the vertical offset or the position of the selected waveform. The horizontal axis and the selected waveform are moved vertically. You can select whether the knob changes the offset or the position in the "Menu" > "Settings" > "Frontpanel" > "Knobs" dialog.

- Position indicates the vertical location in divisions.
- Offset moves the vertical center of the selected channel to the offset value.

If the selected waveform is a math or reference waveform, serial bus, or logic channel, the knob changes its vertical position.

The knob lights up in the color of the selected waveform. Turn clockwise to move up the waveform. To set the value to zero, press the knob.

The current value is shown in the input box in the upper right corner of the screen.

Remote command:
`CHANnel<m>:POSition` on page 1250
`CHANnel<m>:OFFSet` on page 1250

**[Scale]**
Sets the vertical scale in Volts per division. The vertical scale defines the displayed amplitude of the selected waveform.

The knob lights up in the color of the selected waveform. Turn clockwise to stretch the waveform. Doing so, the scale value V/div decreases. Press the knob to toggle between coarse and fine scale adjustment. For analog waveforms, the scale value is shown in the signal icon.

To get the maximum resolution of the waveform amplitude, make sure that the waveforms cover most of the diagram's height.

Remote command:
`CHANnel<m>:SCALe` on page 1249

### 3.2.3.5 Spectrum and User keys

**[Spectrum]**
Opens and closes the "Math" dialog with FFT settings. The key lights up if an FFT is active.

**[Freq Span]**
Opens the FFT setup to configure the frequency, span, and other FFT parameters.

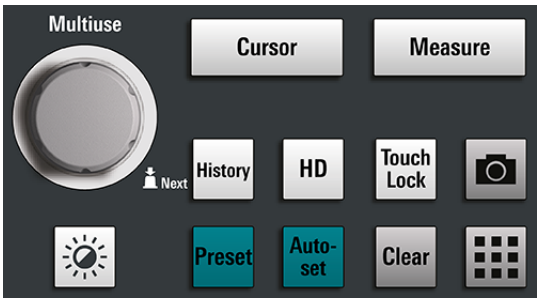
**[Peak]**
Opens the "Peak List" dialog, where you can enable the peak list generation and configure the display of the peak list

**[User]**
Opens the "Frontpanel" settings dialog, where you can configure the function of the camera key.

### 3.2.3.6 Analysis keys

The keys in the Analysis functional block provide direct access to measurement and analyzing functions.

The rotary knob provides an alternative way to navigate in dialog boxes and to enter numeric data.



**[Multiuse]**
The multiuse knob has various functions. It lights up when a function is active on the knob.

- In numeric entry fields: turn to increase or decrease the value.
- To set cursor positions, histogram areas, and mask points in input boxes: press to toggle the parameter, turn clockwise to increase the value or turn counterclockwise to decrease it.
- To move zoom area or gate in diagrams: Turn to move the element that has the focus, and press to toggle the focus.
- In tables: press to activate the edit mode, turn clockwise to increase the value or turn counterclockwise to decrease it, and press to enter the value and move to the next cell.
- To change the intensity: Press the [Intensity] key and turn the knob.

**[Intensity]**
Adjusts the intensity of the waveforms on the screen, or the background transparency of dialog boxes, or the transparency of result boxes.

Press the [Intensity] key and turn the [Multiuse] knob. Press the [Intensity] key again to toggle between the three settings. The controlled parameter and its value are shown in the input box in the upper right corner of the screen.

**[Cursor]**
Starts a cursor measurement: displays vertical and horizontal cursor lines in the active diagram, and the results of the cursor measurement.

If you press the key while a cursor measurement is enabled, the "Cursors" dialog opens. The next press closes the dialog.

Cursors are markers which are placed at points of interest on a waveform. The instrument measures the cursor positions and delta values between parallel cursors.

**[Measure]**
Opens and closes the "Measurements" dialog and the subtab of the last used measurement. In the dialog, you can configure and start automatic measurements.

**[History]**
Activates the history and opens the history player. The second press opens the player configuration. The history shows stored acquisitions that were acquired before the current one.

The key is illuminated as long as the history is active.

**[HD]**
Activates the high definition mode and opens the "HD Mode" dialog box.

**[Touch Lock]**
Locks the touchscreen to prevent unintended use. When the touchscreen is off, the key is illuminated. Press again to unlock the touchscreen.

**Camera**
Performs the action that is assigned to the key in "Menu" > "Settings" > "Frontpanel" > "Hardkeys". By default, the key saves a screenshot of the waveform display.

See also: Chapter 4.5.1, "Hardkeys: function assignment", on page 97

**[Preset]**
Resets the instrument to a predefined state and starts the continuous acquisition. All measurements, mask tests, zoom, and most individual settings are deleted, and all channels except for channel 1 are disabled. You can define preset configurations and save them to a file. The [Preset] key can be configured to set either factory defaults or a user-defined preset configuration.

**[Autoset]**
The instrument analyzes the enabled channel signals, and adjusts appropriate horizontal, vertical, and trigger settings to display stable waveforms.

**[Clear]**
Deletes all measurement results including long-term measurement and statistics, all waveforms, and the history.

**Apps**
Opens the "Apps" dialog, where you can start an application, serial protocol or another analysis function.

## 3.3 Operating the instrument

There are three ways to operate the R&S RTP.

### Manual operation

Use the touchscreen, keys and rotary knobs, or an optional mouse and/or keyboard. The principles of manual operation are explained in this section.

### Remote control

Create programs to automatize repeating settings, tests, and measurements. The instrument is connected to a computer that runs the program.

This way of operation is described in: Chapter 24, "Remote control commands", on page 1165

### Remote operation

The remote desktop connection of Windows can be used for instrument control and file transfer. Even on computers with non-Windows operating systems, a remote desktop connection is possible using RDP applications.

For details, refer to the user manual, chapter "Remote Desktop Connection".

Remote monitoring and control of the instrument from a connected computer is also possible with a standard web browser and a LAN connection.

For details, refer to the user manual, chapter "Web Control".

Alternatively, you can use Virtual Network Computing (VNC), which requires installation of the VNC server on the R&S RTP. Installation and configuration are described in the application note "Remote Monitoring and Control of the R&S RTP with a Web Browser", available on the Rohde & Schwarz internet site.

### 3.3.1 Means of manual interaction

The R&S RTP provides the following means of manual interaction, which you can use alternatively or complementary:

- Touchscreen:
  Using the touchscreen allows for direct interaction with the instrument. Use your finger to place waveforms on the screen, mark areas for zoom, set parameters in dialogs, enter data, and much more. The control elements and actions on the screen are based on common concepts, and you easily become familiar with the user interface.

  Tapping the screen works like clicking mouse buttons:

  – Tap = click: Selects a waveform or parameter, or provokes an action.
  – Double-tap = double-click: Has the same effect as touch and hold, it opens the on-screen keyboard or keypad, or a specific editor if available.

  Use gestures to scale the waveform:

  – Spread or pinch two fingers horizontally to change the time scale.
  – Spread or pinch two fingers vertically to change the vertical scale of the active waveform.
- Function keys and rotary knobs:
  The front panel provides frequently used functions and controls to operate the instrument. Use knobs to set levels and scales, and keys to initiate actions and to open dialogs.
- Optional mouse and/or keyboard
  These devices work in the normal manner.

### 3.3.2 Touchscreen display

#### 3.3.2.1 Information on the display

The touchscreen of the instrument shows the waveforms and measurement results, and also information and everything that you need to control the instrument. All waveform-related display elements are shown in Figure 3-3. An overview of control elements - like dialog box, toolbar - is given in Chapter 3.3.2.2, "Control elements on the touchscreen", on page 48.

*Figure 3-3: Display information*

1 = Diagram
2 = Grid
3 = Trigger position
4 = Reference point (distance from trigger position to reference point = horizontal position)
5 = Trigger, horizontal and acquisition label
6 = Notifications
7 = Zoom area
8 = Zoom diagram
9 = Trigger level
10 = Signal bar with inactive waveforms (10a), minimized live waveform (10b) and signal icon with vertical settings (10c)
11 = Histogram area
12 = Histogram

### Diagram

A diagram shows one or more waveforms: channel, reference, and math waveforms together with histograms, masks etc. Zoom details, XY-waveforms, spectra and other special waveforms are shown in separate diagrams.

By default, the diagram name contains the diagram number and the short names of the waveforms shown inside. To change the diagram name, touch and hold the tab name. The on-screen keyboard opens to enter the new name. Names must be unique.

To arrange the diagrams on the screen, the Rohde & Schwarz SmartGrid function helps you to find the target place simply and quickly. For details, see Chapter 3.3.5, "Rohde & Schwarz SmartGrid", on page 53. You can also adjust the diagram size by dragging the diagram border.

**Grid**
The grid shows the vertical and horizontal divisions. The division lines are labeled with the correspondent values. The grid labels have the color of the waveform to which they belong. If several waveforms are shown in one diagram, the grid has the color of the selected waveform.

**Trigger position and trigger level**
The blue markers show the horizontal position of the trigger and the vertical trigger level. You can touch and move the trigger markers in the diagram to set the positions. The trigger point is the zero point of the diagram.

The trigger position can be moved outside the diagram. A red trigger position marker indicates that the trigger position is not visible.

**Trigger, Horizontal, Acquisition**
The "Trigger", "Horizontal" and "Acquisition" labels show the main timebase and trigger settings. If you tap a label, the relevant dialog box opens.

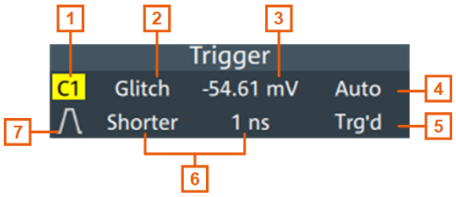



*Figure 3-4: Trigger label on the toolbar*

1 = Trigger source
2 = Trigger type
3 = Trigger level
4 = Trigger mode
5 = Trigger state
6 = Trigger slope

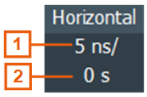



*Figure 3-5: Horizontal label on the toolbar*

1 = Time scale
2 = Horizontal position

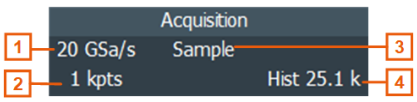



*Figure 3-6: Acquisition label on the toolbar*

1 = Sample rate
2 = Record length
3 = Decimation
4 = Number of acquired waveforms

**Reference point**
The reference point marks the rescaling center. If you modify the time scale, the reference point remains fixed on the screen, and the scale is stretched or compressed to both sides of the reference point.

You can define the position of the reference point ("Menu" > "Horizontal" > "Reference point"), and also its time distance from the trigger point of the diagram (Position knob).

**Notification**
The "Notification" button on the toolbar points to the status messages of the instrument. To open the message box, tap the button. See also: Chapter 3.3.10, "Instrument information and notifications", on page 66.

**Zoom diagram and zoom area**
Zoomed waveforms are shown in separate zoom diagrams, in addition to the waveform diagrams. On the original waveform diagram, a rectangle indicates the zoomed section of the waveform - this is the zoom area. You can modify the zoom area by dragging the rectangle as a whole, and by dragging its edges. To toggle between these modes, tap the zoom area. You can also set exact positions.

The frames of the zoom area and of the associated zoom diagram have the same color, different zooms are marked with different colors. So it is easy to assign zoom area and zoom diagram.

As for waveform diagrams, you can change the name of the zoom diagram. A zoom in a zoom and coupled zooms are also possible.

For details, see Chapter 7.1, "Zoom", on page 261.

**Signal bar**
The signal bar is the control center for all waveforms. All enabled waveforms are shown on the left side of the signal bar. On the right side of the signal bar, you see the signal activators of inactive waveforms. Tap a signal activator to enable the waveform.

Each waveform is represented by a signal icon. If the waveform is shown in a diagram, the signal icon displays its main vertical and acquisition settings. If you tap a signal icon, the dialog with vertical settings for this waveform opens. If you tap the "Minimize" icon on the signal icon, the waveform switches from the diagram area to the signal icon: the icon shows the real-time signal preview. See Chapter 3.3.4, "Working with waveforms", on page 51 for a detailed description.

In Figure 3-3, the signal icons C1 and C2 show the main settings, and the waveforms are displayed in diagrams. Other waveforms are minimized and shown as signal preview.

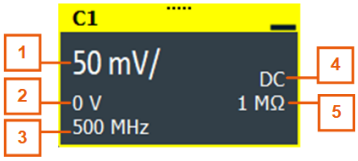



*Figure 3-7: Signal label*

1 = Vertical scale
2 = Offset
3 = Bandwidth
4 = Coupling
5 = Termination

If the signal bar contains many signal icons, and not all icons are visible, touch one of the icons and scroll left or right until the required icon appears.

You can also switch off the signal bar: "Menu" > "Settings" > "Appearance" > "Diagram" > "Show signal bar".

**Histogram and histogram area**
A histogram shows the frequency of occurrence of voltage or time values in a bar chart directly in the diagram. The rectangular histogram area indicates the part of the waveform that is considered in the histogram. The vertical histogram counts the voltage values, and the horizontal histogram counts time values. You can switch between vertical and horizontal mode, and modify the histogram area by dragging the rectangle as a whole, by dragging its edges, or by setting exact positions.

### 3.3.2.2 Control elements on the touchscreen

The touchscreen provides everything you need to control the instrument, to analyze waveforms, and to get measurement results. Figure 3-8 shows the control elements at a glance.

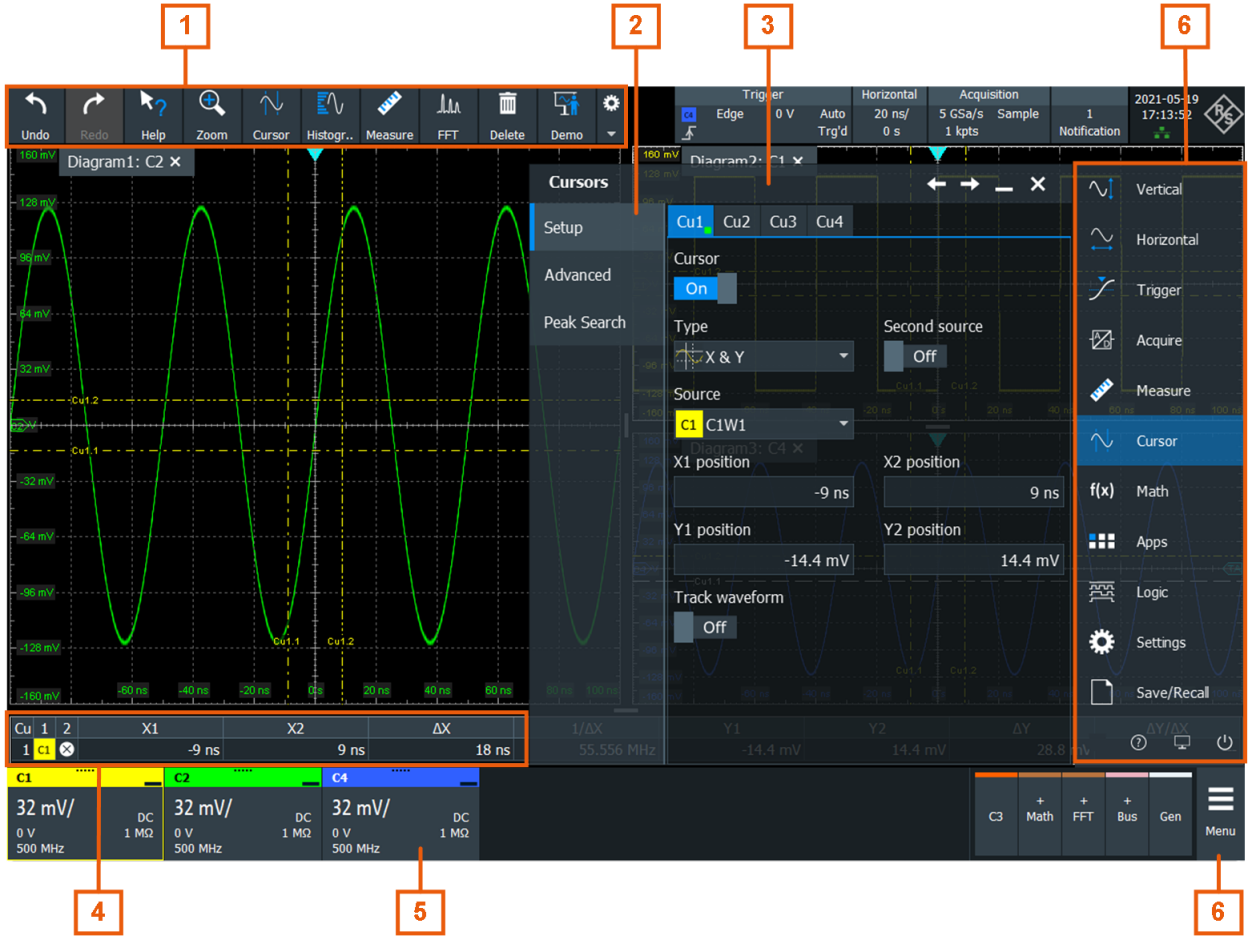



*Figure 3-8: Control elements on the touchscreen*

1 = Toolbar
2 = Tab in a dialog box
3 = Dialog box
4 = Result table (docked)
5 = Signal bar
6 = Menu
Not shown = Input box

### Toolbar (1)

The icons on the toolbar provide quick and easy access to the most important functionality. For a detailed description, refer to Chapter 3.3.6, "Toolbar", on page 55.

If you adjust the settings of an analyzing function, e.g., cursor measurement, the overlay menu is shown instead of the icons. The overlay menu provides the most important settings of the current action. If you need more settings, "Advanced Setup" opens the corresponding dialog box.

### Dialog box (2, 3)

The tabs of the dialog boxes contain all task-oriented settings and operations, and black buttons for calling related tabs. The usage of dialog boxes is described in Chapter 3.3.8, "Using dialogs", on page 63.

**Result table (4)**
If you perform cursor or automatic measurements, mask testing, or a search, the result table shows the results of the action. The position of the result table is adjustable. It can be docked (default for measurements), floating, minimized to a result icon on the signal bar, or displayed in a separate diagram on the screen.

See also: Chapter 3.3.7, "Displaying results", on page 61.

**Signal bar (5)**
The signal bar summarizes all waveforms as described in "Signal bar" on page 47.

**Menu (6)**
The menu provides access to the complete functionality of the R&S RTP.

**Input box**
The input box appears if you adjust a value using one of the rotary knobs, or if you drag an element on the screen, for example, a cursor line. The input box shows the current value of the modified parameter. You can enter the exact numerical value, change the step size, and - if available - autoset the value directly in the input box. The box title shows the name of the currently adjusted parameter.

See also: Chapter 3.3.9, "Entering data", on page 64.

### 3.3.3 Applications

All available applications for special tasks are provided in the "Apps" dialog.

► To open the "Apps" dialog:

- Open "Menu" > "Apps".
- Press the app key on the frontpanel.

### 3.3.4 Working with waveforms

The R&S RTP can create and display many waveform types. The most importand are:

- Channel waveforms:
  Up to three waveforms per input channel can be shown. For a four-channel instrument, 12 channel waveforms are available.
- Reference waveforms:
  Four waveforms can be used as reference for comparison and analysis.
- Math waveforms:
  Eight mathematic waveforms can be created with mathematic operations performed on channel, reference, and other math waveforms.
- Zoom waveforms:
  Show the details of waveforms.
- XY-waveforms:
  Four XY-waveforms can be created. Each XY-waveform is built from the voltage values of two source waveforms.
- Digital waveforms:
  The Mixed Signal Option R&S RTP-B1 provides 16 digital channels grouped in two logic probes (pods) with 8 channels each.

### Waveform handling

The R&S RTP can show and analyze many waveforms. To handle this multitude while keeping track of it, the R&S RTP provides intelligent support:

- The color system helps to distinguish the waveforms. The color of the vertical rotary knobs indicates the signal that is focused (selected). The color of each waveform can be changed, the color of its signal icon and of the illuminated keys is adjusted to the new color. Alternatively, a color table can be assigned to a waveform.
  Settings: "Menu" > "Settings" > "Appearance" > "Colors" tab.
- Waveforms can be minimized to signal icons showing a real-time signal preview. Thus, more space in the diagram area is available without switching off waveforms.
- Diagrams are displayed on tabs – you can arrange them side by side or one above the other. To change the diagram name, double-tap the tab name.
- The Rohde & Schwarz SmartGrid function helps to arrange the diagrams.

### Waveform states

Depending on its place on the screen and the effect of settings, a waveform has one of the following states:

- Off
- Active: The waveform is shown in a diagram.
- Selected:
  One of the active waveforms that has the focus. In each diagram, one of the assigned waveforms is selected – it appears "on top" in the diagram, and the grid labels have the color of the selected waveform. Some of the toolbar functions, like cursor and histogram measurements, are performed on the selected waveform. All waveform-specific settings are applied to the selected waveform of the selected diagram.
  The vertical [Position] and the [Scale] knobs are illuminated in the color of the selected waveform.
  In Figure 3-3, C1 is the selected waveform: The frame of the diagram and the signal icon are highlighted.
- Minimized:
  The waveform is shown as real-time signal preview in its signal icon.

### To switch a waveform on

A channel waveform is activated as soon as you connect the probe. You can switch it on and off according to your needs.

► Choose one of the following ways:

  - Press the channel key.
  - Tap the signal activator on the right side of the signal bar.
  - In the "Vertical" dialog box, select the channel and tap "Channel" > "On".

  The waveform is now active, selected, and is shown in the diagram.

Remote command: `CHANnel<m>:STATe` on page 1248

**To select a waveform**

► Choose one of the following ways:

- Tap the waveform in the waveform diagram.
- To select a channel, reference, or math waveform, press the corresponding key.
- Tap the signal icon.

**Note:** Zoom waveforms in zoom diagrams cannot be selected.

**To minimize a waveform**

► Choose on of the following ways:

- Tap the "Minimize" icon in the upper right corner of the waveform's signal label in the signal bar.
- Drag the waveform from the diagram to the signal bar.

The waveform disappears from the diagram and the signal preview is shown in the signal icon.

Remote command: `LAYout:SIGNal:UNASsign` on page 1215

► To set the waveform back to its previous diagram immediately, use "Undo".

**To arrange a waveform using the SmartGrid**

See Chapter 3.3.5, "Rohde & Schwarz SmartGrid", on page 53.

**To switch off a waveform**

► Do one of the following:

- To switch off a signal preview, tap the "Close" icon in the upper right corner of the minimized signal view.
- Disable "Channel" in the "Vertical" > "Setup" tab.
- Tap the "Delete" icon (Recycle bin) in the toolbar, and then the waveform.
  If several waveforms overlap or lie close together, the upper (selected) waveform is switched off.

Remote command: `CHANnel<m>:STATe` on page 1248

### 3.3.5 Rohde & Schwarz SmartGrid

The Rohde & Schwarz SmartGrid helps to create and arrange the diagrams on the screen with drag&drop. The diagram layout depends on the position where you drop the signal icon, in relation to an existing diagram.

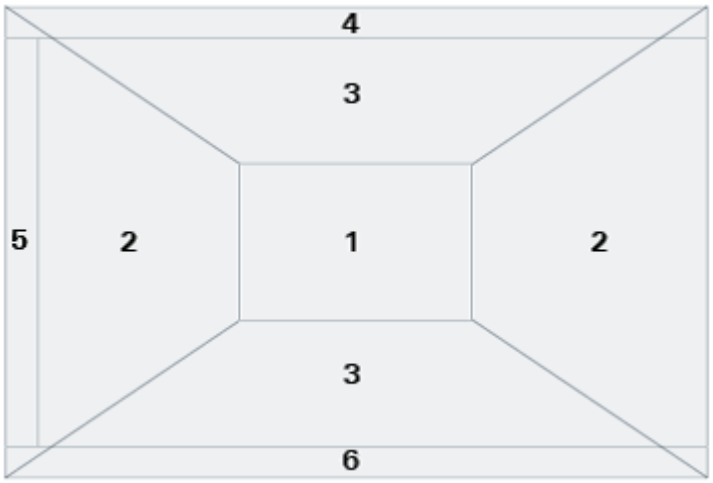



*Figure 3-9: SmartGrid positions*

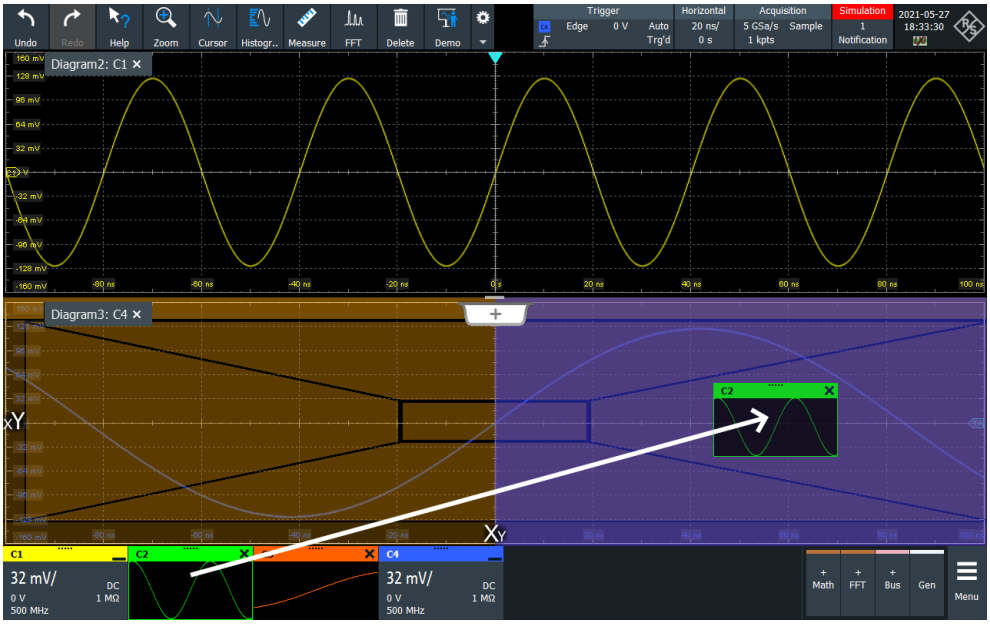
1 = In the existing diagram, overlay of signal
2 = New diagram on the left or right
3 = New diagram above or below
4 = New diagram on top of the existing diagram
5 = XY-diagram
6 = YX-diagram

The diagram configuration is deleted when you use [Preset] and `*RST`.

**To arrange a waveform using the SmartGrid**

You can arrange waveforms in one of the existing diagrams, or in a new diagram.

1. Drag the signal icon to the diagram area, and move it around.
   The Rohde & Schwarz SmartGrid appears and a blue area shows where the waveform will be placed.
2. Drop the waveform in the target area.
   The waveform appears in an existing or in a new diagram, and it is selected for further actions.

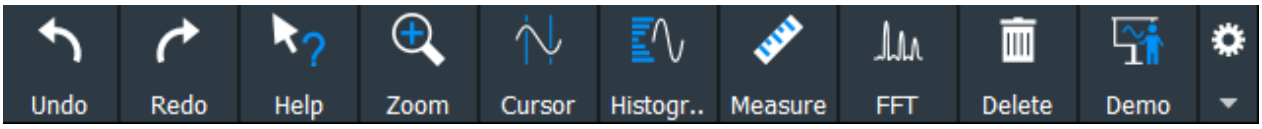



3. To change the size of a diagram, tap the handle between two diagram frames and drag it to the required position.

Remote commands: see Chapter 24.7.2.4, "Smartgrid", on page 1211.

## 3.3.6 Toolbar

The toolbar provides direct access to important control and measurement functions.




By default, the toolbar shows the most frequently used functions. You can configure the content of the toolbar, see Chapter 3.3.6.2, "Configuring the toolbar", on page 56.

### 3.3.6.1 Using the toolbar

Using the toolbar is easy and straightforward.

Some of the toolbar functions are one-click actions. These actions are performed immediately when you tap the icon.

Other toolbar functions are analyzing functions. These functions are interactive actions.

**To use analyzing functions (interactive actions)**

1. If several waveforms are shown in the diagram, select the waveform that you want to analyze.
   See: "To select a waveform" on page 53

2. Tap the icon of the function in the toolbar.
   If there are many functions on the toolbar, and not all icons are visible, scroll the toolbar using the arrow on the right end of the toolbar.
3. Check and adjust the settings in the overlay menu.

4. To define the analyzed area, do one of the following:
   - Tap the required diagram.
   - Drag a rectangle on the diagram.

### 3.3.6.2 Configuring the toolbar

You can configure the content of the toolbar so that only the required functions are displayed. The toolbar configuration is part of the user preferences. It is retained when you switch off and on the instrument, and you can save it in the user preferences and user-defined preset.

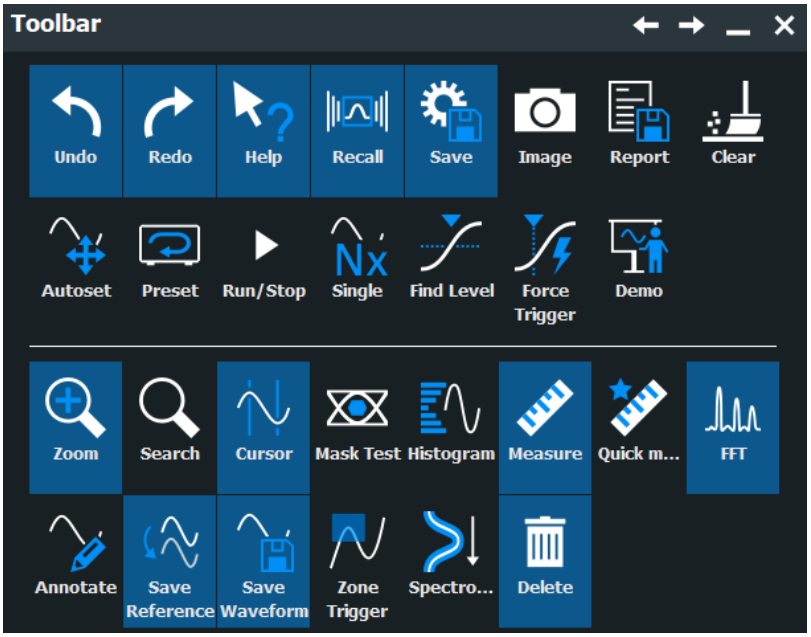
1. To open the toolbar configuration, tap the icon in the toolbar:

2. Select the required toolbar functions:
   - Disable the functions that you do not need.
   - Enable the functions that you want to add to the toolbar.
3. Close the dialog.

A detailed description of the toolbar functions is given in Chapter 3.3.6.3, "Toolbar functions", on page 57.

### 3.3.6.3 Toolbar functions

This section describes all toolbar functions in detail.




| One-click actions | Interactive actions |
|---|---|
| Undo | Zoom |
| Redo | Search |
| Help | Cursor |
| Recall | Mask Test |
| Save | Histogram |
| Image | Measure |
| Report | Quick meas |
| Clear | FFT |
| Autoset and Preset | Annotate |
| Run / Stop and Single | Save Reference |
| Find Level | Save Waveform |
| Force Trigger | Zone trigger |
| Demo | Spectrogram (option R&S RTP-K37) |

You can configure the content of the toolbar, see Chapter 3.3.6.2, "Configuring the toolbar", on page 56.

The following list describes at first the default toolbar functions and then the additional functions.

**Undo**
Undoes the last setting actions step by step. Some actions cannot be revoked: locking the touchscreen with [Touch Lock], and saving data. The undo stack is deleted during the following actions: Reloading settings from file, reference waveform actions (save, load and preset with active reference waveform), and some actions that require an internal re-alignment to ensure correct measurements.

**Redo**
Recovers the undo steps in reverse order.

**Help**
Enables the tooltip display. A short description appears when you tap a parameter in a dialog. To open the corresponding help topic, tap the "Show Help" button in the lower right corner of the tooltip. See also: Chapter 3.3.11, "Getting help", on page 67.

**Recall**
Opens a window to select and load instrument settings that were previously stored in a saveset. A graphical preview helps you to find the required settings.

**Save**
Saves the current instrument settings in a saveset.

You can reload the saveset using the "Recall" toolbar icon, or using "Menu" > "Save/Recall" > "Recall" tab > "Saveset with preview".

The filename is created according to the autonaming pattern, defined in "Menu" > "Settings" > "Save/Recall" > "Autonaming" tab.

**Image**
Saves a screenshot of the current display using the settings defined in "Menu" >"Save/Recall" > "Save" tab > "Screenshot".

**Report**
Creates a report of the current measurement settings and results using the settings defined in "Menu" > "Save/Recall" > "Save" tab > "Report".

**Clear**
Deletes all measurement results including long term measurement and statistics, all waveforms, and the history.

Remote command:
`DISPlay:CLR` on page 1231

**Autoset and Preset**
Performs an autoset, or a preset to a default state. The icons have the same functionality as the corresponding keys on the front panel. They are useful when you operate the instrument remotely.

**Run / Stop and Single**
Starts and stops the continuous acquisition, or starts a defined number of acquisition cycles. The icons have the same functionality as the corresponding keys on the front panel. They are useful when you control the instrument remotely.

**Find Level**
Analyses the signal and sets the trigger level to the middle of the signal peaks.

**Force Trigger**
Starts an immediate single acquisition. If the acquisition is running in normal mode and no valid trigger occurs, use "Force Trigger" to confirm that a signal is available. Then you can use the displayed waveform to determine how to trigger on it.

**Demo**
Opens the "Demo" dialog, where you can find examples of operating the R&S RTP.

**Zoom**
Creates a zoom. Tapping the icon opens the overlay menu, where you select one of the four zoom types: standard zoom, hardware zoom, coupled zoom, fingertip zoom.

**Standard zoom ← Zoom**
Displays a magnified section of the diagram in an additional zoom diagram. It is a display zoom, instrument settings are not changed.

Touch and hold the zoom area to open the "Zoom" dialog box.

Remote command:
`LAYout:ZOOM:ADD` on page 1379

**Hardware zoom ← Zoom**
Changes the instrument settings - horizontal and vertical scales as well as trigger level and offset - to display a part of the diagram in greater detail.

**Coupled zoom ← Zoom**
Creates a coupled zoom area and its related zoom diagram. If you change the size of one zoom area, the size of all coupled zoom areas is changed as well.

Remote command:
`LAYout:ZOOM:ADDCoupled` on page 1379

**Fingertip zoom ← Zoom**
Magnifies the waveforms around your fingertip.

Tap the icon and put your finger on the waveform. The touched part of the waveform is displayed in a magnifier. Drag your finger on the screen to move the magnifier. You can change the zoom factor using the [Multiuse] knob.

**Search**
Performs a search according to the settings in the "Search Setup" dialog box. Tap the icon and then tap the diagram with the waveform to be searched, or drag a rectangle to define a search gate. The search is performed on the selected waveform.

**Cursor**
Performs a cursor measurement. Tapping the icon opens the overlay menu, where you select the cursor type and source.

Tap the diagram where you want to set the cursors, or draw a rectangle in the diagram to position the cursor lines. The resulting cursor lines measure the selected waveform, and the results are shown.

**Mask Test**
Starts the on-screen mask definition and the testing against the defined mask.

Tap the icon and then tap the points that build the mask. To finish the shape definition, tap "Finish segment" in the overlay menu. To create a rectangular mask, draw a rectangle on the screen. You can add more segments to the test, and add another test. You can also move the mask segments on the screen.

To configure the mask test settings, tap the icon in the "Mask" result box.

**Histogram**
Creates a histogram. Tapping the icon opens the overlay menu, where you select the histogram type and source.

Drag a rectangle on the diagram to define the histogram area. The histogram for the selected waveform appears.

To change settings, touch and hold the histogram area, and the histogram overlay menu opens.

**Measure**
Performs automatic measurements. Tapping the icon opens the overlay menu, where you select the category, the measurement types and measured waveform. Tap the diagram with the waveform to be measured. To define a measurement gate, draw a rectangle on the screen.

You can run up to 8 automatic measurement groups in parallel. The "Measure" icon starts the measurement groups one after the other. A measurement group consists of minimum one measurement, and can consist of many measurements of the same category.

To modify the measurement, double-tap one of the result values.

**Quick meas**
Performs a set of measurements on the selected waveform or on the selected gate. You can configure up to 8 measurements to be included in quick measurement.

Tap the icon and then tap the waveform to be measured or draw a rectangle to define the gate.

**FFT**
Transforms a waveform to the frequency spectrum by fast Fourier transform (FFT). The FFT trace is shown in a new diagram.

Tap the icon and adjust the settings in the overlay menu. Select the source of the spectrum in the overlay menu. The FFT diagram is created from the selected waveform.

To adjust FFT settings, double-tap the FFT diagram.

**Annotate**
Defines a waveform label that names or explains the waveform. Tap the icon and then tap the waveform to be labeled. Enter the label text using the on-screen keyboard. The text is shown in the same color as the waveform. If you tap the display background, the label is assigned to the focused waveform. You can drag the label to another position.

**Save Reference**
Copies the selected source waveform with all its settings to the reference waveform. If the acquisition is running, the reference waveform is a snapshot. You can configure up to four reference waveforms.

Tap the icon, and select the required reference waveform (R1 to R4) in the overlay menu. Tap the waveform to be copied.

**Save Waveform**
Exports the waveform data to file using the settings defined in "Menu" > "Save/Recall" > "Save" tab > "Waveform" > "Setup" tab.

The filename is created according to the autonaming pattern.

Tap the icon and then tap the waveform to be exported. If you tap the display background, the focused waveform is exported, or a multichannel export is performed if configured.

**Zone trigger**
Defines a zone trigger, which combines the trigger condition with the intersection or non-intersection of one or more zones or masks.

Tap the icon and then tap the corner points of the zone on the screen. Tap "Finish zone". You can add more zones to the trigger condition.

**Spectrogram (option R&S RTP-K37)**
Starts an FFT and the spectrogram. The FFT trace and the spectrogram are shown in separate diagrams.

Tap the icon. Select the source in the sidebar, or tap diagram with the waveform to be transformed. The diagrams are created from the selected waveform.

**Delete**
Removes waveforms, measurements, zoom, histograms, mask segments and other elements from the display.

Tap the icon and then tap the element to be deleted, or the waveform to be switched off.

## 3.3.7 Displaying results

The results of measurements, mask tests, searches, protocol decoding and others are displayed immediately. There are several places to display the results:

- In a table that is docked below the diagram (only for measurement and decode results)
- In a floating result box in front of the diagrams, which you can move on the display

- In a minimized view (result icon) on the signal bar
- In a separate tab that can be arranged similar to a diagram

The default position and the font size can be adjusted.

### To define the default position of measurement results

For results of automatic, cursor and protocol measurements and for decode results, the docked table below the diagram is the initial default position.

To change the default position of the results:

► Open the "Menu" > "Settings" > "Appearance" dialog.

- Automatic measurements: select "Measurement" tab > "Result position".
- Cursor measurements: select "Cursor" tab > "Result position".
- Protocols: select "Protocol" tab > "Result position".

Mask test and search results are always shown in a floating result box, which you can drag to a convenient position using the SmartGrid.

### To arrange the results on the display

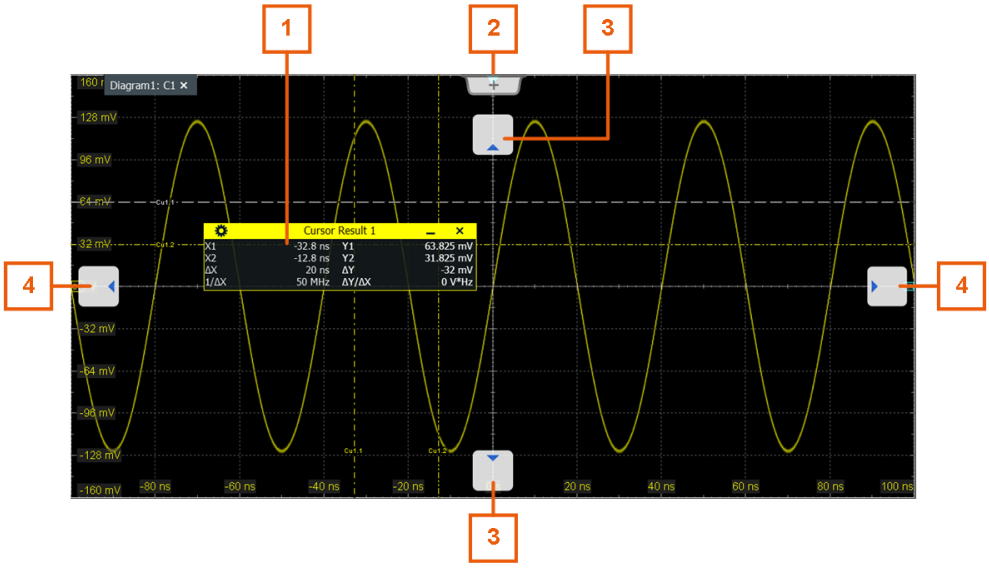
1. If the results are listed in a docked table, touch and hold a result line and drag it.

   The results are now shown in a floating result box.

   **Note:** Once you undock the measurement results, you cannot move the result box back to the docked position.

2. Touch and hold the title of the result box and drag the box on the screen. The SmartGrid indicates where the result box will be placed.

   - If you drop the box on one of the buttons, the results are shown in a separate tab besides, above, or below the diagram.
   - If you drop the box on the signal bar, a result icon is created.
   - If you drop the box somewhere else, the floating result box is placed there.



1 = Floating result box
2 = New tab
3 = Table in a tab above or below
4 = Table in a tab on the left or right

**To open the corresponding settings**

1. Double-tap one of the result values.
   If a result box is shown, you can also tap the ⚙ icon.

   The overlay menu appears and provides the most important settings.

2. In the overlay menu, tap "Advanced Setup".

   The corresponding dialog opens.

**To adjust the font size of results**

1. Open the "Menu" > "Settings" > "Appearance" dialog.
2. Select the "Dialogs" tab.
3. Set the "Result dialog" > "Font size".

## 3.3.8 Using dialogs

Dialogs provide groups of related settings and allow to make selections and enter data in an organized way.

**To open a dialog**

► Perform one of the following actions:

- Open the "Menu", and select the menu entry.
- Press the function key on the front panel.
- If the signal bar is open, tap "Advanced Setup".
- To open the "Vertical" dialog of a waveform, tap the signal icon.
- Tap the "Horizontal", "Acquistion" or "Trigger" label to open the corresponding dialog.

For direct access to important control and measurement functions, use the toolbar, see Chapter 3.3.6, "Toolbar", on page 55.

Each dialog has four icons in the upper right corner:

| | |
|---|---|
| | Go back: opens the previously opened dialog box. |
| | Go forward: opens the next dialog box. |

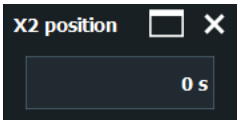
| | |
|---|---|
| | Minimizes the dialog box to a small box that only contains the last selected function. |
| | Closes the dialog box. |

**To minimize a dialog**

If you want to change only one setting during analysis, and you need to change it often, you can display a small box that only contains the required setting.

1. Open the dialog.
2. Tap the setting that you need to modify.
3. Tap the "Minimize" icon in the upper right corner of the dialog box.

   The dialog box turns into a small box that contains only the setting, that was last selected.

4. To restore the complete dialog box, tap the "Maximize" icon in the small box.

## 3.3.9 Entering data

To set parameter values and enter other data, you use the various knobs and the on-screen keypad or keyboard.

**Using scale, position and level knobs**

The instrument has dedicated rotary knobs to set vertical and horizontal positions and scales, and the trigger level. When you turn a knob, the input box opens in the upper right corner of the screen, showing the parameter name and current value.

1. Turn the knob to change the value.
2. Press the knob:
   - [Scale]: toggles the increment.
   - [Position]: sets to zero.
   - [Level]: sets the trigger level to 50% of the signal.
3. The input box also has settings:
   - Tap the "Steps" icon to toggle the increment.
   - Tap the "Reset" icon to set the parameter to the autoset value (if available).

**To enter values with the on-screen keypad**

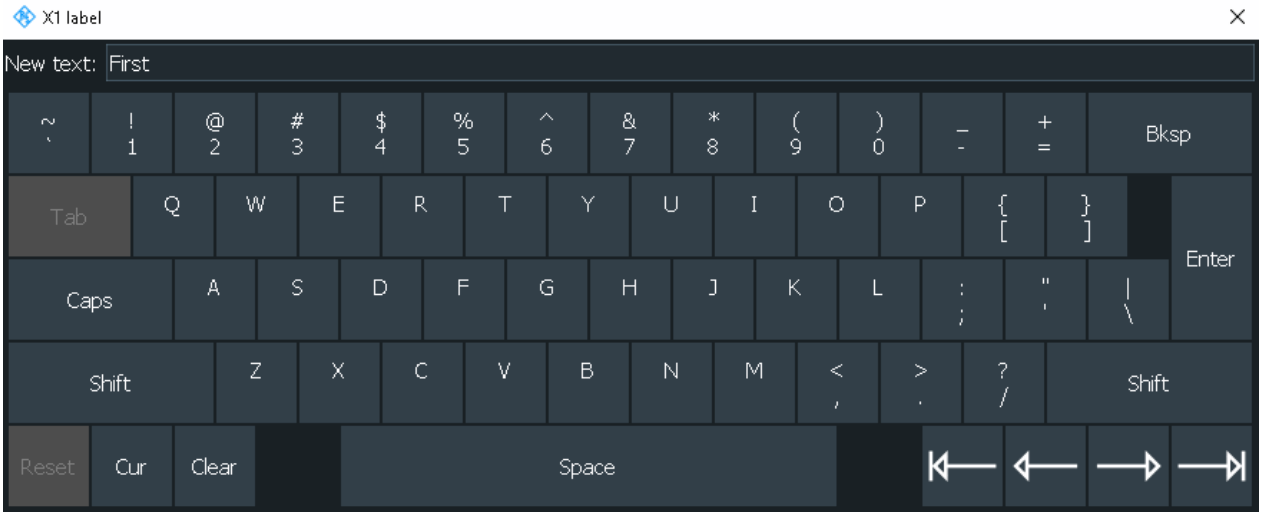
To enter numeric values and units in dialogs, the touchscreen provides an on-screen keypad. For text input, the on-screen keyboard with English key layout is used.

1. Double-tap the entry field.
   The on-screen keypad opens.

2. Enter the numeric value using one of the following methods:
   - To use the default value, tap "Reset" (if available).
   - To use the minimum or maximum value, tap "Min" or "Max", respectively.
   - To increase the displayed value in fixed steps, tap "Inc".
     To decrease the value in fixed steps, tap "Dec".
     To toggle between small steps and large steps, tap the "Steps" icon.
   - To get the value that was used before the keypad opened, tap "Cur".
   - To enter a user-defined value, tap the numbers and complete the entry by tapping the unit button.
     – The arrow buttons move the cursor left or right.
     – "Bksp" deletes the last character before the cursor.
     – "±" changes the sign of the value.

**To enter data with the on-screen keyboard**

1. Double-tap the entry field to open the on-screen keyboard.
   If available, you can also tap the keyboard icon on the right of the entry field.

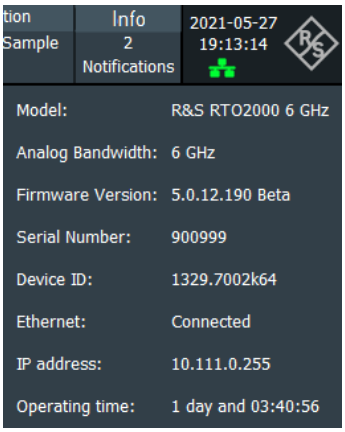



2. Enter the text as you would do on a normal keyboard.
   - To enter a series of capital characters, tap "Caps".
     To enter one capital character, tap "Shift".
   - To get the value that was used before the keyboard opened, tap "Cur".
   - The arrow buttons move the cursor left or right.
   - "Bksp" deletes the last character before the cursor.
3. Tap "Enter" to complete the entry.

### 3.3.10 Instrument information and notifications

In the upper right corner of the screen, you see the Rohde & Schwarz logo, date and time, the symbolic information on LAN connection and the notifications status.

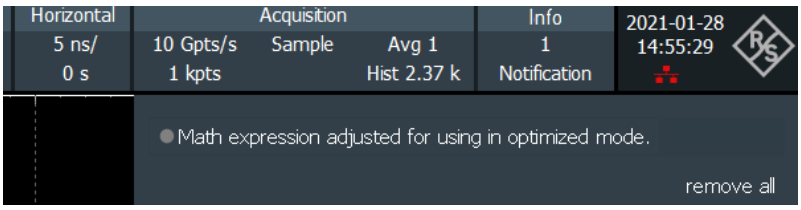
► To see the instrument information, tap the Rohde & Schwarz logo.




Notifications are status messages, information on mismatching settings and similar information. They are displayed for a few seconds and saved.

The color of the dot before the text indicates the severity: gray for information, orange for warnings, and red for errors.

► To read the notifications, tap "Notification".
You can also delete the list of notifications.




1. To hide the date and time, open "Menu" > "Settings" > "Appearance" > "Diagram" tab.
2. Disable "Show date / time".

## 3.3.11 Getting help

If you need information on the instrument's functionality, you can use the following sources:

- Tooltips give a short description of the parameter.
- The context help provides functional description on a setting, and the corresponding remote command.
- The general help opens the help window with a table of contents, where you can browse and search for information.
- Videos explain various analyzing and measurement tasks.
They are available in the R&S®RTP - Media Center.

### 3.3.11.1 Displaying help

**To display tooltips and context help**

1. Enable the "Help" icon on the toolbar.

2. Tap the parameter for which you need information.

   The tooltip opens.

3. To open the corresponding help topic, tap the "Show Help" button in the lower right corner of the tooltip.

   The "Help" window opens and displays the comprehensive description and the corresponding remote command. You can browse the help for further information.

**Note:** The "Help" icon disables automatically when you tap a parameter. To show another tooltip, tap the "Help" icon again.

**To open general help**

1. Open the "Menu".
2. Tap the "Help" icon.

   The help window opens with the "Contents" page, where you can select the topics.

### 3.3.11.2 Using the Help window

The Help window contains several tabs:

- "View" - shows the selected help topic
- "Contents" - contains a table of help contents
- "Index" - contains index entries to search for help topics
- "Search" - provides text search

The Help toolbar provides some buttons:

- To browse the topics in the order of the table of contents: up arrow = previous topic, down arrow = next topic
- To browse the topics visited before: left arrow = back, right arrow = forward
- To increase or decrease the font: magnifier with + or - sign, respectively

To navigate the Help, use the touchscreen.

**To search for a topic in the index**

The index is sorted alphabetically. You can browse the list, or search for entries in the list.

1. Switch to the "Index" tab.
2. Select the "Keyboard" icon besides the entry field.
3. Enter the first characters of the keyword you are interested in.

   The entries containing these characters are displayed.
4. Double-tap the suitable index entry.

   The "View" tab with the corresponding help topic is displayed.

**To search topics for a text string**

1. Switch to the "Search" tab.
2. Select the "Keyboard" icon besides the entry field.

3. Enter the string you want to find.
   If you enter several strings with blanks between, topics containing all words are found (same as AND operator).

   For advanced search, consider the following:
   - To find a defined string of several words, enclose it in quotation marks. For example, a search for *"trigger qualification"* finds all topics with exactly *"trigger qualification"*. A search for *trigger qualification* finds all topics that contain the words *trigger* and *qualification*.
   - Use "Match whole word" and "Match case" to refine the search.
   - Use operators AND, OR, and NOT.

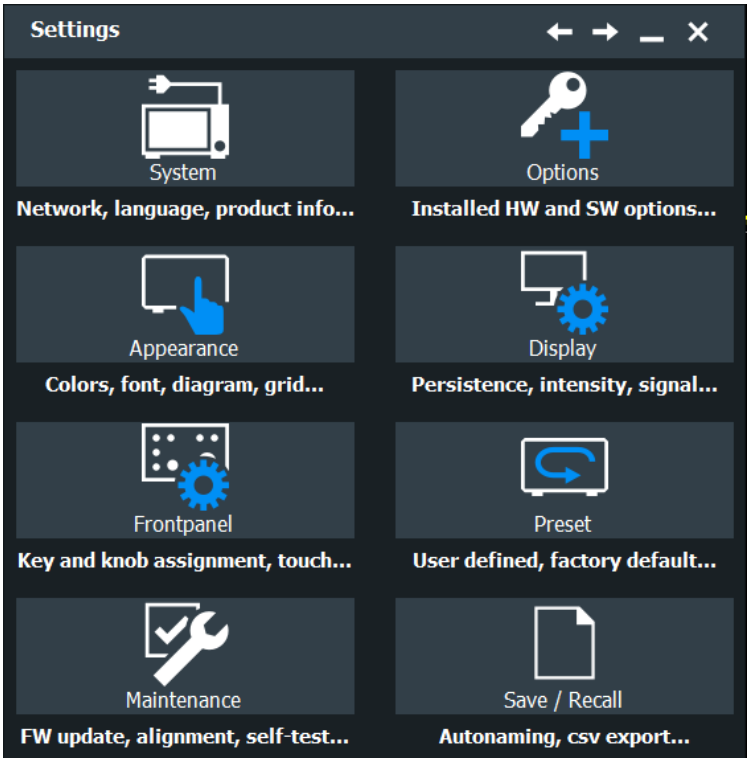
**To close the Help window**

► Select the "Close" icon in the upper right corner of the help window.

# 4 Instrument setup

Access: "Settings" menu > "Settings" dialog.

In the "Settings" dialog, you can adapt various instrument settings to your requirements, such as language, display appearance, and assign functions to some keys.




The following settings and procedures are described in the current chapter:

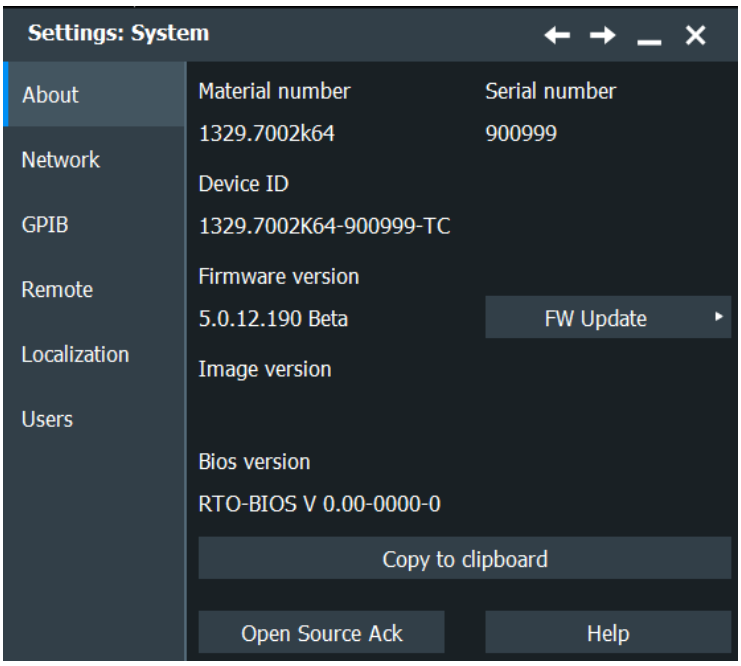
- System settings....70
- Option settings....74
- Appearance settings....77
- Display settings....93
- Frontpanel settings....97
- Preset setup....101
- Maintenance settings....103
- Save / recall....107

## 4.1 System settings

In the "Settings" > "System" dialog box, you find all instrument, firmware and network related information. Here you can also set the language that is used in the dialogs.

### 4.1.1 About settings

Access: "Menu" > "Settings" > "System" > "About"




**FW Update**
Opens the Firmware update dialog box.

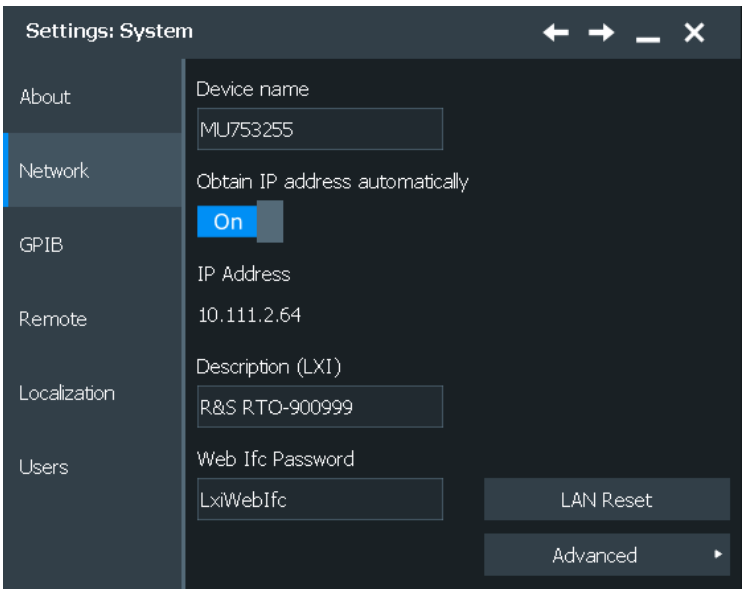
**Copy to clipboard**
Copies the instrument information, which you see in the dialog box, to the clipboard.

**Open Source Ack**
Opens the open source acknowledgment document, which provides verbatim license texts of the used open source software.

**Help**
Opens the help window with a table of contents, where you can browse and search for information.

### 4.1.2 Network settings

Access: "Menu" > "Settings" > "System" > "Network"




**Device name**
Indicates the currently defined computer name, the defined IP address and DHCP address enabling. These values are required to configure the instrument for work in a network.

**NOTICE!** Risk of network problems. All parameters can be edited here; however, beware that changing the device name has major effects in a network. For details, see Chapter 23.2, "Setting up a network (LAN) connection", on page 1137.

Remote command:
`DIAGnostic:SERVice:COMPutername` on page 1206

**Obtain IP address automatically**
If enabled, the IP address of the oscilloscope is obatained automatically.

**IP Address**
Indicates the currently defined computer name, the defined IP address and DHCP address enabling. These values are required to configure the instrument for work in a network.

**Description (LXI)**
Instrument description of the R&S RTP.

**Web Ifc Password**
Password for LAN configuration. The default password is *LxiWebIfc*.

**LAN Reset**
Resets the LAN configuration to its default settings using the network configuration reset mechanism (LCI) for the instrument. The following parameters are reset:

| Parameter | Value |
|---|---|
| TCP/IP mode | DHCP + auto IP address |
| Dynamic DNS | Enabled |
| ICMP ping | Enabled |
| Password for LAN configuration | LxiWebIfc |

The LAN settings are configured using the instrument's web browser.

**Advanced**
Opens the standard Windows "Network Connections" dialog box to configure a network connection. Only users with administrator rights can fulfill this task.

### 4.1.3 GPIB settings

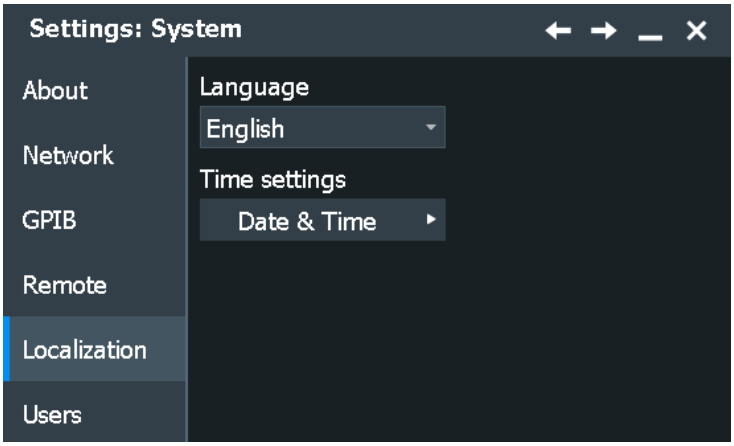
For a description of the GPIB settings, see: Chapter 23.5.2.2, "Remote settings", on page 1152.

### 4.1.4 Remote settings

For a description of the remote settings, see: Chapter 23.5.2.2, "Remote settings", on page 1152.

### 4.1.5 Localization settings

Access: "Menu" > "Settings" > "System" > "Localization"




**Language**
Selects the language in which the dialog boxes, result boxes and other screen information is displayed.

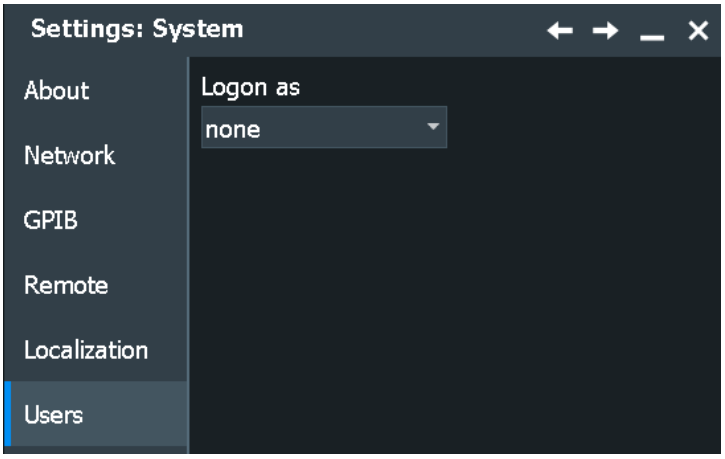
**Time settings**
Opens the standard Windows "Date and Time Properties" dialog box to set the correct date and time. Only users with administrator rights can fulfill this task.

**Note:** Usually date and time are set correctly. To adjust your regional time, select the correct time zone rather than changing the time.

Remote command:
`SYSTem:DATE` on page 1205
`SYSTem:TIME?` on page 1206

### 4.1.6 Users settings

Access: "Menu" > "Settings" > "System" > "Users"




**Log on as**
Sets the user that is automatically logged on during the startup process of the instrument. The change of this setting takes effect at the next instrument startup.

See Chapter 23.1.1, "Logon", on page 1133 for restrictions of the standard user and how to change the auto-logon.

| | |
|---|---|
| "User autologon" | Auto-logon as standard user with limited access. Enter the "User name": *NormalUser* and the "Password" of the standard user. |
| "Admin autologon" | Auto-logon with unrestricted access to the instrument and network. The setting is only available for administrators. Enter the "User name": *Instrumen* and the administrator's "Password" to enable the auto-logon. |
| "None" | No auto-logon, user name and password are requested at instrument startup. |

## 4.2 Option settings

Additional options for the R&S RTP are enabled using a license key. To obtain the license key, consult your sales representative.

The license type defines the duration of applicability and the portability of a license. The following license types are provided: evaluation, permanent, portable, quantified, timed with duration of 1, 3, 6 or 12 months. A license can also be in the states deactivated and expired.

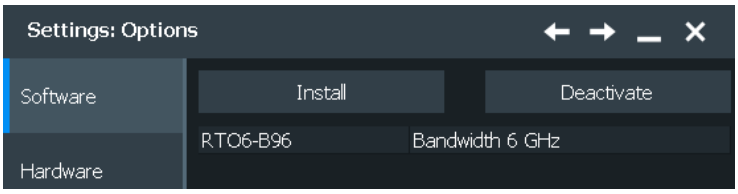
**Unregistered licenses**

Unregistered licenses are not assigned to a particular instrument. The instrument accepts only registered licenses. If your license is delivered unregistered, use the online tool R&S License Manager to register the license for your instrument. The registration of a permanent license is irreversible, so ensure that you register it for the correct instrument. The address of the tool is https://extranet.rohde-schwarz.com/service. For registration, you need the device ID of the instrument on which the option will be installed.

The R&S License Manager also allows you to move a portable license to another instrument.

### 4.2.1 Software options settings

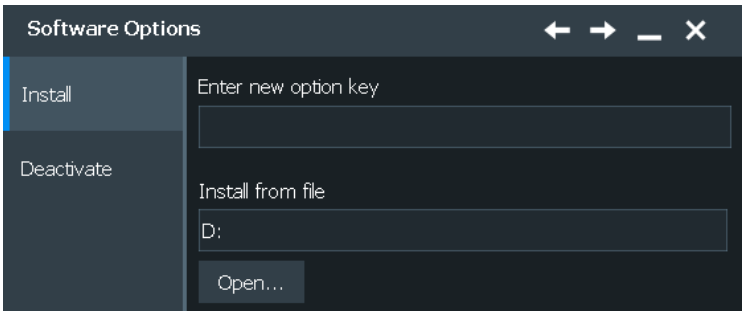
Access: "Menu" > "Settings" > "Options" > "Software"




In this dialog, you can access settings for installing and deactivating options.

#### 4.2.1.1 Install options

Access: "Menu" > "Settings" > "Options" > "Software" > "Install"




In the "Install" tab, you can install new options or deactivate existing options using license keys.

**Enter new option key**

Enter the license key here to activate the option. For license keys delivered as a file, use Install from file. Only users with administrator rights can activate options.

**Install from file**
If you got a license file, install the license here.

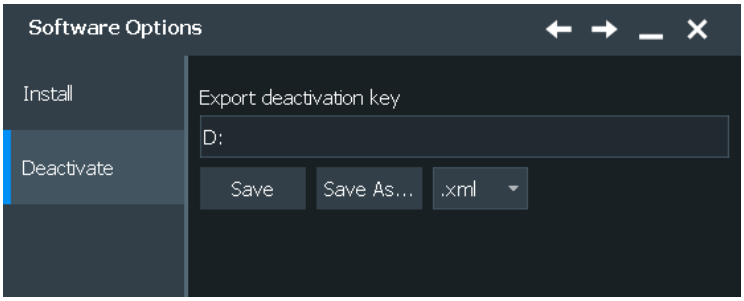
Tap "Open" to open the file selection dialog, or enter the complete path and filename. For details, see Chapter 12.5, "File selection dialog", on page 513. Only users with administrator rights can activate options.

When you move a portable license, use this function to import the deactivation key that is generated by the "R&S License Manager". See also Chapter 4.2.1.3, "Using a license server", on page 76.

### 4.2.1.2 Deactivate options

Access: "Menu" > "Settings" > "Options" > "Software" > "Deactivate"

In the "Deactivate" tab, you can export the deactivation response.




**Export deactivation key**
When you move a portable license, or deactivate an option, you have to note the response key, or to save the response to a file. The "R&S License Manager" needs the response key.

### 4.2.1.3 Using a license server

Software licenses can also be provided by the R&S®License Server. In this case, all available licenses are stored on a specified server. When you need a particular application, you can obtain a license from the server. When you no longer need the license, you return it to the server and it becomes available to other users again. These licenses are also referred to as *floating licenses*, as opposed to permanent licenses.

The R&S®License Server must be set up as described in the R&S®License Server - Managing Floating Licenses - User Manual.

You can find the manual in the Windows menu of your instrument: "Start > All Programs > R&S License Server > Open License Server Manual".

**To open the License Server**

► In the Windows notification area, select , and then "Open Manager".

a) Optional: If the status icon ( , or ) in the Windows notification area is not shown yet, click "Start > All Programs > R&S License Server > License Server Manager"".
The icon is now available in the Windows notification area.
b) Optional: If the status is (stopped), select the icon, and then select "Start License Server". Then, select > again.

The license server opens in the default web browser. The program adds all available Rohde & Schwarz products (smart cards or devices) to a list and automatically shows the first product that is detected. By default, the "Licenses" list opens with an initial filtering for showing only active licenses.

### 4.2.2 HW options settings

This tab informs about the availability of hardware options.

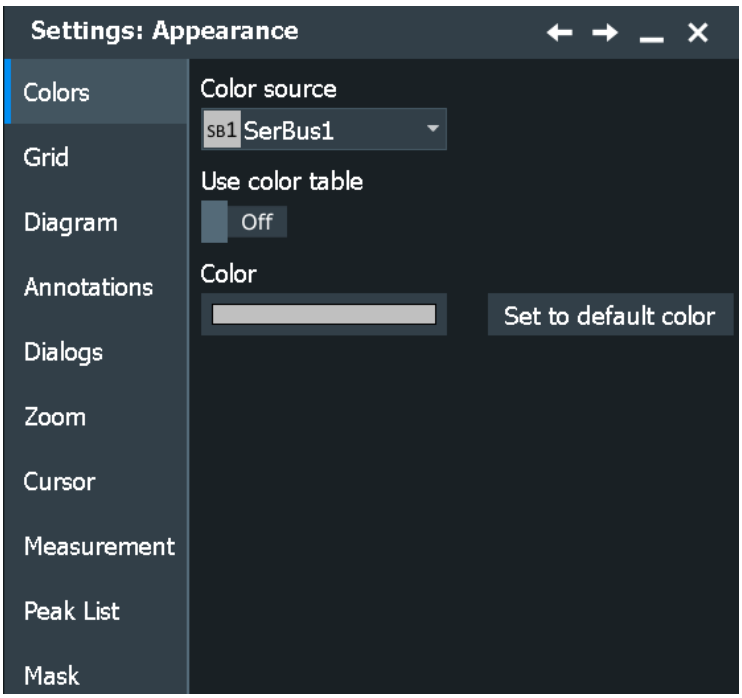
## 4.3 Appearance settings

In the "Settings" > "Appearance" dialog box, you define the look and feel of the display element, e.g. waveform colors, result position, or grid behavior.

### 4.3.1 Colors settings

Access: "Settings" > "Appearance" > "Colors"

In this dialog, you can define the colors to be used for the different waveforms.




**Color source**
Selects the waveform to which the color table and the labels are assigned.

A spectrogram (option R&S RTP-K37) always has the same color as the math (spectrum) waveform from which it is created.

**Use color table**
If enabled, the selected waveform is displayed according to its assigned color table.

If this option is disabled, the selected color is displayed, and the intensity of the specific signal color varies according to the cumulative occurrence of the values.

Remote command:
`DISPlay:COLor:SIGNal<m>:USE` on page 1207

**Color**
Shows the current color of the selected waveform. To change the color, tap the button and select a color. The color of the waveform, its signal icon, channel icon, and of the illuminated keys is adjusted to the new color.

Remote command:
`DISPlay:COLor:SIGNal<m>:COLor` on page 1207

**Assigned color table**
Adjust the waveform colors to suit your preferences. For each of the following waveform types you can assign a suitable color table:

- Analog and digital channels
- Reference waveforms
- Results of a mathematical function, also for FFT and derived spectrogram.
- Measurements and tracks
- XY-traces

- Serial buses if a protocol option is activated
- Parallel buses if MSO option is installed

See also: Chapter 4.3.2, "Color tables", on page 79

Remote command:
`DISPlay:COLor:SIGNal<m>:ASSign` on page 1207

**Set to default color**
Resets the color of the selected waveform to the factory default.

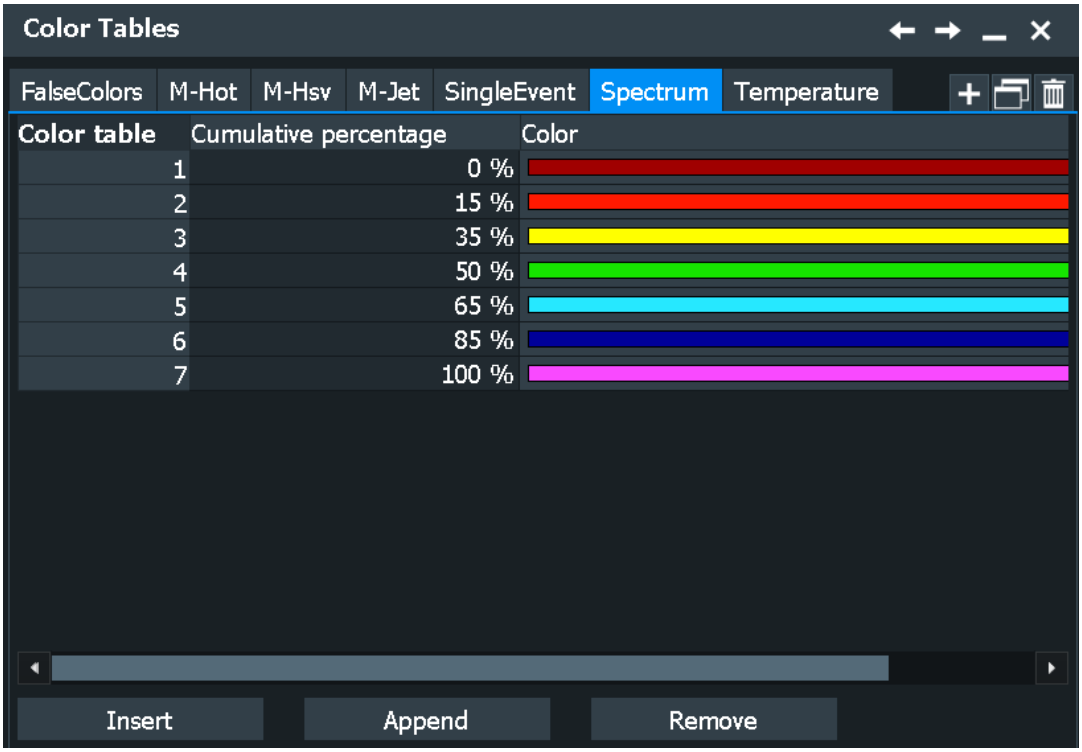
## 4.3.2 Color tables

Access: "Settings" > "Appearance" > "Color" > "Edit Color Table"

Color tables define the color of the waveform pixels depending on the cumulative occurrence of the associated values. By default, the intensity of the specific waveform color varies according to the cumulative occurrence of the values. The more often a value occurs, the darker the color of the data point is displayed.

The following color tables are provided:

- "False colors": color changes gradually in a wide color spectrum.
- "M-Hot"
- "M-Hsv"
- "M-Jet"
- "Spectrum": colors display the wave lengths of the light. Low cumulative occurrence is displayed blue like high wave length.
- "Single Event": single events and very seldom events appear yellow, a higher cumulative occurrence is shown with blue color. This view helps to identify specific events.
- "Temperature": color changes gradually from blue (low temperature) to red (high temperature) with increasing cumulative occurrence.




The editing table allows you to edit existing color tables or add new ones that can then be assigned to the waveforms.

See also:

- "Assigned color table" on page 78

**Remote commands**

The following remote commands are used to configure color tables:

`DISPlay:COLor:PALette:COUNt?` on page 1208

`DISPlay:COLor:PALette:ADD` on page 1208

`DISPlay:COLor:PALette:REMove` on page 1208

`DISPlay:COLor:PALette:POINt:INSert` on page 1208

`DISPlay:COLor:PALette:POINt:ADD` on page 1208

`DISPlay:COLor:PALette:POINt[:VALue]` on page 1209

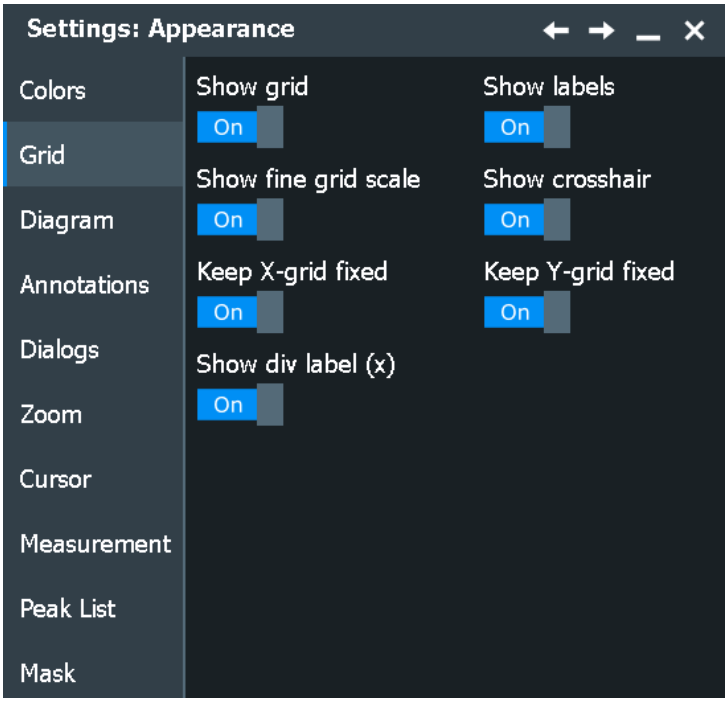
`DISPlay:COLor:PALette:POINt:COUNt?` on page 1209

`DISPlay:COLor:PALette:POINt:REMove` on page 1209

`DISPlay:COLor:PALette:COUNt?` on page 1208

### 4.3.3 Grid appearance settings

Access: "Settings" > "Appearance" > "Grid"



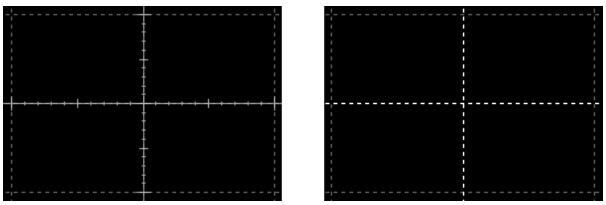

**Show grid**
If selected, a grid is displayed in the diagram area. A grid helps you associate a specific data point to its exact value on the x- or y-axis.

Remote command:
`DISPlay:DIAGram:GRID` on page 1210

**Show labels**
If selected, labels mark values on the x- and y-axes in specified intervals in the diagram.

Remote command:
`DISPlay:DIAGram:LABels` on page 1210

**Show fine grid scale**
If selected, the crosshair is displayed as a ruler with scale markers. If disabled, the crosshair is shown as dashed lines.



Remote command:
`DISPlay:DIAGram:FINegrid` on page 1210

**Show crosshair**
If selected, a crosshair is displayed in the diagram area. A crosshair allows you to select a specific data point by its co-ordinates.

Remote command:
`DISPlay:DIAGram:CROSshair` on page 1210

**Keep X-grid fixed**
If enabled, the vertical grid lines remain in their position when the horizontal position is changed. Only the values at the grid lines are adapted.

Remote command:
`DISPlay:DIAGram:XFIXed` on page 1211

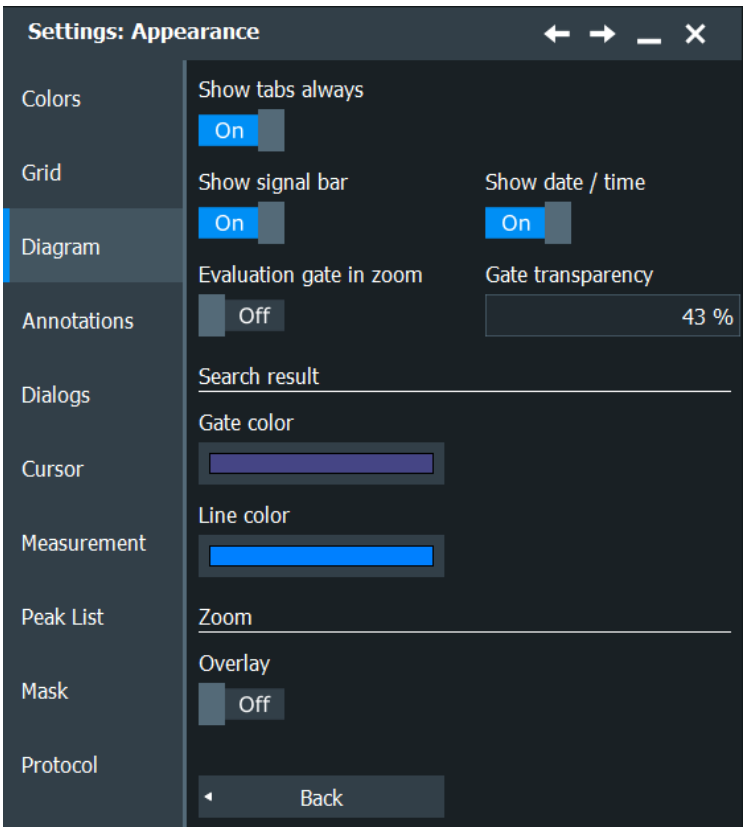
**Keep Y-grid fixed**
If enabled, the horizontal grid lines remain in their position when the position of the curve is changed. Only the values at the grid lines are adapted. Fixed horizontal grid lines correspond to the behavior of traditional oscilloscopes.

Remote command:
`DISPlay:DIAGram:YFIXed` on page 1211

**Show div label (x)**
If selected, the time scale value is shown at the diagram bottom instead of the horizontal grid labels. For example, 10 ns/div is shown instead of the values 0, 10, 20, 30... ns.

### 4.3.4 Diagram appearance settings

Access: "Settings" > "Appearance" > "Diagram" tab




**Show tabs always**
If selected, the tab titles of all diagrams are displayed: "Diagram1", "Diagram2" ...

If cleared, the tab titles are not shown except for titles in a tabbed diagram. In tabbed diagrams, the tab titles are required to change the tabs.

Remote command:
`DISPlay:DIAGram:TITLe` on page 1215

**Show signal bar**
If enabled, the signal bar is displayed on the right of the diagram area.

The signal bar contains signal icons that display either real-time views of minimized waveforms, or the main settings of displayed waveforms. On the top of the bar, the timebase label and trigger label provide general information for all displayed channels.

Remote command:
`DISPlay:SIGBar[:STATe]` on page 1215

**Show date / time**
If enabled, the date and time are shown near to the Rohde & Schwarz logo at the upper right corner of the screen.

**Evaluation gate in zoom**
If enabled, the available histogram areas, masks, and measurement gates are shown in the zoom diagrams. If the evaluation gate is within the zoom area, the display helps to move or modify the evaluation gates in the zoom window.

Make sure that the option is disabled if the zoom area and the evaluation gate are of nearly the same size to avoid conflicts in operation.

**Gate transparency**
Sets the transparency of the area that is defined as measurement or search gate. The setting only takes effect if "Show gate" is enabled.

Remote command:
`DISPlay:GATE:TRANsparency` on page 1215

**Search result gate symbol color**
Sets the color of the search zoom area. The search zoom area is displayed if "Show search zoom windows" is enabled.

See also: Chapter 11.4.2, "Display zoom settings", on page 470.

**Search result line color**
Sets the color of the search result markers. The markers are displayed if "Show search zoom windows" is enabled.

**Zoom overlay**
Shows all zooms of a diagram in one zoom window. The zoomed areas are overlaid for better comparison of the zoomed waveforms.

The setting affects all zoom diagrams.

See also: Chapter 7.1, "Zoom", on page 261

Remote command:
`LAYout:ZOOM:ONEDiagram` on page 1216

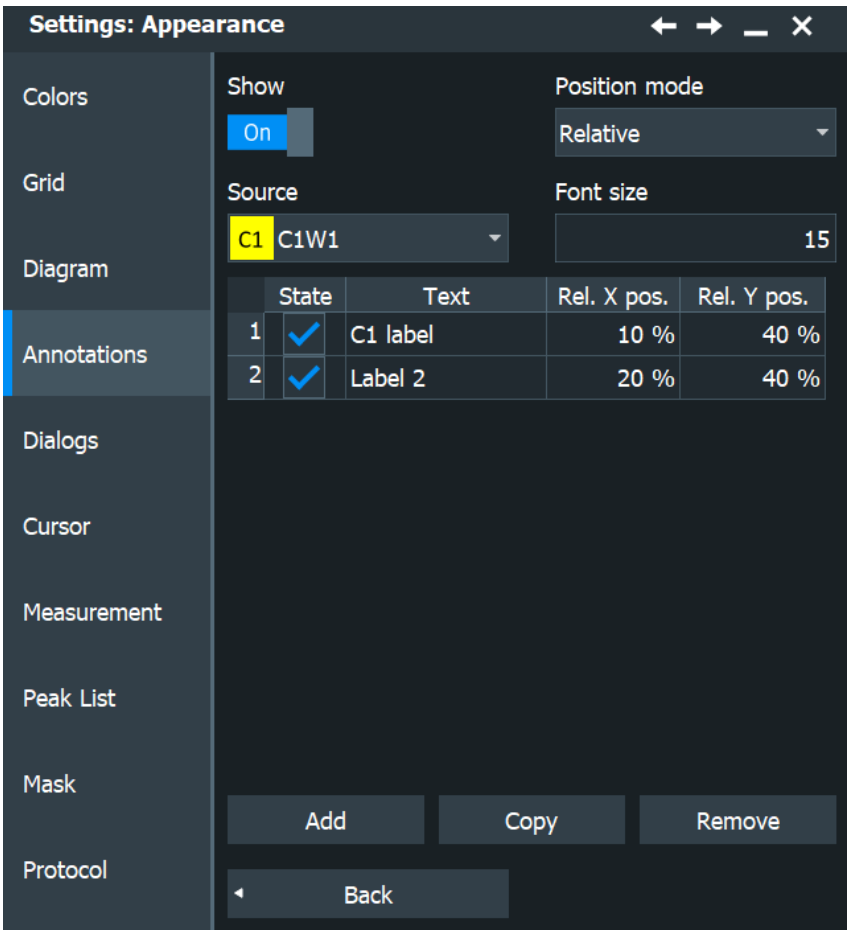
### 4.3.5 Annotations appearance settings

Access: "Settings" > "Appearance" > "Annotations" tab

Using labels, you can annotate the waveforms to name or explain each waveform. The text is shown in the same color as the assigned waveform. Each label has its individual position.

You can enter exact positions in the dialog box, or drag the labels on the screen to the required position. The position can be a fixed one (relative to the screen), or a flexible position (absolute, assigned to the axes).

To add labels quickly, you can add the "Label" icon to the toolbar and use it.




**Show / Source**
Selects the souce of the label and enables the display.

Remote command:
`DISPlay:SIGNal:LABel:SHOW` on page 1219

**Position mode**
Defines the label position either relative to the diagram or with absolute values according to the units of the waveform. Relative positions are fixed, whereas absolute positions move with the waveform display when the scales, the vertical position or offset, or the reference point are changed.

The position mode applies to all labels of the selected waveform. For different waveforms, different position modes can be selected.

"Relative" Sets a fixed position in percent of the screen counting from the upper left corner.

"Absolute" Sets the position in time and voltage values, or in other units depending on the waveform character.

Remote command:
`DISPlay:SIGNal:LABel:POSMode` on page 1217

**Font size**
Defines the size of the labels in the diagram.

Remote command:
DISPlay:SIGNal:LABel:FONTsize on page 1217

**Labels**
For each waveform, the "Labels" table shows the defined labels with their visibility state, label texts and positions. The labels belog to the selected waveform.

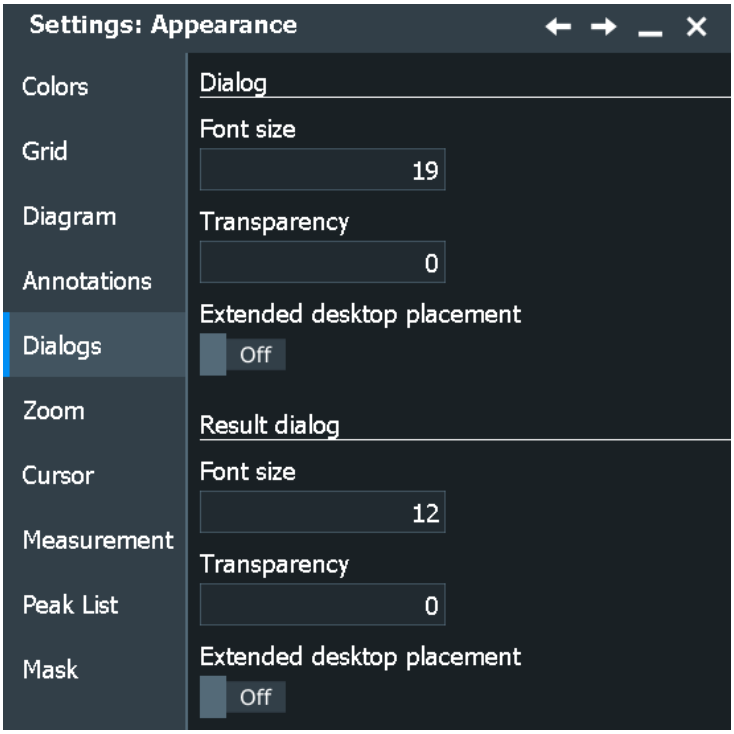
Enter the label text and the horizontal and vertical positions for each label. Enable "State" to show the label on the screen. The individual state of the labels takes effect if "Show" is active.

| | |
|---|---|
| "Add" | Adds a line at the end of the list. |
| "Copy" | Copies the selected line in a new line. |
| "Remove" | Deletes the selected line. Only single lines can be removed.<br>You can also delete a label by using the toolbar: Tap the "Delete" icon and then the label. |

Remote command:
DISPlay:SIGNal:LABel:ADD on page 1217
DISPlay:SIGNal:LABel:STATe on page 1220
DISPlay:SIGNal:LABel:IDSList on page 1219
DISPlay:SIGNal:LABel:REMove on page 1219
DISPlay:SIGNal:LABel:TEXT on page 1221
DISPlay:SIGNal:LABel:HORizontal:ABSolute:POSition on page 1221
DISPlay:SIGNal:LABel:VERTical:ABSolute:POSition on page 1221
DISPlay:SIGNal:LABel:HORizontal:RELative:POSition on page 1222
DISPlay:SIGNal:LABel:VERTical:RELative:POSition on page 1222

### 4.3.6 Dialogs appearance settings

Access: "Settings" > "Appearance" > "Dialogs" tab




**Dialog**
With these settings, you can change the appearance of the dialogs.

**Font size ← Dialog**
Defines the font size of the text in dialog boxes.

**Dialog transparency ← Dialog**
Defines the transparency of the dialog box background. For high transparency values, you can see the waveform display in the background, and possibly check the effect of the changed setting. For lower transparency values, readability in the dialog box improves.

**Result dialog**
With these settings, you can change the appearance of the result dialogs.

**Font size ← Result dialog**
Defines the font size of the text in result boxes. The size of the result box is adapted to the font size.

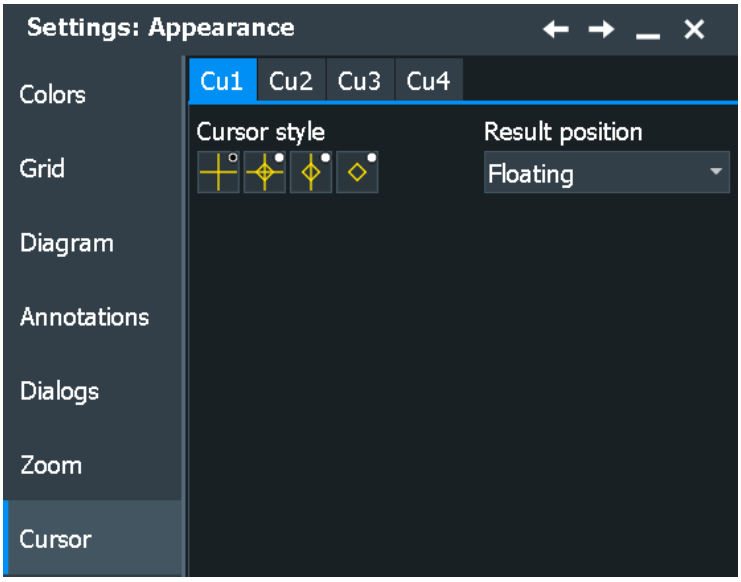
**Transparency ← Result dialog**
Defines the transparency of the measurement result boxes in the same way as Dialog transparency.

**Extended desktop placement**
If an external monitor is connected to the instrument, you can enable these settings to display dialog boxes and/or result boxes on the external monitor. Thus, the boxes do not cover the waveforms on the instrument display.

Remote command:
DISPlay:EXTended:PORDialogs on page 1222
DISPlay:EXTended:POSDialogs on page 1222

### 4.3.7 Cursor appearance settings

Access: "Settings" > "Appearance" > "Cursor" tab




**Cursor style**
Defines how the cursor is displayed in the diagram.

| | |
|---|---|
| "Lines" | The cursors are displayed as lines. |
| "Line & Rhombus" | The cursors are displayed as lines. The intersections of the cursors with the waveforms are displayed by rhombus-shaped points. |
| "Vertical line and rhombus" | The cursors are displayed as vertical lines. The intersections of the cursors with the waveforms are displayed by rhombus-shaped points. |
| "Rhombus" | The intersections of the cursors with the waveforms are displayed by rhombus-shaped points. |

Remote command:
CURSor<m>:STYLe on page 1223

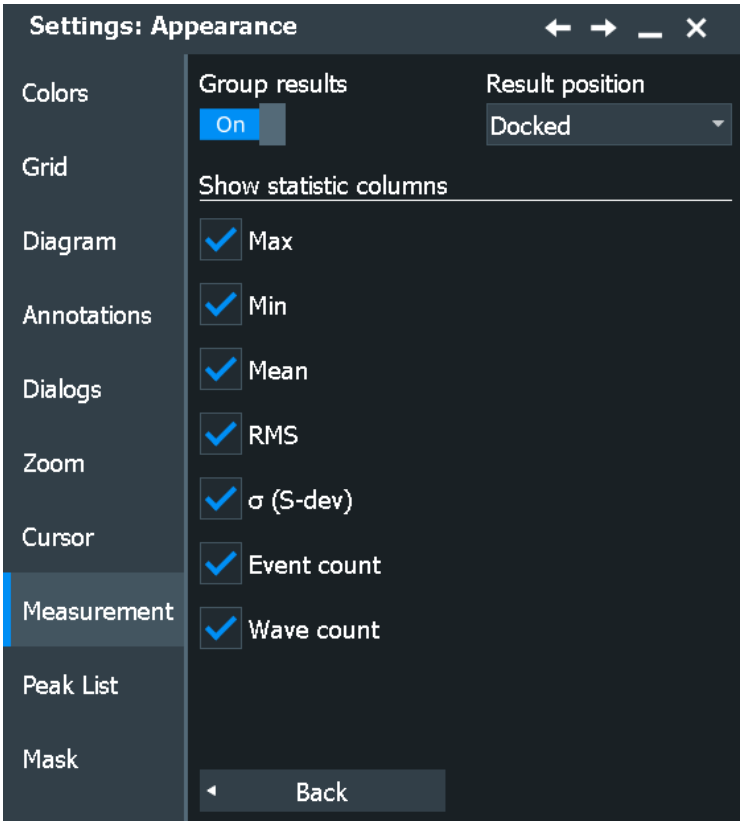
**Result position**
Defines the position of the cursor measurement results.

| | |
|---|---|
| "Floating" | Floating result box in front of the diagrams. |
| "Preview" | Result icon on the signal bar. |
| "Docked" | Fixed tab below the diagrams. |

Remote command:
DISPlay:RESultboxes:CUPosition on page 1223

### 4.3.8 Measurement appearance settings

Access: "Settings" > "Appearance" > "Measurement" tab




**Group results**
If enabled, all results are shown in one result box (default).

If disabled, the results of each measurement group are shown in a separate result box. The default position is ignored.

The setting affects all measurements except for the peak list.

Remote command:
`MEASurement<m>:DISPlay:GROuping` on page 1224

**Result position**
Defines the default position of the result table, when a new measurement is started:

- "Docked": fixed tab below the diagrams
- "Preview": result icon on the signal bar
- "Floating": floating result box in front of the diagrams

The setting affects only grouped results.

Remote command:
`DISPlay:RESultboxes:MEPosition` on page 1224

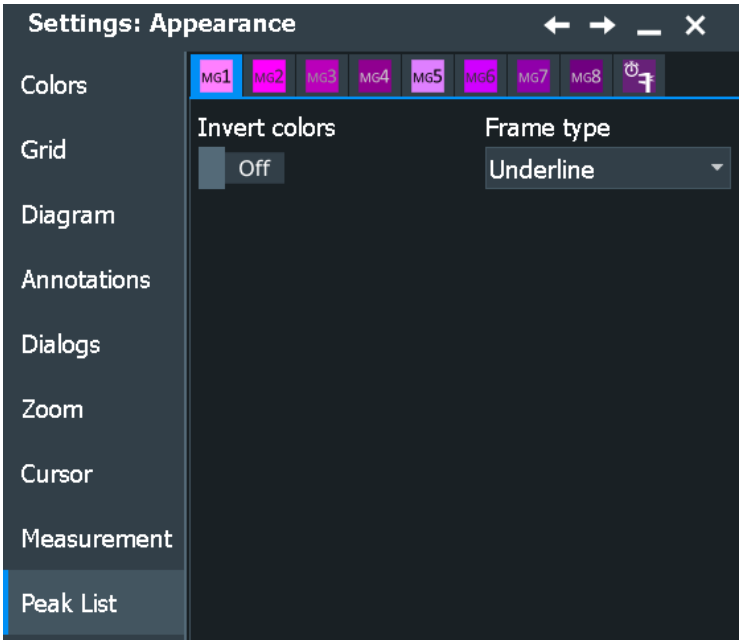
**Show statistic columns**
Select the statistical values that you want to see in the results table.

### 4.3.9 Peak list appearance settings

Access: "Settings" > "Appearance" > "Peak List" tab

For peak lists measurements only. You can enable the "Peak list" for "Spectrum" measurements, see Chapter 8.2.7, "Spectrum measurements", on page 363.

In this dialog, you can configure the look of the labels for peaks in the spectrum diagram.




**Frame type**
Defines the layout of the labels (full border, underline, or none).

Remote command:
`MEASurement<m>:RESult:LABorder` on page 1224

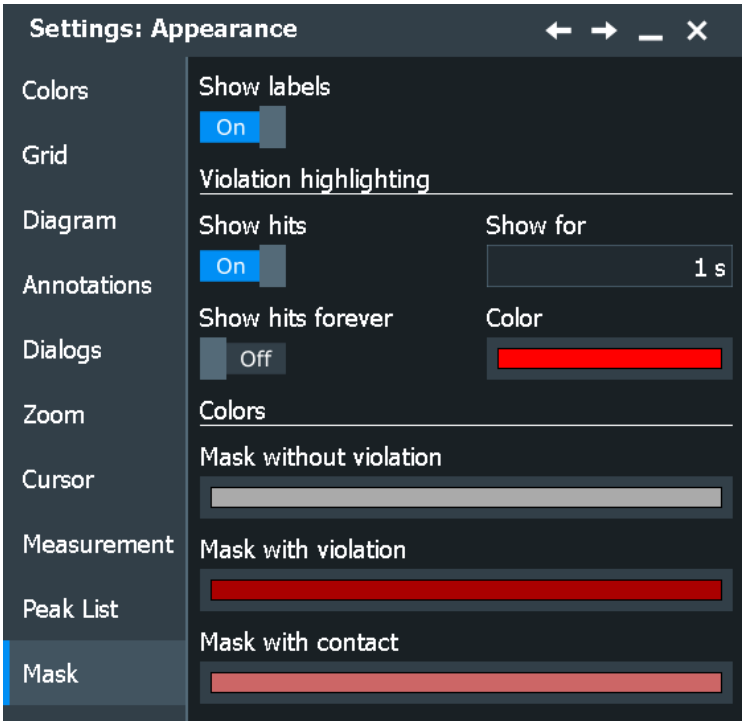
**Invert**
Displays black font on white background using the "Full border" frame type.

Remote command:
`MEASurement<m>:RESult:INVerse` on page 1225

### 4.3.10 Mask appearance settings

Access: "Settings" > "Appearance" > "Mask"

In this dialog, you can define the settings for mask and hit display. See also Chapter 10, "Mask testing", on page 423.




**Show labels**
Switches the display of the mask test name on or off.

To change the name of the mask test, open the "Mask Test" dialog > "Define Mask" tab. Double-tap the mask test subtab and enter the new mask test name.

Remote command:
`MTESt:LABel` on page 1226
`MTESt:REName` on page 1226

**Show hits**
If selected, the mask hits are highlighted on the screen. You can define the color and the time of the hit display.

These settings are not applied to trigger zones, which are based on masks.

Remote command:
`MTESt:HIGHlight:STATe` on page 1227

**Show for**
Sets the time for which the mask hits are shown.

Remote command:
`MTESt:HIGHlight:TIME` on page 1227

**Show hits forever**
If selected, the mask hits are shown for an unlimited period of time.

Remote command:
`MTESt:HIGHlight:INFinite` on page 1226

**Color**
Sets the color of samples that violated the mask.

Remote command:
`MTESt:COLor:MATCh` on page 1227

**Mask without violation**
Sets the color of masks segments that were not hit.

Remote command:
`MTESt:COLor:UNMatch` on page 1227

**Mask with violation**
Sets the color of mask segments the signal has entered into.

Remote command:
`MTESt:COLor:INTerior` on page 1228

**Mask with contact**
Sets the color of masks segments that were touched at the border. In this case, the resolution is not sufficient to detect if the mask was hit or not. Zoom into the concerned area to see the actual result.

Remote command:
`MTESt:COLor:BORDer` on page 1228

### 4.3.11 Protocol appearance settings

Access: "Settings" > "Appearance" > "Protocol"

In this dialog, you can define the position of the result table for the protocol analysis.

See also Chapter 13, "Protocol analysis", on page 515.

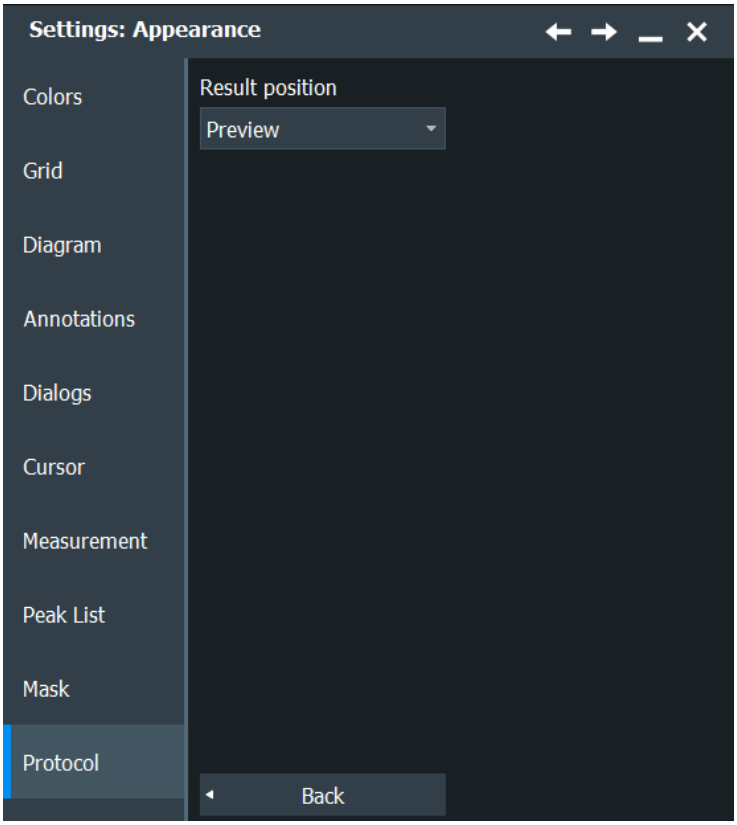



**Table position**
Defines the position of the decode table on the screen.

"Floating" Floating result box in front of the diagrams.

"Preview"" Result icon on the sidebar.

"Docked" Fixed tab below the diagrams.

Remote command:
`DISPlay:RESultboxes:DEPosition` on page 1640

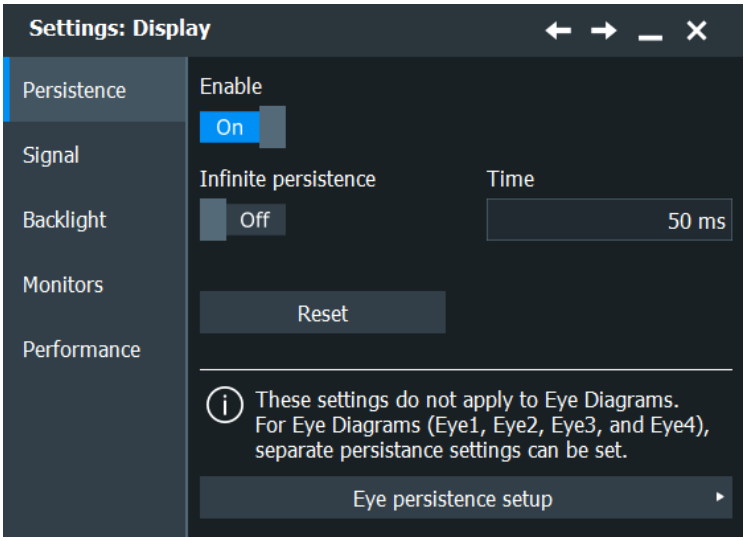
## 4.4 Display settings

In the "Settings" > "Display" dialog box, you can define the display settings like brightness and signal intensity. Yo can also set up an external monitor.

### 4.4.1 Persistence settings

Access: "Menu" > "Settings" > "Display" > "Persistence"

The settings in this dialog do not apply to eye diagrams.




**Enable**
If enabled, each new data point in the diagram area remains on the screen for the duration that is defined using Time, or as long as Infinite persistence is selected.

If disabled, the waveform points are displayed only for the current acquisition.

Remote command:
`DISPlay:PERSistence[:STATe]` on page 1229

**Infinite persistence**
If infinite persistence is enabled, each new waveform point remains on the screen until this option is disabled. Use infinite persistence to display rare events in the signal.

Remote command:
`DISPlay:PERSistence:INFinite` on page 1229

**Time**
Sets a time factor that controls how long the waveforms points fade away from the display. Thus, the R&S RTP emulates the persistence of analog phosphor screens.

Remote command:
`DISPlay:PERSistence:TIME` on page 1229

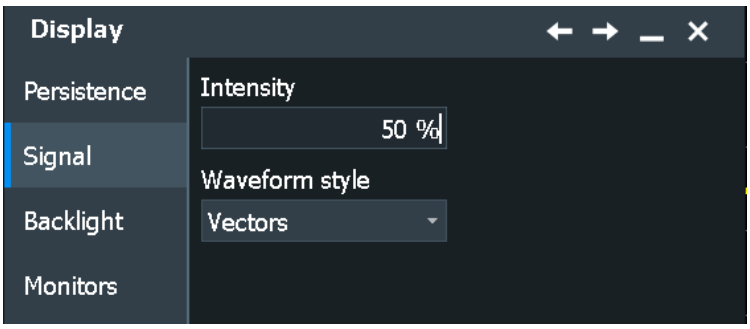
**Reset**
Resets the display, removing persistent all waveform points.

Remote command:
`DISPlay:PERSistence:RESet` on page 1230

**Eye persistence setup**
Opens the dialog for defining the persistence settings for the eye diagram settings, see Chapter 13.23.4.2, "Display settings", on page 931.

### 4.4.2 Signal settings

Access: "Menu" > "Settings" > "Display" > "Signal"




**Intensity**
This value determines the strength of the waveform line in the diagram. Enter a percentage between 0 (not visible) and 100% (strong). The default value is 50%.

You can also use the [Intensity] knob on the left side of the screen to adjust the waveform intensity directly.

**Note:** Use of color tables. The exact mapping of the cumulative value occurrences according to the assigned color table is guaranteed only if the intensity is set to 50%. All other intensity values falsify the mapping but can improve the visibility of the signal.

Remote command:
`DISPlay:INTensity` on page 1230

**Waveform style**
Select the style in which the waveform is displayed:

"Vectors" The individual waveform points are connected by a line.
Define the strength of the line using the [Intensity] knob.

"Dots" Only the individual waveform points are displayed. Waveform sample points are the ADC sample points and additional interpolated points if "Interpolated time" is used for resolution enhancement. To see the dots of one waveform, perform one acquisition with [Single] and N=1 ("Average count" = 1). During continuous acquisition, or a [Single] acquisition with N > 1, the dots of multiple subsequent waveforms are displayed on the screen, and the waveform looks like a line.
Consider also the "Interpolation" on page 130.

Remote command:
`DISPlay:DIAGram:STYLe` on page 1230

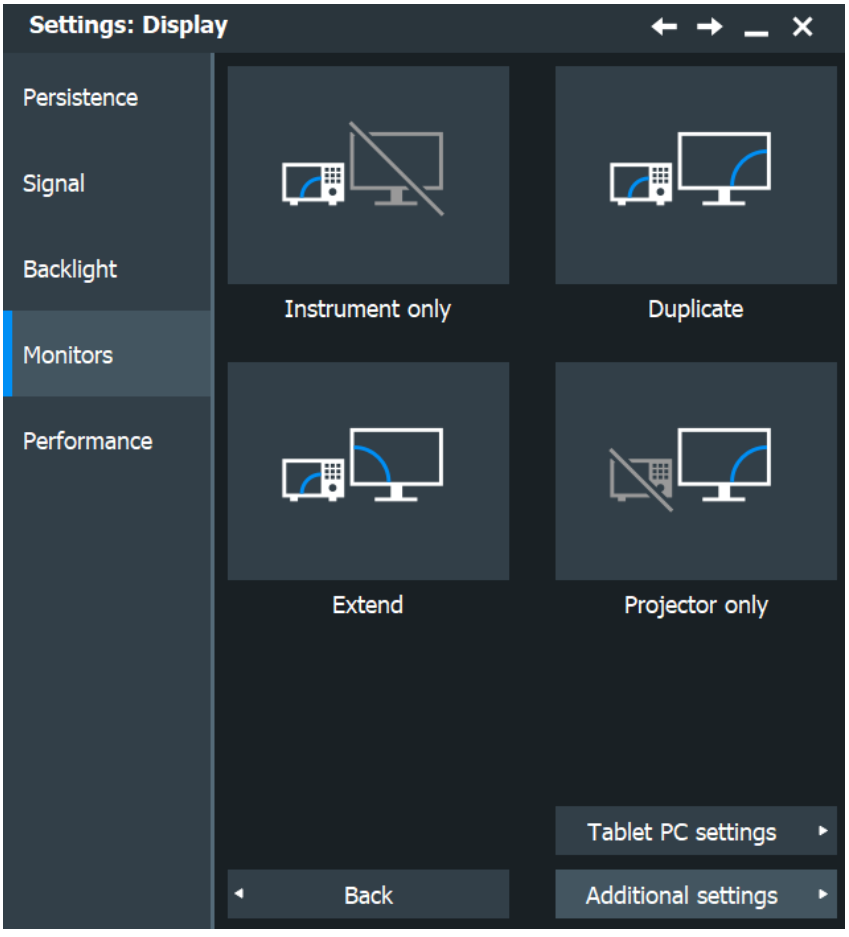
### 4.4.3 Backlight settings

Access: "Menu" > "Settings" > "Display" > "Backlight"

**Brightness**
Changes the background luminosity of the touchscreen.

### 4.4.4 Monitors settings

Access: "Menu" > "Settings" > "Display" > "Monitors"



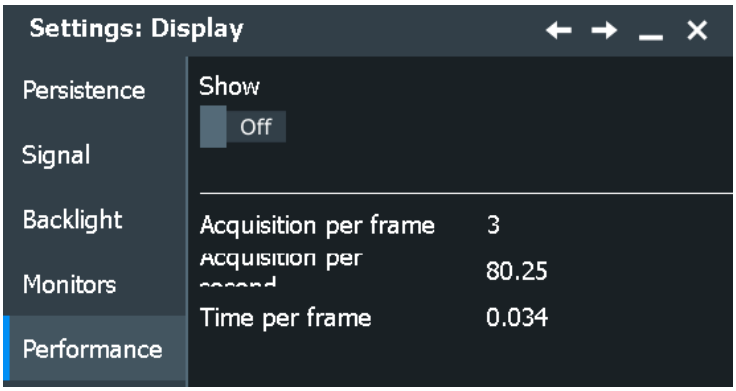

In the "Monitors" tab you can extend or duplicate the instrument display to a second monitor or projector (external display). The following options are available:

- "Instrument only": the instrument display is on, the external display is off.

- "Duplicate": the external display shows the same content as the instrument display.
- "Extend": the instrument display and the external display show different content.
- "Projector only": the instrument's user interface is only shown on the external display, the instrument display is off.
- "Tablet PC settings": opens the Windows setup for touch display to adjust the touch functionality when an additional monitor is in use.
- "Additional settings": opens the Windows configuration for display settings.

### 4.4.5 Performance settings

Access: "Menu" > "Settings" > "Display" > "Performance"




**Show**
Displays the "Performance" results box.

The "Performance" result box shows information on the current acquisition performance values of the R&S RTP.

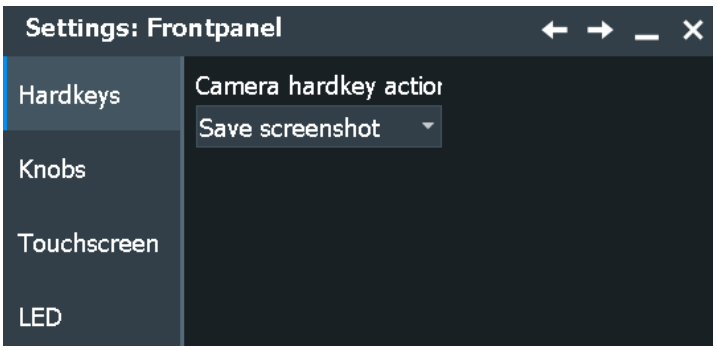
The instrument groups acquired waveforms together in a frame, and displays the frame content. The maximum number of frames displayed per second is about 30. The current number of frames per second is indicated as reciprocal "Time per frame". If the time scale decreases, and thus the number of acquisitions per second also decreases, the number of acquisitions per frame can drop to 1.

## 4.5 Frontpanel settings

In the "Frontpanel" dialog box, you can assign functions to keys and knobs, adjust the Multiuse knob and adjust the brightness of the keys.

### 4.5.1 Hardkeys: function assignment

Access: "Settings" > "Frontpanel" > "Hardkeys"



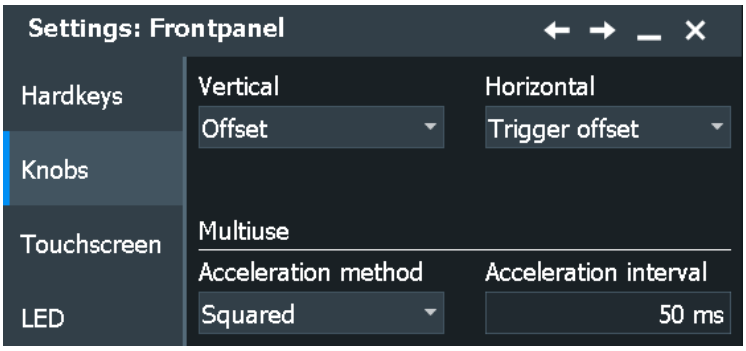

**Camera hardkey action**
The Camera key on the left side of the display is a shortcut key that initiates an associated action.

You can assign one of the following actions:
- Save a screenshot
- Open screenshot setup
- Save a report
- Open report setup

Configure the settings for the selected action.
- Screenshots: "Save/Recall" key > "Save" tab > "Screenhot", see Chapter 12.3.1, "Screenshot settings", on page 508.
- Report: "Save/Recall" key > "Save" tab > "Report", see Chapter 12.4.1, "Report settings", on page 512.

## 4.5.2 Knobs

Access: "Settings" > "Frontpanel" > "Knobs"




Vertical ..... 98
Horizontal ..... 99
Acceleration method ..... 99
Acceleration interval ..... 99

**Vertical**
Selects the action that you want to perform with the vertical [Position] knob. It can change the offset or the waveform position of the selected waveform.

See also: "[Position]" on page 40.

**Horizontal**
Selects the action that you want to perform with the horizontal [Position] knob. It can change the horizontal position of the trigger point (trigger offset), or the reference point.

See also: "[Position]" on page 38.

**Acceleration method**
Selects a method to accelerate the movement of the element on the screen compared to the actual movement of the [Multiuse] knob.

Acceleration is useful if you need to move from one end of the screen to the other, for example. Without acceleration, you have to turn the knob quite a while to reach the other end. But acceleration can make precise selection difficult, because a small movement of the knob causes a relatively large movement on the screen.

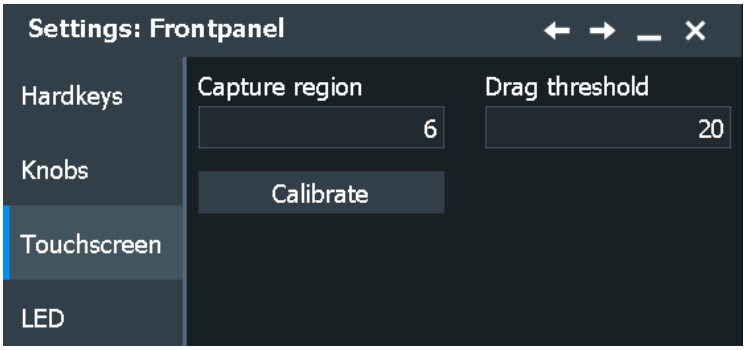
| | |
|---|---|
| "None" | No acceleration method used. |
| "Squared" | Moderate acceleration method used. |
| "Exponential" | Strong acceleration method used. |

**Acceleration interval**
Defines the delay time during which the movement of the [Multiuse] knob is analyzed before acceleration is applied. For short intervals, acceleration sets in quickly, but is not as effective. For long intervals, acceleration is more effective. However, it takes longer until the instrument reacts on the knob's input.

Furthermore, when you turn the knob slowly during fine-tuning, subsequent movements that occur during the same interval are accelerated, making precise selection difficult.

### 4.5.3 Touchscreen

Access: "Settings" > "Frontpanel" > "Touchscreen".




**Capture region**
Defines the number of pixels around each element (e.g. button, icon, data point) that create a capture region. If you tap your finger or click the mouse pointer within this capture region, this element is considered to be selected. If you tap or click outside this area, a different or no element is selected.

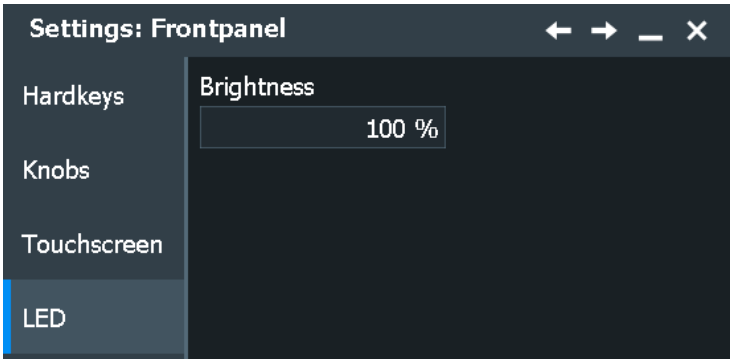
The larger the region, the easier is it to select an element. However, when selecting data points, for example, a large frame does not allow you to select precisely.

**Drag threshold**
Defines the maximum number of pixels around an element (e.g. data point) within which your pointing device must stay to "click" the element. When you tap or click a specific element and move your finger or the mouse outside this range, it is considered to be a "dragging" or "moving" operation.

**Calibration**
Opens the touchscreen calibration application. Only users with administrator rights can fulfill this task.

### 4.5.4 LED

Access: "Settings" > "Frontpanel" > "LED".




**Brightness**
Defines the luminosity of illuminated front panel keys and knobs.

### 4.5.5 Aligning the touchscreen

When the device is delivered, the touchscreen is initially calibrated. However, to ensure that the touchscreen responds to the finger contact correctly, a touchscreen alignment is required. Only users with administrator rights can fulfill this task.

Alignment of the touchscreen is useful:

- At first use
- If the position of the instrument has been changed, and you cannot look straight on the screen
- If another person operates the instrument
- If you notice, that touching a specific point on the screen does not achieve the correct response

1. In the "Menu" > "Settings" menu, select "Frontpanel".

2. Select the "Touchscreen" tab.
3. Tap "Calibrate".

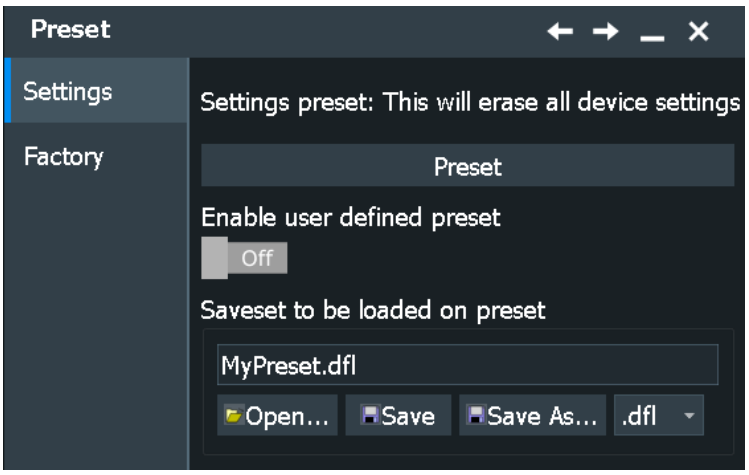
   A blinking cross appears in the lower left corner of the screen.
4. Touch and hold the cross until "OK" is shown.
5. Repeat this action for the crosses in the other corners.
6. Tap the Rohde & Schwarz logo to display the instrument's user interface.

## 4.6 Preset setup

A user-defined preset contains the complete instrument setup including display settings. You can save the current configuration to a preset file, and load a previously saved preset file. You can then specify that these settings are to be applied with the [Preset] key.

### 4.6.1 User-defined preset

Access: "Menu" > "Settings" > "Preset" > "Settings" tab




**Preset**
Performs a user-defined preset of the instrument.

**Saveset to be loaded on preset**
The file name with extension `.dfl` to load or to save the settings to.

Remote command:
`USRDefined:RST:NAME` on page 1231
`USRDefined:RST:SAVE` on page 1231
`USRDefined:RST:OPEN` on page 1231

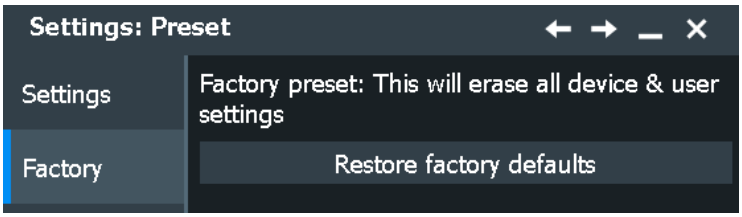
**Enable user defined preset**
If enabled, the settings from the selected preset file are restored when the [Preset] key is pressed.

If disabled, [Preset] sets the instrument to the factory defaults.

Remote command:
`USRDefined:RST[:ENABle]` on page 1231

## 4.6.2 Factory preset

Access: "Menu" > "Settings" > "Preset" > "Factory"




**Restore factory defaults**
Resets the instrument to the factory default settings, to the initial state. Factory settings comprise all instrument settings, including display, intensity and transparency settings. After loading factory defaults, perform a self-alignment to synchronize the signal data.

Remote command:
Chapter 24.7.1, "System", on page 1205

## 4.6.3 Restoring settings

When you have changed many different settings on the instrument and are no longer sure which settings are causing which effect in the measurement, you may want to restore the default settings and start anew. The following methods are available:

- Saving instrument settings to a user-defined preset and restoring the instrument settings to user-defined default values
- Restoring all settings on the R&S RTP to the factory-defined values
- Restoring settings from a file (see "To load settings from a saveset file" on page 481)

For details on save/recall instrument settings and associated remote commands, see Chapter 12.1.1, "Savesets", on page 475.

**To save a user-defined preset**

1. Open the "Menu" > "Settings" > "Preset" tab.
2. Enter a name for the preset file. Select the file format.
3. Tap Save.

**Note:** If you want to store the file in another directory than the default one, select "Save As". See also: Chapter 12.5, "File selection dialog", on page 513

**To restore the instrument settings to user-defined default values**

1. Open the "Menu" > "Settings" > "Preset" tab.
2. Tap "Open" and select the preset file that contains the required settings.

   The instrument settings are restored to values that are stored in the file.
3. To use these settings as preset values, select "Enable user-defined preset".
4. Press the [Preset] key.

**To restore all settings to the factory defaults**

1. Open the "Menu" > "Settings" > "Preset" > "Factory" tab.
2. Tap the "Restore factory defaults" button.

   All settings on the R&S RTP are reset to their factory-defined values. As long as no user-defined preset file is loaded and "Enable user defined preset" is disabled, the [Preset] key also resets the instrument settings to factory defaults.

## 4.7 Maintenance settings

In the "Settings" > "Maintenance" dialog box, you define the the look and feel of the display element, e.g. waveform colors, result position, or grid behavior.

### 4.7.1 Firmware update

Access: "Menu" > "Settings" >"Maintenance" > "FW Update"

Your instrument is delivered with the latest firmware version. Firmware updates are provided on the internet at:

www.rohde-schwarz.com/firmware/rtp.

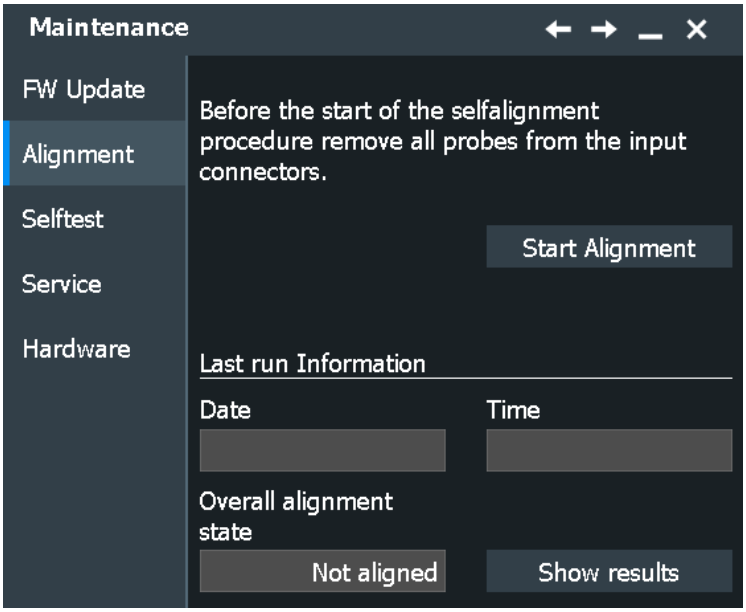
The "Release Notes" describe the improvements and modifications of all firmware versions. They also explain how to update the firmware. They are available along with the firmware on the same web page.

### 4.7.2 Alignment

When data from several input channels is displayed at the same time, it may be necessary to align the data vertically or horizontally to synchronize the time bases or amplitudes and positions. This is the case, for example, when strong temperature changes occur (> 5°).

#### 4.7.2.1 Control

Access: "Menu" > "Settings" >"Maintenance" > "Alignment"




**Start Alignment**
Starts the self-alignment procedure for all channels.

Remote command:
`*CAL?` on page 1197

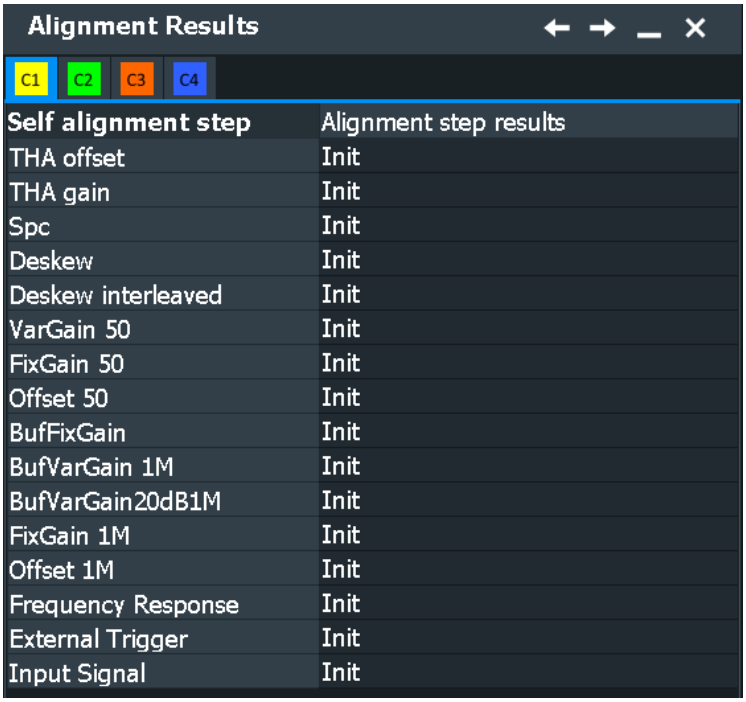
**Date / Time / Overall alignment state**
Show the date, time and the summary result of the self-alignment process: not aligned, passed or failed. Detailed results are provided on the "Results" tab.

Remote command:
`CALibration:DATE?` on page 1232
`CALibration:TIME?` on page 1232
`CALibration:RESult?` on page 1232

#### 4.7.2.2 Results

For each channel, the results of the individual alignment steps are shown for all technical channel component. In case you require support, you may be asked to provide this information.

Access: "Settings" >"Maintenance" > "Alignment" > "Show results"




#### 4.7.2.3 Performing a self-alignment

The self-alignment aligns the data from several input channels vertically and horizontally to synchronize the timebases, amplitudes and positions. The self-alignment process includes a basic hardware check.

Recommendation on performing the self-alignment:

- When putting the instrument into operation for the first time
- After a firmware update
- Once a week
- When major temperature changes occur (> 5°)

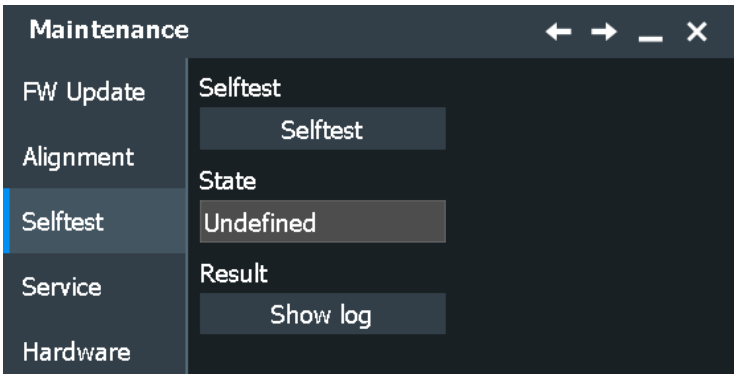
1. Warm up the instrument before you start the self-alignment. The minimum warm-up time is indicated in the data sheet.
2. Remove the probes from the input connectors.
3. In the "Menu" > "Settings" menu, select "Maintenance".
4. In the "Alignment" tab, tap "Start Alignment".

   The alignment is performed, the process might take several minutes. A message box informs you about the running process, wait until this message box closes. The overall pass/fail result is shown in the "Overall alignment state" field. The results of the individual alignment steps for each input channel are indicated in the "Results" tab. This information is required if problems arise.

### 4.7.3 Self-test

The instrument's self-test checks the hardware for correct operation. Perform the self-test if you suspect problems in hardware operation.

Access: "Settings" >"Maintenance" > "Selftest"




The test can take several minutes. The summary result is shown in the "State" field, which can be helpful in case you need support.

**Selftest**
Starts the self-test.

Remote command:
`*TST?` on page 1200

**State**
Shows the summary result of the self-test: Pass or Fail.

Remote command:
`DIAGnostic:SERVice:STST:STATe?` on page 1233

**Result**
Opens a log file with detailed information on the self-test steps and operation of hardware components, which can be helpful in case you need support.

### 4.7.4 Maintenance information

Access: "Settings" >"Maintenance" > "Hardware" tab > "Advanced"

The "Maintenance" dialog box provides information on your R&S RTP configuration, which can be helpful in case you need support.

**System Info**
This tab provides general information on the hardware configuration, and indicates where system information can be found on the instrument. Here you can also show the content of the device footprint file.

Remote command:
DIAGnostic:SERVice:PARTnumber on page 1234
DIAGnostic:SERVice:SERialnumber? on page 1234
SYSTem:DFPRint on page 1233

**Mainboard**
This tab provides information on the mainboard configuration in your instrument.

**Frontend**
This tab provides information on the frontend configuration in your instrument.

**Frontpanel**
This tab provides information on the front panel module installed in your instrument.

**MSO**
This tab is only relevant if the MSO option R&S RTP-B1 is installed. The tab provides information on the MSO hardware module that is installed in your instrument.

**AWG**
This tab is only relevant if the waveform generator option R&S RTP-B6 is installed. The tab provides information on the generator hardware module that is installed in your instrument.

**PSC**
This tab is only relevant if the pulse source option R&S RTP-B7 is installed. The tab provides information on the pulse source hardware module that is installed in your instrument.

**Service**
This tab allows the service personnel to enter a password that activates further service functions.

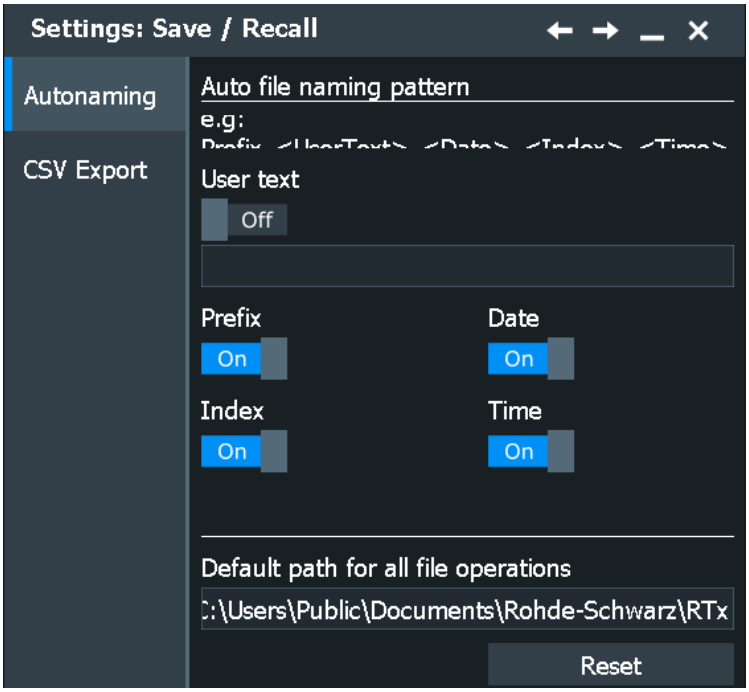
Remote command:
- DIAGnostic:SERVice:PWD on page 2463

## 4.8 Save / recall

### 4.8.1 Autonaming

Access:"Menu" > "Settings" key > "Maintenance" > "Autonaming" tab.

#### 4.8.1.1 Autonaming settings




In this tab, you can define the pattern for automatic file name generation. This name is used as the default file name. The default path is the storage location for all saved files and their subdirectories.

**Prefix**
If enabled, inserts the default prefix in the file name. The prefix indicates the type of data that is saved, for example, Histogram, RefCurve, Settings.

Remote command:
`MMEMory:AUTonaming:PREFix` on page 1633

**User text (enable)**
If enabled, inserts the specified user text after the prefix.

Remote command:
`MMEMory:AUTonaming:USERtext` on page 1633

**Text input**
User-defined text to be inserted after the prefix.

Remote command:
`MMEMory:AUTonaming:TEXT` on page 1633

**Date**
If enabled, inserts the current date.

Remote command:
`MMEMory:AUTonaming:DATE` on page 1633

**Index**
If enabled, inserts an index.

Remote command:
MMEMory:AUTonaming:INDex on page 1633

**Time**
If enabled, inserts the current time.

Remote command:
MMEMory:AUTonaming:TIME on page 1633

**Default path for all file operations**
Defines the default path displayed in the file selection dialog box for loading and storing operations. If a USB flash drive is connected, the path is set automatically to the drive letter of the USB flash drive.

To switch the directory quickly, double-tap the input field. Use the symbols on the left of the file explorer box to change the directory.

Remote command:
MMEMory:AUTonaming:DEFaultpath on page 1634

**Reset**
Resets the default file path to the factory default.

Remote command:
MMEMory:AUTonaming:RESPath on page 1634
MMEMory:AUTonaming:RESall on page 1634

#### 4.8.1.2 Defining default file paths and names

When a save or load operation is performed, a default file name and path is provided. You can configure which path is used and how the file name is generated. In the file selection dialog box, you can change the folder and name as desired.

**To define the default file path**

1. Tap "Menu" > "Settings".
2. Select the "Save /Recall" tab.
3. Select the "Autonaming" tab.
4. Double-tap the "Default path for all file operations" field.

   The directory selection dialog box is opened.
5. Select the folder in which the data is stored by default. Use the symbols on the left of the file explorer box to switch to often used directories.
6. To restore the factory-set default path, tap "Reset" next to the path field.

**To define the automatic file name pattern**

The automatic file name pattern can consist of the following elements:

<Prefix>_<UserText>_<Date>_<Index>_<Time>

The prefix depends on the data type to be stored and cannot be changed by the user. The other elements can be enabled or disabled as required.

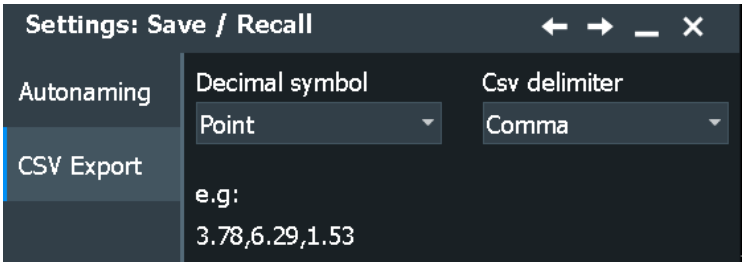
1. Tap "Menu" > "Settings".
2. Select the "Save /Recall" tab.
3. Select the "Autonaming" tab.
4. To insert a user-defined text after the prefix, enter the text in the edit field. and enable "User text".
5. If you want to exclude the prefix, current date, time or an index (serial number), disable the corresponding option.

   The specified elements are used to generate the default file name for the next storage operation.

### 4.8.2 CSV export

Access: "Menu" > "Settings" key > "Save /Recall" > "CSV Export" tab.

In this dialog, you can define the format of your CSV file.




**Decimal symbol**
Selects if point or comma is used as a decimal symbol in the exported CSV file.

Remote command:
`EXPort:RESult:DECSymbol` on page 1634

**Csv delimiter**
Selects the list separator symbol from a list. Available are semicolon, comma, space, tab and colon.

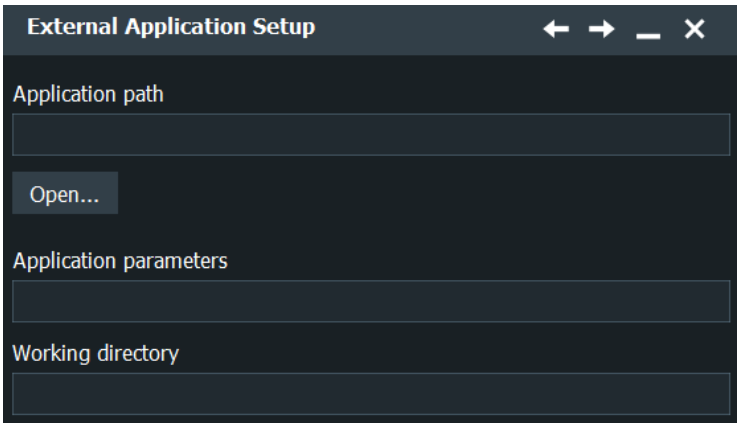
Remote command:
`EXPort:RESult:DELimiter` on page 1635

### 4.8.3 External application

The R&S RTP can start an external application on the instrument or in the network (if connected) when an event occurs.

The following events can start an application:

- Trigger event
- Mask violation
- Successful completion of mask test
- Limit or margin violation of measurements
- Successful completion of limit and margin tests




Set the path of the application executable, optional parameters, and the working directory as in a Windows shortcut definition. The setup is valid for all events.

Remote commands:

- `EXECutable:NAME` on page 1635
- `EXECutable:PARameter` on page 1635
- `EXECutable:WDIRectory` on page 1636

# 5 Acquisition and waveform setup

This chapter describes the horizontal and vertical settings as well as the acquisition and probe setup.

## 5.1 Basics

This chapter provides background information on the essential settings in the vertical and horizontal systems, on acquisition setup and probing.

### 5.1.1 Vertical system

The controls and parameters of the vertical system are used to scale and position the waveform vertically.

#### 5.1.1.1 Input coupling

The input coupling influences the signal path between input connector and the following internal signal stage. The coupling can be set to DC or ground.

- DC coupling shows all parts of an input signal. DC coupling is the default for 50 Ω input impedance to connect, for example, active probes.
- Ground coupling disconnects the input signal from the vertical system to see the ground level (zero volts) on the screen. Ground coupling is useful for reference purposes.
- Using the R&S RT-Z1M probe adapter, you can connect probes requiring 1 MΩ load impedance, for example, standard passive probes.

#### 5.1.1.2 Vertical scale and position

Vertical scale and vertical position directly affect the resolution of the waveform amplitude. The vertical scale corresponds to the ADC input range. To get the full resolution of the ADC, set up the waveforms to cover most of the height of the diagram.

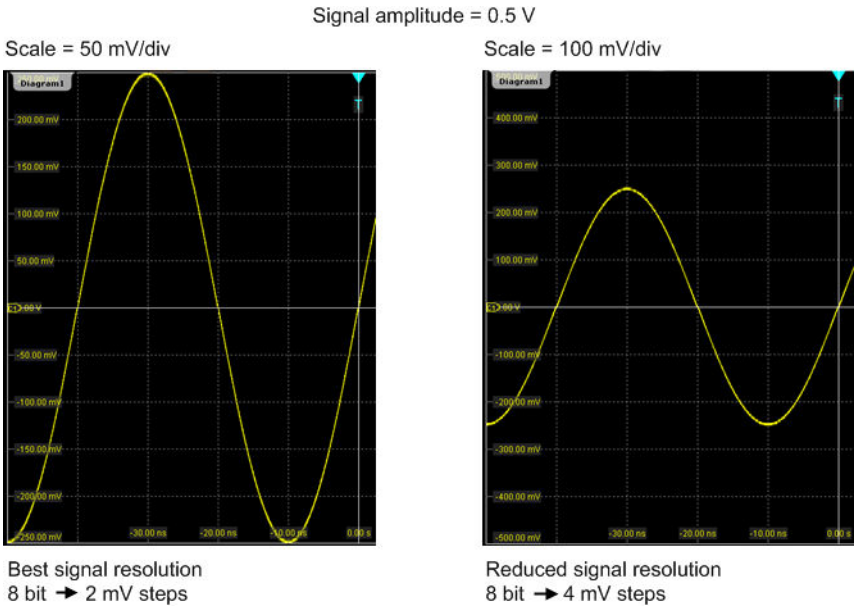



*Figure 5-1: Input range and resolution of the ADC*

With R&S RTP, you can work with multiple diagrams, and each diagram obtains the full vertical resolution, no matter where the diagram is placed. Therefore, use a separate diagram for each waveform instead of the traditional setup that arranges the waveforms side by side in one diagram.

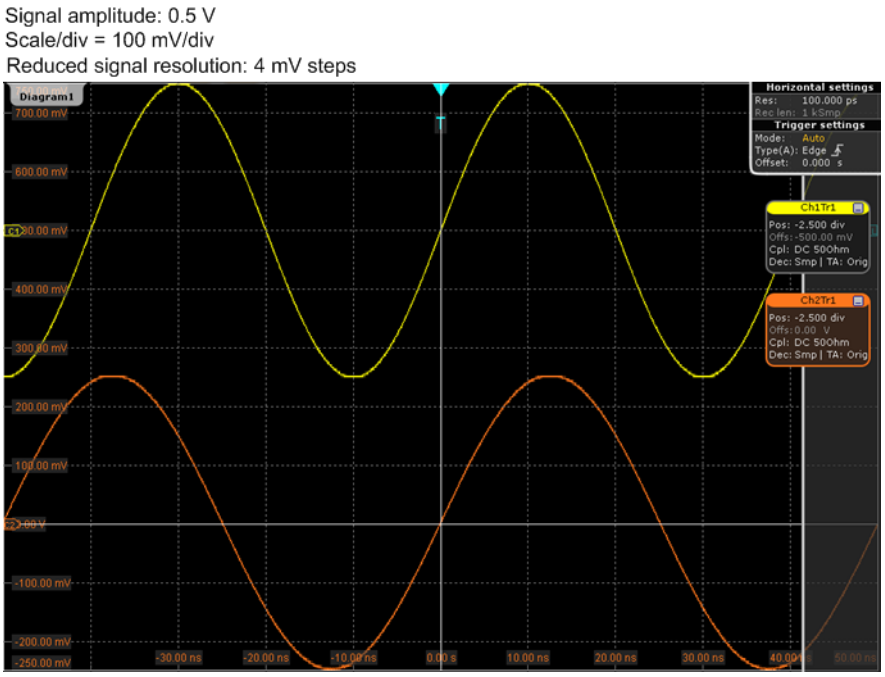



*Figure 5-2: Traditional setup of multiple waveforms in one diagram: reduced resolution*

Signal amplitude: 0.5 V
Scale = 50 mV/div
Best signal resolution: 2 mV steps

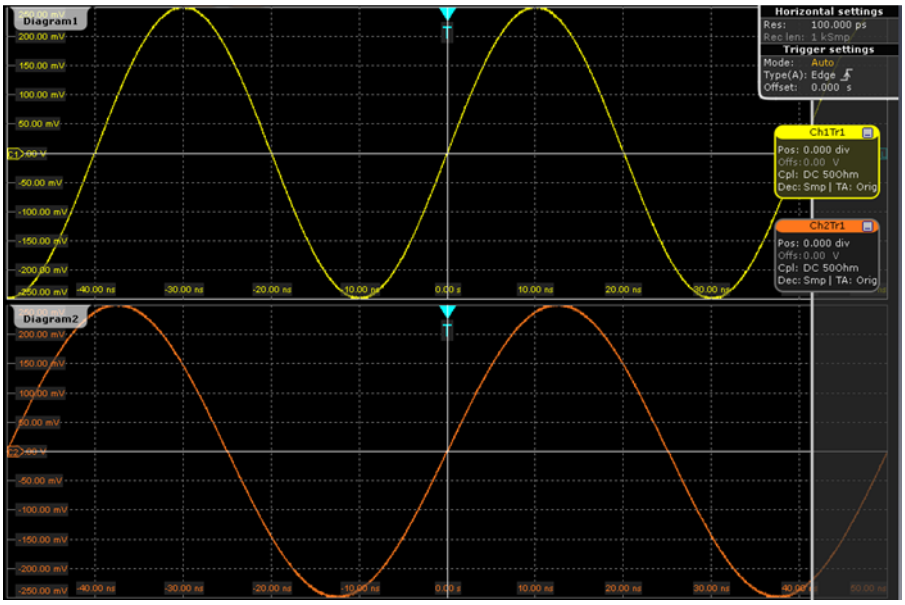



***Figure 5-3: R&S RTP setup of multiple waveforms in separate diagrams: best resolution***

### 5.1.1.3 Bandwidth

For analog applications, the highest signal frequency determines the required oscilloscope bandwidth. The oscilloscope bandwidth should be slightly higher than the maximum frequency included in the analog test signal to measure the amplitude with very little measurement error.

Most test signals are more complex than a simple sine wave and include several spectral components. A digital signal, for example, is built up of several odd harmonics. As a rule, for digital signals the oscilloscope bandwidth should be 5 times higher than the clock frequency to be measured.

The oscilloscope is not a stand-alone system. You need a probe to measure the signal of interest, and the probe has a limited bandwidth, too. The combination of oscilloscope and probe creates a *system bandwidth*. To reduce the effect of the probe on the system bandwidth, the probe bandwidth must exceed the bandwidth of the oscilloscope, the recommended factor is 1.5 x oscilloscope bandwidth. See also: Chapter 5.1.4.1, "Voltage probes", on page 119

#### Maximum bandwidth of R&S RTP134B and R&S RTP164B

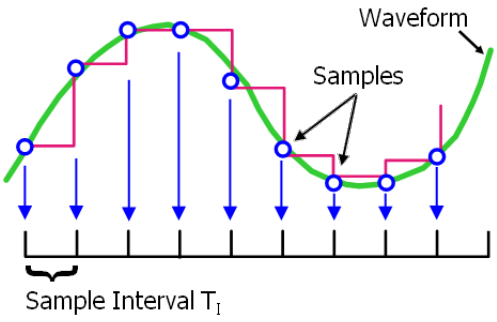
The maximum instrument bandwidth of 13/16 GHz is achieved when the instrument is running in interleaving mode. Interleaving is active if only two channels are used: either channel 1 or channel 2, and either channel 3 or 4. The maximum bandwidth is available on the two interleaved channels. If more channels are used, interleaving is disabled, and the maximum bandwidth is 8 GHz.

## 5.1.2 Sampling and acquisition

The vertical system of a digital oscilloscope conditions the test signal in a way that the following A/D converter (ADC) can transform the measured voltage into digital data.

### 5.1.2.1 Sampling and processing

The A/D converter samples the continuous signal under test at specific points in time and delivers digital values called **ADC samples**. The rate at which the converter is working is the **ADC sample rate**, a constant value specified in GHz: $f_{ADC} = 1 / T_I$




The digital ADC samples are processed according to the acquisition settings. The result is a waveform record that contains **waveform samples** and is stored in the **waveform memory**. The waveform samples are displayed on the screen and build up the waveform.

The number of waveform samples in one waveform record is called **record length**. The rate of recording waveform samples - the number of waveform samples per second - is the **sample rate**. The higher the sample rate, the better the resolution is and the more details of the waveform are visible.

#### Minimum sample rate and aliasing

A sufficient resolution is essential for correct reconstruction of the waveform. If the signal is undersampled, aliasing occurs - a false waveform is displayed. To avoid aliasing and accurately reconstruct a signal, Nyquist theorem postulates that the sample rate must be at least twice as fast as the highest frequency component of the signal. However, the theorem assumes ideal conditions, so the Nyquist sample rate is usually not sufficient.

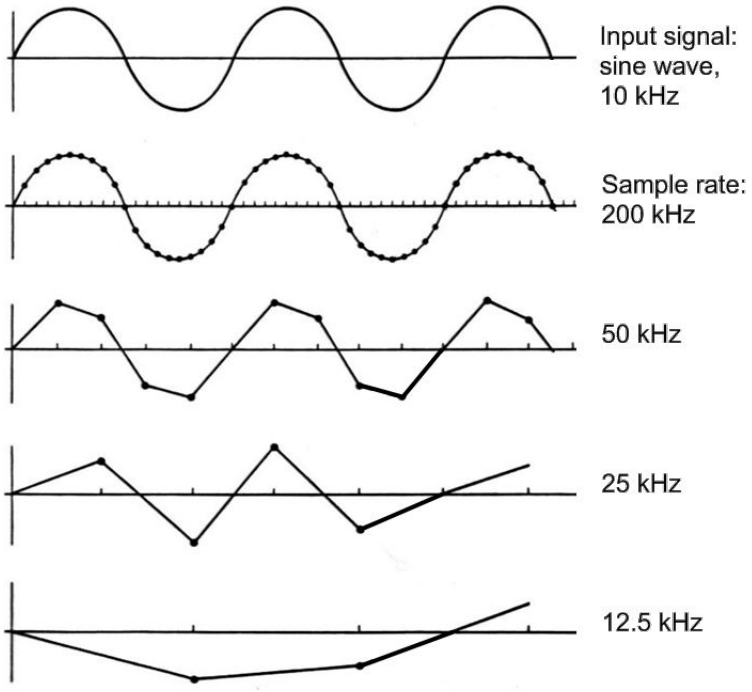



*Figure 5-4: Waveforms acquired with different sample rates*

To avoid aliasing, the sample rate must be set to a value 3 to 5 times the fastest frequency component of the signal. A higher sample rate increases signal fidelity, increases the chance to capture glitches and other signal anomalies, and improves the zoom-in capabilities.

#### 5.1.2.2 Acquisition settings

The sample rate can be the same as the constant ADC sample rate, or higher, or lower. To get a higher sample rate, interpolation as method of **resolution enhancement** is used. To reduce the sample rate, **decimation** methods help: sample, peak detect, high resolution and RMS.

As digital waveform data is stored in the memory, and the memory can save many waveform records, further **waveform arithmetic** processing is possible: average and envelope waveforms are resulting waveforms, created from a composite of sample points taken from multiple acquisitions.

You can display up to three waveforms from one input signal and apply different decimation and arithmetic to each waveform.

#### 5.1.2.3 Acquisition control

You can run the R&S RTP in two ways:

- Run / Stop: the instrument acquires data until you stop it manually.
- Single: the instrument samples and processes a specified number of acquisitions.

The determining point of an acquisition is the trigger. The instrument acquires continuously and keeps the sample points to fill the pre-trigger part of the waveform record. When the trigger occurs, the instrument continues acquisition until the post-trigger part of the waveform record is filled. Then it stops acquiring and waits for the next trigger. When a trigger is recognized, the instrument does not accept another trigger until the acquisition is complete.

The trigger modes define how the instrument triggers:

- Normal: The instrument acquires a waveform only if a real trigger occurs, that is, if all trigger conditions are fulfilled.
- Auto: The instrument triggers repeatedly after a fixed time interval if the trigger conditions are not fulfilled. If a real trigger occurs, it takes precedence. If the real trigger is faster than the auto trigger, both modes are virtually the same.

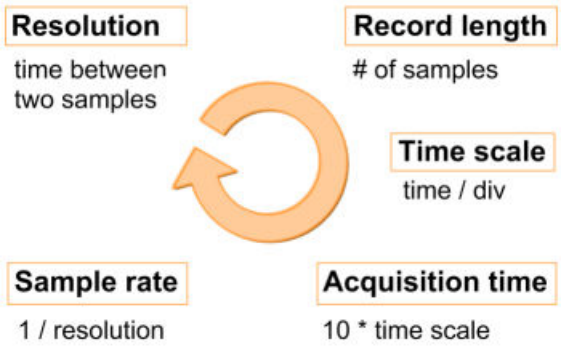
In practice, both trigger modes are useful: The auto mode lets you see the signal with little adjustment, while the normal mode selects the interesting part of the waveform. If you want to acquire a specified number of waveforms, make sure to select the normal trigger mode. Thus you get only the required number of interesting acquisitions.

See also: Chapter 6, "Triggers", on page 193.

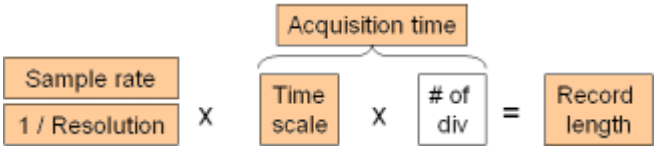
## 5.1.3 Horizontal system

### 5.1.3.1 Parameters of the horizontal system

The control parameters of the horizontal system are tightly connected. Thus, changing one parameter affects the other parameters as well.




The mathematical dependencies can be summarized as follows:




The number of divisions is 10, which is the only constant parameter.

When you set up horizontal parameters, you can choose whether the record length or the resolution remains constant.

- With constant resolution, increasing the time scale also increases the record length, and vice versa. You can limit the record length to a maximum value.
- With constant record length, increasing the time scale coarsens the resolution, that is, the time between two waveform samples gets longer.

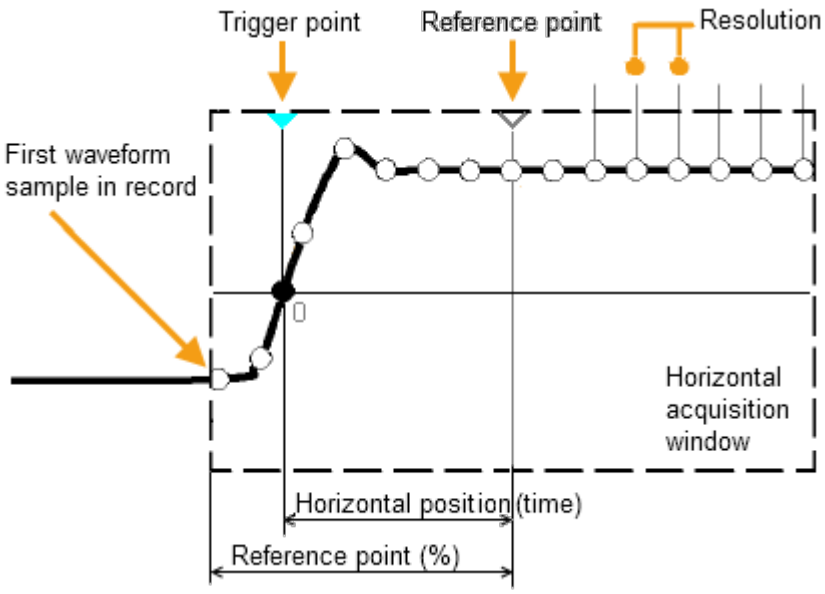
For both settings, the "Auto adjustment" ensures a sufficient resolution to prevent undersampling.

#### 5.1.3.2 Horizontal position

As described before in Chapter 5.1.2.3, "Acquisition control", on page 116, the trigger is the determining point of the waveform record.

In many scenarios, you want to analyze the waveform some time before or after the trigger. To adjust the horizontal acquisition window to the waveform section of interest, you can use the following parameters:

- The **horizontal position** defines the time distance from the trigger point (the zero point of the diagram) to the reference point. Changing the horizontal position, you can move the trigger point, even outside the screen.
- The **reference point** is the rescaling center of the time scale on the screen. If you modify the time scale, the reference point remains fixed on the screen, and the scale is stretched or compressed to both sides of the reference point.




### 5.1.4 Probes

A probe connects the signal source (DUT) to the oscilloscope, and delivers the signal to be measured. It is the essential first link in the measurement chain.

An ideal probe fulfills the following requirements:

- Safe and reliable contacts
- Infinite bandwidth
- The probe should not load the signal source and thus impact the circuit operation.
- The connection should not introduce or suppress signal components (hum, noise, filter) and thus degrade or distort the transferred signal.

In reality, the probe can never be an ideal one, it always affects the signal transmission and the signal source, and thus the measured signal. It depends on the frequency to be measured and on the signal source to determine the acceptable loading, and to determine which kind of probe delivers good results.

The solution depends on the quantity to be measured regarding:

- Signal type: voltage, current, power, pressure, optical, etc.
- Signal amplitude: The oscilloscope itself can only display voltages in a limited range. Most probes can adjust the dynamic range to amplitudes from a few mV to 10 V. Smaller or much larger signals require specialized equipment.
- Signal frequency: High frequencies require advanced equipment to get correct results.
- Source characteristic: The source impedance is the decisive factor when choosing the suitable connection.

#### 5.1.4.1 Voltage probes

The following table provides an overview on common voltage probes and their usage.

*Table 5-1: Voltage probes overview*

| Probe type | Attenuation | Typical bandwidth range | Oscilloscope input | Usage |
|---|---|---|---|---|
| Passive, high impedance | 1:1 | 10 MHz | 1 MΩ | Low-speed signals, low-level signals |
| Passive, high impedance | 10:1 | 500 MHz | 1 MΩ | General purpose |
| Passive, low impedance | 10:1 | up to 10 GHz | 50 Ω | High frequency |
| Active, single-ended | 10:1 | up to 10 GHz | 50 Ω | High speed |
| Active, differential | 10:1 | | 50 Ω | Floating |

For a list of recommended probes, refer to the R&S RTP product brochure.

Besides the possible input voltage range, two factors are important when selecting a voltage probe: Bandwidth and impedance over frequency.

- **Bandwidth:**
  The combination of probe and oscilloscope builds up a system. The resulting system bandwidth is approximately determined with:

$$\frac{1}{BW_{system}} = \sqrt{\left(\frac{1}{BW_{probe}}\right)^2 + \left(\frac{1}{BW_{scope}}\right)^2}$$

To measure the signal with low measurement error, the system bandwidth should be higher than the highest frequency component of the signal. The probe bandwidth must be even higher than the system bandwidth.

- **Impedance:**
  A minimum impedance is required to keep the circuit loading low. Over frequency, the impedance decreases, in particular with passive probes. The probe impedance should be approximately 10 times the impedance of the circuit test point at the highest signal frequency.

### Active voltage probes - general

Active probes require operating power from the instrument and have a proprietary interface to the instrument. Their main qualities are:

- Low loading on signal source
- The probe is automatically recognized by the instrument, no adjustment is required.
- Adjustable DC offset at probe tip allows for high resolution on small AC signals which are superimposed on DC levels.
- Connections should be as short as possible to keep the usable bandwidth high.
- Observe the operating voltage range.
- The probe impedance depends on the signal frequency.

RT-ZS single-ended active probes and RT-ZD differential active probes provide special features for easier use and precise measurements. These special features are not available on RT-ZSxxE probes.

- The micro button on the probe head remotely controls important functions on the instrument, like running and stopping the acquisition, autoset, AutoZero and setting the offset to mean value.
- The R&S ProbeMeter measures DC voltages between the probe tip and the ground connection with very high precision. The result is displayed on the instrument's screen. So you can check DC voltages with different levels without having to adjust the measurement range of the oscilloscope. The R&S ProbeMeter also measures the zero error of the probe to optimize measurement results at small signal levels.

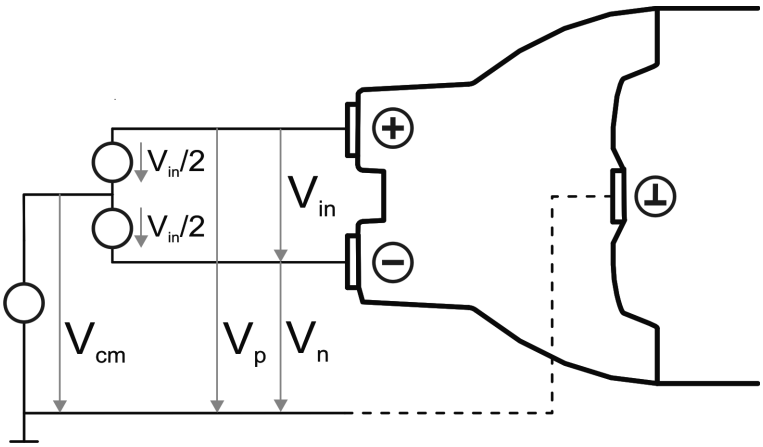
When you connect an R&S RT-ZSxx active probe to a channel input of the R&S RTP, the oscilloscope recognizes the probe. It reads the identification and calibration data from the probe box and shows the result in the "Setup" and "Probe Attributes" tabs. This data together with the deskew time for a given channel is stored and processed by the R&S RTP. If you connect the probe the next time to the same channel, the information is fetched and used.

### Differential active probes

Differential active probes are designed to measure signals that are referenced against each other, and voltages that are not references to ground, for example twisted-pair signal lines. The R&S RT-ZD probes are differential probes with high input impedance, they can be used to measure voltages between any two test points.

Compared with two-channel measurement setup with single-ended probes, the measurement with differential probes is symmetric due to the same amplification and cable length on both paths. It is also immune to interference and noise and occupies only one input channel.

A differential probe has three sockets: the positive signal socket (+), the negative signal socket (-), and the ground socket.




Multiple input voltages can be defined for a differential probe:

- Differential mode input voltage ($V_{in}$, $V_{dm}$)
  Voltage between the positive and negative signal sockets
- Positive single-ended input voltage ($V_p$)
  Voltage between the positive signal socket and the ground socket
- Negative single-ended input voltage ($V_n$)
  Voltage between the negative signal socket and the ground socket
- Common mode input voltage ($V_{cm}$)
  Mean voltage of positive and negative signal sockets referred to the ground socket, respectively

Two of these voltages are independent values, the other two can be calculated:

$$V_{in} = V_p - V_n$$

$$V_{cm} = \frac{V_p + V_n}{2}$$

R&S RT-ZD probes detect only differential input voltages and provide it to the oscilloscope. Common mode signals are suppressed by the probe. This characteristic is described by the common mode rejection ratio (CMRR):

$$CMRR = \frac{DifferentialGain}{CommonModeGain}$$

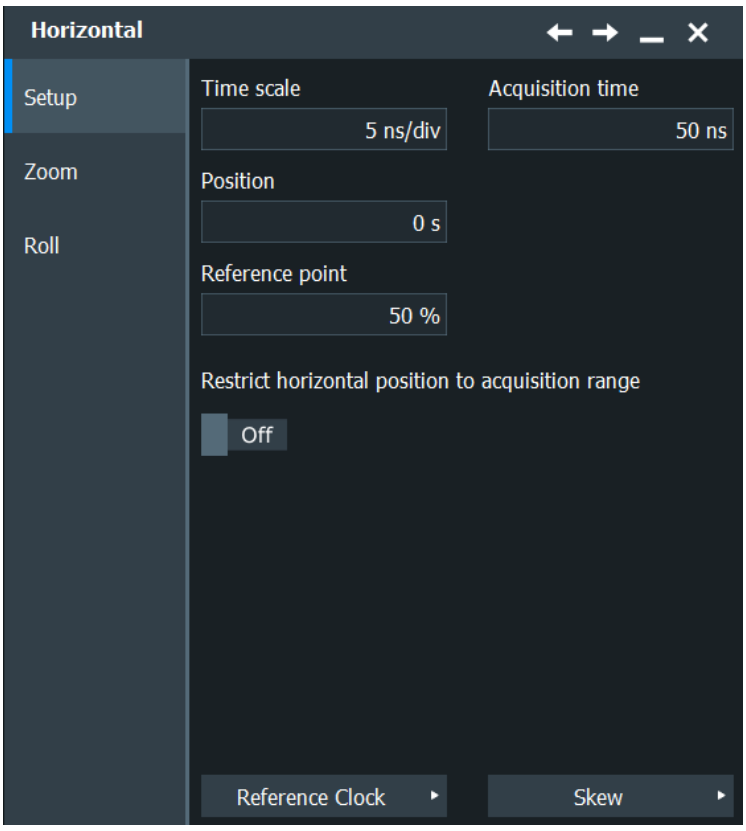
In addition, the R&S ProbeMeter of R&S RT-ZD differential probes can measure differential and common mode DC voltages. The measurement result is displayed on the oscilloscope's screen. The common mode measurement of the R&S ProbeMeter allows you to check the input voltage relative to ground. Thus, the CM measurement is a convenient way to detect breaches of the operating voltage window, and the reason of unwanted clippings.

## 5.2 Horizontal

The "Horizontal" dialog provides the time base configuration for channel and spectrum waveforms.

### 5.2.1 Setup settings

Access: "Menu" > "Horizontal" > "Setup" tab.



**Time scale**
Sets the horizontal scale for all channel and math waveforms in seconds per division. Increase the scale to see a longer time interval of the waveform. Decrease the scale to see it in more detail. The scale has a point that remains fixed on the screen when the scale value is changing - the reference point.

Remote command:
`TIMebase:SCALe` on page 1236

**Acquisition time**
Shows the time of one acquisition, that is the time across the 10 divisions of the diagram:

Acquisition time = Time scale * 10 divisions

Changing the acquisition time changes the time scale too.

Remote command:
`TIMebase:RANGe` on page 1237

**Position**
Defines the time distance between the reference point and the trigger point (the zero point of the diagram). If you want to see a section of the waveform some time before or after the trigger, enter this time as horizontal position. The requested waveform section is shown around the reference point. Use positive values to see waveform sections after the trigger - the waveform and the diagram origin move to the left.

See also "Reference point" on page 123.

Remote command:
`TIMebase:HORizontal:POSition` on page 1237

**Reference point**
Sets the position of the reference point in % of the screen. The reference point marks the rescaling center of the time scale. It is indicated by a grey triangle outline at the top of the diagram. If you modify the time scale, the reference point remains fixed on the screen, and the scale is stretched or compressed to both sides of the reference point.

Remote command:
`TIMebase:REFerence` on page 1237

**Restrict horizontal position to acquisition range**
If enabled, the horizontal position cannot be set outside the visible waveform diagram.

Remote command:
`TRIGger<m>:OFFSet:LIMited` on page 1238

**Refence Clock**
Opens a dialog for configuring the "Refence Clock" settings, see Chapter 5.3.2, "Reference clock (OCXO)", on page 127.

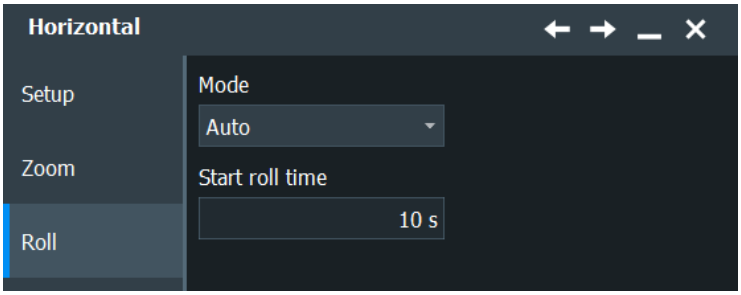
**Skew**
Opens a dialog for configuring the "Skew" settings, see Chapter 5.3.1, "Skew", on page 125.

### 5.2.2 Zoom settings

The zoom settings are described in Chapter 7.1, "Zoom", on page 261.

### 5.2.3 Roll settings

Access: "Menu" > "Horizontal" > "Roll" tab.




In this dialog, you can define the roll mode.

**Roll mode**
Selects, if the roll mode is set automatically by the instrument or if it is turned off.

In roll mode, the instrument shows the waveforms immediately, without waiting for the complete acquisition of the waveform record. If the time base is slow - at long time scale values - the roll mode saves waiting for the waveform display. The instrument displays newly acquired waveform points at the right edge of the display and moves the waveform to the left.

The roll mode has following restrictions:
- Roll mode disables persistence
- History is not available
- Event actions are not possible

The instrument activates the roll mode automatically if the following conditions are fulfilled:

- Acquisition time exceeds the defined "Minimum acquisition time"
- Waveform arithmetic is disabled ("Off")
- Only one waveform per channel is active
- All channel waveforms are set to the same decimation mode, and only to one of these values: "Sample", "Peak detect", or "High res"
- All mask tests are disabled
- Fast segmentation is disabled
- Event actions are disabled
- FFT is disabled
- All serial buses are disabled
- All digital channels are disabled (MSO option R&S RTP-B1)
- No CDR jitter data is acquired
- No zone trigger is active

The roll mode depends also on sample rate and record length. In roll mode, the sample rate limit is 2 MSa/s. At 50  s, the resulting record length limit is 100 MSa.

If the acquisition time is >50 s, the record length limit is effective, and the maximum sample rate depends on the acquisition time:

*Sample rate ≤ 100 MSa / Acquisition time*.

If the acquisition time is <50 s, the maximum sample rate in roll mode depends on the number of active channels:

*Sample rate = 2 MSa/s / Number of active wfms*.

The corresponding maximum record length is:

*Record length ≤ 2 MSa/s * Acquisition time / Number of active wfms*.

Thus, the roll mode switches off, or it does not activate automatically if:

- The record length exceeds the limit at acquisition times >50 s.
- The sample rate exceeds the limit.
- Too many waveforms are active.

Remote command:
`TIMebase:ROLL:ENABle` on page 1241
`TIMebase:ROLL:STATe?` on page 1241

**Start roll time**
The instrument can activate the roll mode automatically if the Acquisition time exceeds the value given here.

Remote command:
`TIMebase:ROLL:MTIMe` on page 1242

## 5.3 Horizontal accuracy

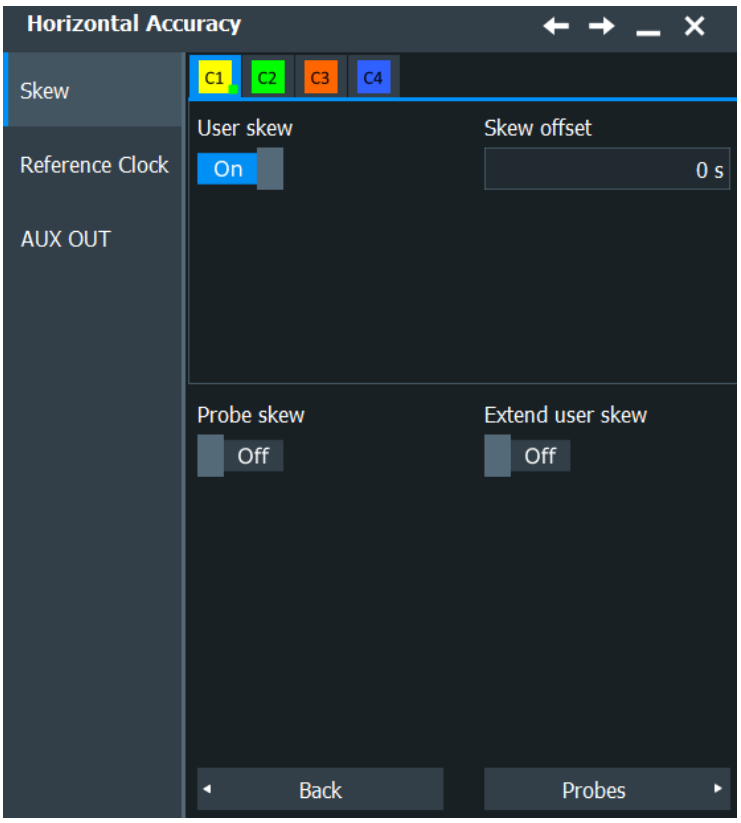
Access: "Menu" > "Horizontal" > "Setup" tab > "Skew".

The "Horizontal Accuracy" dialog box contains standard and optional settings to improve measurement and analysis accuracy and to reduce jitter effects.

### 5.3.1 Skew

Access: "Menu" > "Horizontal" > "Setup" tab > "Skew" > "Skew" tab.

Skew compensates signal propagation differences between channels caused by the different length of cables, probes, and other sources. Correct skew values are important for accurate triggering and timing relations between channels.




Make sure that the correct channel tab is selected.

**User skew**
If enabled, the "Skew offset" value is used for compensation.

Remote command:
`CHANnel<m>:SKEW:MANual` on page 1299

**Skew offset**
Sets a delay value, which is known from the circuit specifics but cannot be compensated by the instrument automatically. It affects only the selected input channel.

The offset range and possible values depend on Extend user skew.

Remote command:
`CHANnel<m>:SKEW:TIME` on page 1300

**Probe skew**
Measures the skew of all connected active probes and includes it in the total skew offset.

"Use probe group delay ..."
If enabled, the skew of all connected active probes is measured, displayed, and used for deskewing. The setting affects all active channels.

"Active probe" Shows the type of the probe that is connected to the selected channel.

"Probe group delay" Shows the result of the probe skew measurement on the selected channel.

Remote command:
`PROBe<m>:SKEState` on page 1300

**Total skew offset**
If Extend user skew is disabled, the sum of the measured "Probe group delay" and the "Skew offset" is shown. If "Use skew offset" is disabled, the skew offset is ignored.

If Extend user skew is enabled, only the probe skew is shown, and the skew offset is always ignored.

**Extend user skew**
Allows you to set higher value in "Skew offset" to compensate for the delay of the measurement setup. Without extension, the deskew range is ±100 ns, and delays shorter than the sample interval can be compensated.

With extension, the maximum delay is ±1 s, and the instrument can compensate complete samples. Compensation takes effect after the trigger. The maximum number of acquisitions in the memory is reduced.

Use the extended skew range to compare signals: One signal goes directly from the generator to the scope, and the other signal goes from the generator through the DUT to the scope. The delay of the DUT is higher than ±100 ns.

Remote command:
`CHANnel<m>:SKEW:EXTended` on page 1300

### 5.3.2 Reference clock (OCXO)

Access: "Menu" > "Horizontal" > "Setup" tab > "Skew" > "Reference clock" tab.

The oven-controlled crystal oscillator (OCXO) produces a 10 MHz internal reference signal with precise and stable frequency. With OCXO, you can also use an external reference signal. The input and output connectors for the external reference signal are on the rear panel.

**Detected**
Indicates if the OCXO is detected by the instrument.

**Oven hot**
Indicates when the oven has reached its nominal temperature and is operating with the specified accuracy.

**External reference**
Sets the frequency of an external reference input signal that is connected to the external reference input on the rear panel of R&S RTP. A frequency range from 1 MHz to 20 MHz is supported.

Remote command:
SENSe[:ROSCillator]:EXTernal:FREQuency on page 1303

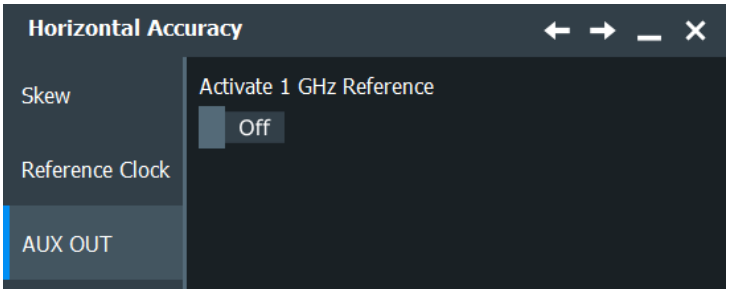
**Use external reference**
Enables the use of the external reference signal instead of the internal OCXO reference.

If an external input reference is used, no signal is output at Ref Out. Without external reference, the 10 MHz frequency of the OCXO is output at Ref Out.

Remote command:
SENSe[:ROSCillator]:SOURce on page 1303

### 5.3.3 AUX OUT

Access: "Menu" > "Horizontal" > "Setup" tab > "Skew" > "AUX OUT" tab.




**Activate 1GHz Reference**
Enables the 1 GHz reference signal and sends it to the [Aux Out] connector. The signal is required for performance test to measure the frequency of the internal calibration signal.

Remote command:
CALibration:SOURce:FREQuency on page 1301
CALibration:SOURce:STATe on page 1301

## 5.4 Acquire settings

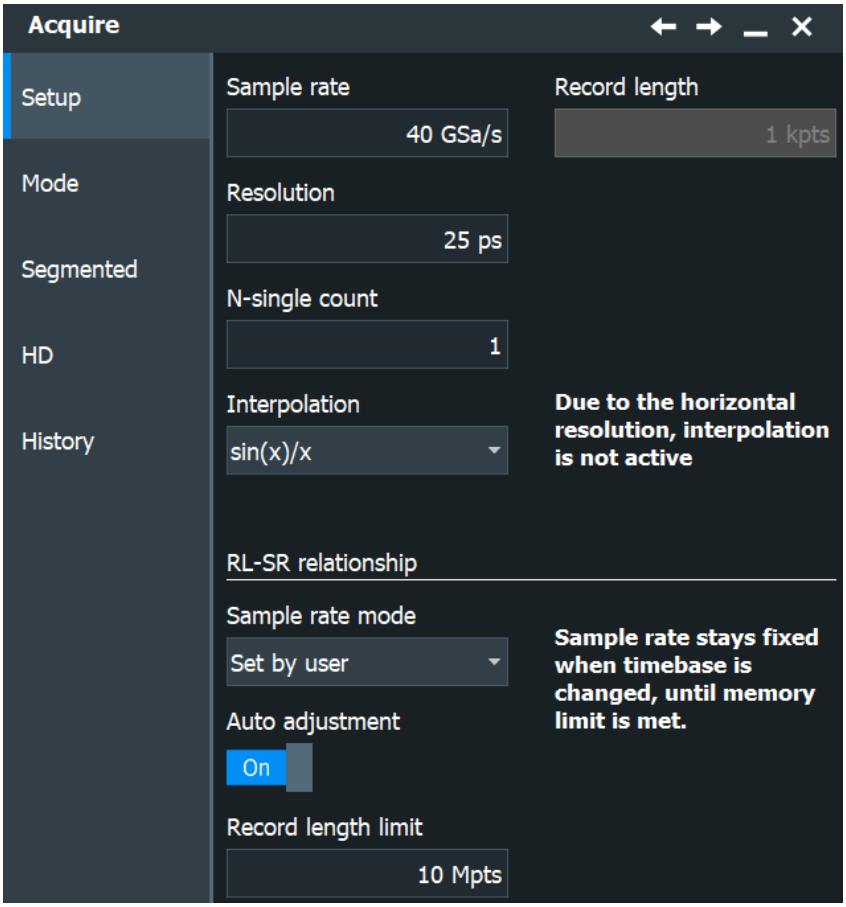
The "Acquire" dialog provides the acquisition configuration for channel and spectrum waveforms.

### 5.4.1 Setup settings

The "Setup" tab in the "Acquire" dialog provides the settings for the time axis.

For background information, see Chapter 5.1.3, "Horizontal system", on page 117.

Access: "Menu" > "Acquire" > "Setup" tab.




**Sample rate**
Sets the number of captured waveform points per second. It considers the samples of the ADC, and the reduction of waveform points by decimation.

If interpolation is not active, the sample rate is the reciprocal value of the resolution, and thus also depends on the acquisition time and the record length.

If interpolation is active, the sample rate is limited to the ADC sample rate.

See also:
- Chapter 5.1.2, "Sampling and acquisition", on page 115
- Chapter 5.1.3, "Horizontal system", on page 117

Remote command:
`ACQuire:SRReal` on page 1240

**Record length**
Indicates the number of waveform samples that build the waveform across the acquisition time.

Remote command:
`ACQuire:POINts[:VALue]` on page 1241

**Resolution**
Sets the time between two waveform samples. A fine resolution with low values produces a more precise waveform record.

Remote command:
ACQuire:RESolution on page 1240

**Average count (N-single count)**
Access:
- "Menu" > "Acquire" > "Setup" tab > "N-single count"
- "Menu" > "Acquire" > "Mode" tab > "Average count"
- "Menu" > "Acquire" > "Segmented" tab > disable "Acquire maximum" > "Required"
- [Math] > "Arithmetic" tab > "Mode" is not "Off" > "Average count"

The acquisition and average count has several effects:
- It sets the number of waveforms acquired with [Single]
- It defines the number of waveforms used to calculate the average waveform. Thus, the instrument acquires sufficient waveforms to calculate the correct average if "Average" is enabled for waveform arithmetic. The higher the value is, the better the noise is reduced.
- It sets the number of acquisitions to be acquired in a fast segmentation acquisition series. Thus, you can acquire exactly one fast segmentation acquisition series with [Single].
  If fast segmentation is enabled and configured to acquire the maximum number of acquisitions, the acquisition count is set to that maximum number and cannot be changed.
  See also "Number of acquisitions" on page 135.
- It is the "Finished" criteria for the state of a mask test.

Remote command:
ACQuire:COUNt on page 1245

**Interpolation**
Selects the interpolation method. If the defined "Sample rate" is higher than the ADC sample rate, interpolation adds points between the captured samples of the waveform by various mathematic methods.

"Linear" Two adjacent ADC sample points are connected by a straight line, the interpolated points are located on the line. You see a polygonal waveform similar to the real signal, and also the ADC sample points as vertexes.

"sin (x)/x" Two adjacent ADC sample points are connected by a sin(x)/x curve, and also the adjoining sample points are considered by this curve. The interpolated points are located on the resulting curve. This interpolation method is precise and shows the best signal curve.

"Sample/Hold" The ADC sample points are displayed like a histogram. For each sample interval, the voltage is taken from the sample point and considered as constant, and the intervals are connected with vertical lines. Thus, you see the discrete values of the ADC - the measured samples.

Remote command:
ACQuire:INTerpolate on page 1242

**Sample rate mode**
You can choose to keep constant either the sample rate or the record length when you adjust the time scale or acquisition time.

See also: Chapter 5.1.3, "Horizontal system", on page 117

| | |
|---|---|
| "Set by user" | With constant sample rate and resolution, increasing the time scale also increases the record length, and vice versa. You can limit the record length to a maximum value. |
| "Fixed record length" | With constant record length, increasing the time scale reduces the sample rate and coarsens the resolution, that is, the time between two waveform samples gets longer. |

Remote command:
`ACQuire:POINts:AUTO` on page 1238

**Auto adjustment**
Prevents undersampling and ensures a sufficient resolution to acquire the correct waveform if the time scale is changed. The setting takes effect if the changed parameter - resolution or record length - reaches a limit. The instrument automatically keeps this parameter constant at its limit, and changes the other parameter regardless of the "Resolution / Record length" setting.

See also: Record length limit

Remote command:
`ACQuire:POINts:AADJust` on page 1239

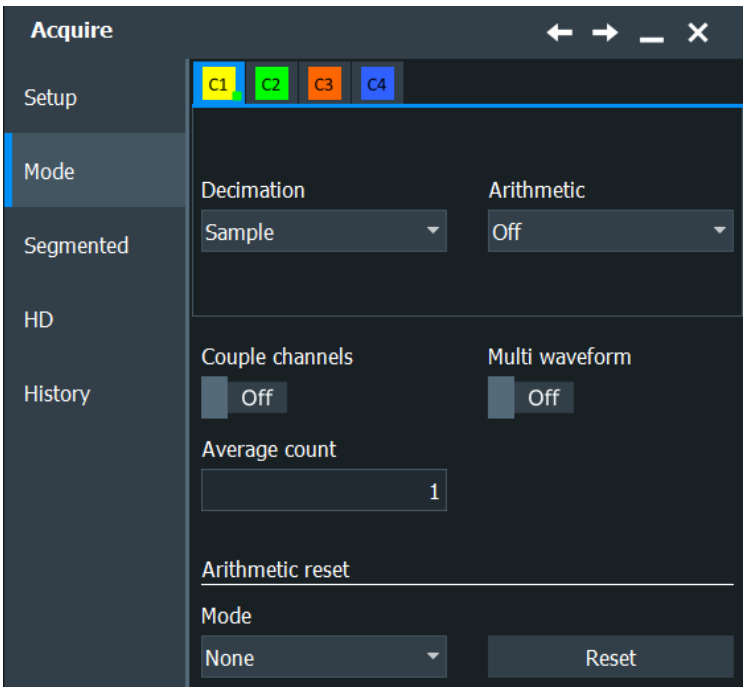
**Record length limit**
Sets a limit for the record length to prevent very large records. This value is only available if "Auto adjustment" is on and a constant resolution is selected. If you increase the time scale, the resolution remains constant, and the record length increases until the limit is reached. Further increase of the time scale changes the resolution and keeps the record length limit.

See also:
- Record length
- Auto adjustment

Remote command:
`ACQuire:POINts:MAXimum` on page 1239

## 5.4.2 Mode settings

Access: "Menu" > "Acquire" > "Mode" tab.




**Decimation**
Selects the decimation mode. Decimation reduces the data stream of the ADC to a stream of waveform points with lower sample rate and a less precise time resolution. The R&S RTP uses decimation, if the waveform "Sample rate" is less than the ADC sample rate. In this case, interpolation is not possible.

The decimation mode is waveform-specific, you can select another mode for each waveform.

There are different methods to define the recorded waveform point out of n sample points:

| | |
|---|---|
| "Sample" | One of n samples in a sample interval of the ADC is recorded as waveform point, the other samples are discarded. The time between the two adjacent waveform points is exactly the resolution. Very short glitches might remain undiscovered by this method. |
| "Peak detect" | The minimum and the maximum of n samples in a sample interval are recorded as waveform points, the other samples are discarded. |
| "High res" | The average of n sample points is recorded as one waveform sample. Averaging reduces the noise, the result is a more precise waveform with higher vertical resolution. In "High res" mode, decoding of serial protocols does not work. |
| "RMS" | The waveform point is the root mean square of n sample values. Thus, the RMS value reflects the instantaneous power. This arithmetic mode is used to average a measured power waveform. Linear averaging of power signals causes an error dependent on the noise of the signal to be averaged. |

Remote command:
`CHANnel<m>[:WAVeform<n>]:TYPE` on page 1244

**Arithmetic**
Waveform arithmetic builds the resulting waveform from several consecutive acquisitions of the signal. The arithmetic works with interpolated and decimated waveforms.

This setting is waveform-specific.

The methods are:

"Off" The data of only one acquisition is recorded according to the decimation settings. In effect, no waveform arithmetic is processed.

"Envelope" Detects the minimum and maximum values in a sample interval over several acquisitions. Each acquisition is done in the "Peak detect" decimation mode, and the most extreme values for all acquisitions build the envelope. The resulting diagram shows two envelope waveforms: the minimums (floor) and maximums (roof).
The envelope is built until the restart criterion is reached, see "Arithmetic Reset > Mode" on page 134.

**Note:** If you change from "Envelope" to "Off", make sure to set also the "Decimation" to the required value.

"Average" The average is calculated from the data of the current acquisition and several acquisitions before. The method reduces random noise and other heterodyne signals. It requires a stable, triggered and periodic signal for correct function.
The number of acquisitions for average calculation is defined with "Average count"
The "Auto reset mode" defines the restart condition.

Remote command:
`CHANnel<m>[:WAVeform<n>]:ARIThmetics` on page 1244

**Couple channels**
Sets the acquisition mode and the waveform arithmetic of all channels to the last set value.

If the acquisition settings are coupled, "Multi waveform" is not available, only one waveform per channel can be used.

Remote command:
`ACQuire:CDTA` on page 1243

**Multi waveform**
For each channel, up to three waveforms can be shown and analyzed. The decimation mode and the waveform arithmetic are specific for each waveform. So you can analyze several aspects of the signal: For example, waveform1 shows the peaks, and waveform2 shows the average of the signal.

Remote command:
`ACQuire:MUWaveform` on page 1243

**Average count (N-single count)**
Access:
- "Menu" > "Acquire" > "Setup" tab > "N-single count"
- "Menu" > "Acquire" > "Mode" tab > "Average count"
- "Menu" > "Acquire" > "Segmented" tab > disable "Acquire maximum" > "Required"

- [Math] > "Arithmetic" tab > "Mode" is not "Off" > "Average count"

The acquisition and average count has several effects:

- It sets the number of waveforms acquired with [Single]
- It defines the number of waveforms used to calculate the average waveform. Thus, the instrument acquires sufficient waveforms to calculate the correct average if "Average" is enabled for waveform arithmetic. The higher the value is, the better the noise is reduced.
- It sets the number of acquisitions to be acquired in a fast segmentation acquisition series. Thus, you can acquire exactly one fast segmentation acquisition series with [Single].
  If fast segmentation is enabled and configured to acquire the maximum number of acquisitions, the acquisition count is set to that maximum number and cannot be changed.
  See also "Number of acquisitions" on page 135.
- It is the "Finished" criteria for the state of a mask test.

Remote command:
`ACQuire:COUNt` on page 1245

**Arithmetic Reset > Mode**
Defines when the envelope and average evaluation restarts.

| | |
|---|---|
| "None" | No restart, the number of acquisitions considered by the waveform arithmetics is not limited. |
| "Time" | Restarts the envelope and average calculation after the time defined in "Time". |
| "Waveforms" | Restarts the envelope and average calculation after a number of acquired waveforms defined in "Count". |

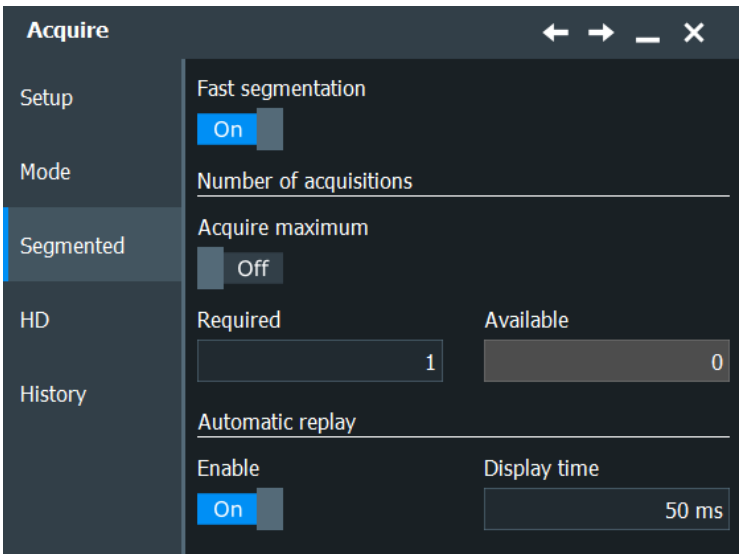
Remote command:
`ACQuire:ARESet:MODE` on page 1246
`ACQuire:ARESet:TIME` on page 1246
`ACQuire:ARESet:COUNt` on page 1246

**Reset**
Forces the immediate restart of the envelope and average calculation for all waveforms.

Remote command:
`ACQuire:ARESet:IMMediate` on page 1245

### 5.4.3 Segmented settings

Access: "Menu" > "Acquire" > "Segmented" tab.




**Fast segmentation**
Switches the fast segmentation mode on and off.

Remote command:
`ACQuire:SEGMented:STATe` on page 1247

**Number of acquisitions**
You can define the number of acquisitions to be stored in a fast segmentation acquisition series:

- Acquire the maximum number of acquisitions that can be stored in the sample memory.
  To acquire the maximum number, enable "Acquire maximum". The maximum number of acquisitions is shown in the "Required" field.
- Acquire a given number of acquisitions.
  Enter the number in the "Required" field.

The acquisition count (Average count (N-single count)) is always set to the required number of acquisitions. Thus you can acquire exactly one fast segmentation acquisition series with [Single]. The [Run / Stop] key works in the same way as [Single], it stops acquisition when the series is completed.

You can stop the running acquisition before the series is completed.

The number of acquired waveforms is shown in "Available" and can be displayed with "Show history".

Remote command:
`ACQuire:SEGMented:MAX` on page 1247

**Enable automatic replay**
If enabled, the instrument starts processing and displaying the data when the acquisition series is captured completely. Depending on the number of acquisitions, it can take some time until the acquisition series is displayed. If the setting is disabled, the instrument only captures the data and stores it in the sample memory.

Remote command:
ACQuire:SEGMented:AUToreplay on page 1247

**Display time**
Defines the display speed of the fast segmentation acquisition series. Display starts after the series has been captured completely.

## 5.4.4 High definition mode

The high definition mode offers up to 16 bits of vertical resolution. Higher vertical resolution reduces quantization noise and acquires waveforms of higher accuracy with finer details of the signal to be seen.

The number of vertical resolution bits defines the number of vertical levels that the acquisition samples are mapped to (quantization). 16 bits of resolution represent 65536 voltage quantization levels, while 8 bits of resolution represent only 256 voltage levels. The waveform values are recorded with 16 bit word length, except for peak detect decimation.

The higher vertical resolution is achieved by applying a digital lowpass filter (DSP filter) to the output of the ADC, which reduces the bandwidth of the signal. Increasing the bandwidth reduces the resulting digital resolution. The high definition is also applied to the digital trigger, thus the R&S RTP can trigger with the same high resolution with which they can display signals.

High definition can be used, for example, to measure slow pulses with high accuracy, or to analyze AM signals with very low modulation index, as used in radar.

See also:

- Chapter 5.1.1, "Vertical system", on page 112
- Chapter 5.1.2, "Sampling and acquisition", on page 115

### 5.4.4.1 High definition settings

Access: "Menu" > "Apps" > "HD"

High definition is a special acquisition mode of the oscilloscope. This mode has only one setting - the filter bandwidth.

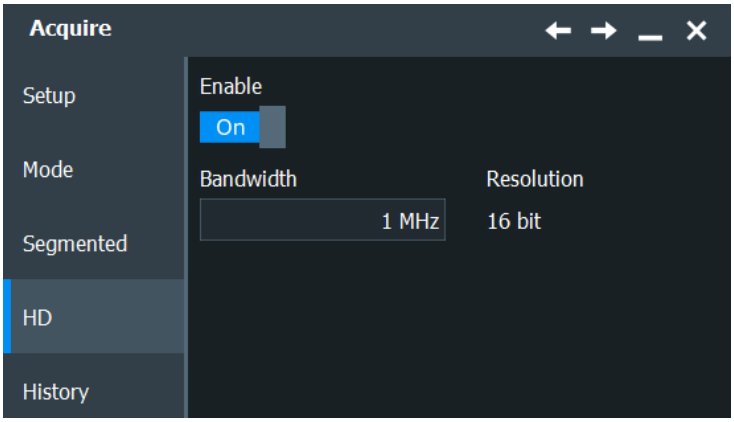



*Figure 5-5: Setting the instrument into high definition mode*

**Enable**
Enables the higher digital resolution, up to 16 bit.

Remote command:
`HDEFinition:STATe` on page 1302

**Bandwidth**
Sets the filter bandwidth for the high definition mode.

The maximum filter bandwidth is 2GHz.

Remote command:
`HDEFinition:BWIDth` on page 1302

**Resolution**
Shows the resulting vertical resolution in high definition mode. The higher the filter bandwidth, the lower the resolution. For details, refer to the R&S RTP Specifications.

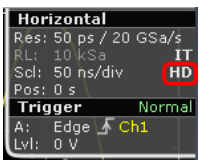
Remote command:
`HDEFinition:RESolution?` on page 1302

### 5.4.4.2 Effects of the high definition mode

The high definition mode has several effects:

**Acquisition**

The active high definition mode is indicated by "HD" in the "Acquisition" label.




The high definition mode works with half the real-time sample rate. For FFT, the instrument halves this sample rate again.

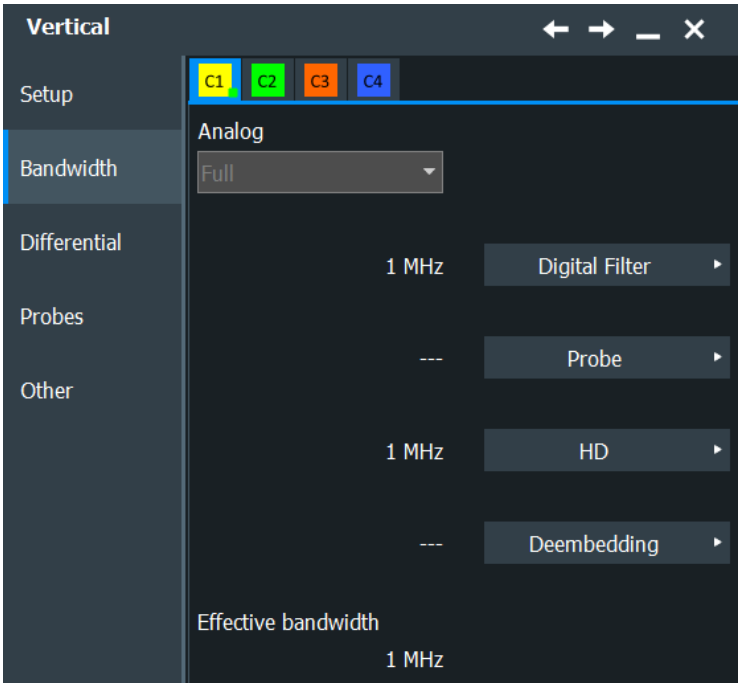
The waveform values are recorded with 16-bit word length, except for peak detect decimation (2 values with 8 bit).

### Vertical system

The current bandwidth is shown in the channel label.

In the "Vertical" > "Bandwidth" dialog (C<x>), the "Bandwidth" setting is not available because the bandwidth is set by the high definition filter.

The minimum vertical scale is 1 mV/div instead of 2 mV/div in normal mode.




### Digital filter

The digital filter settings are set automatically.

You can change the high definition "Bandwidth" in the "Acquire" > "HD" dialog.

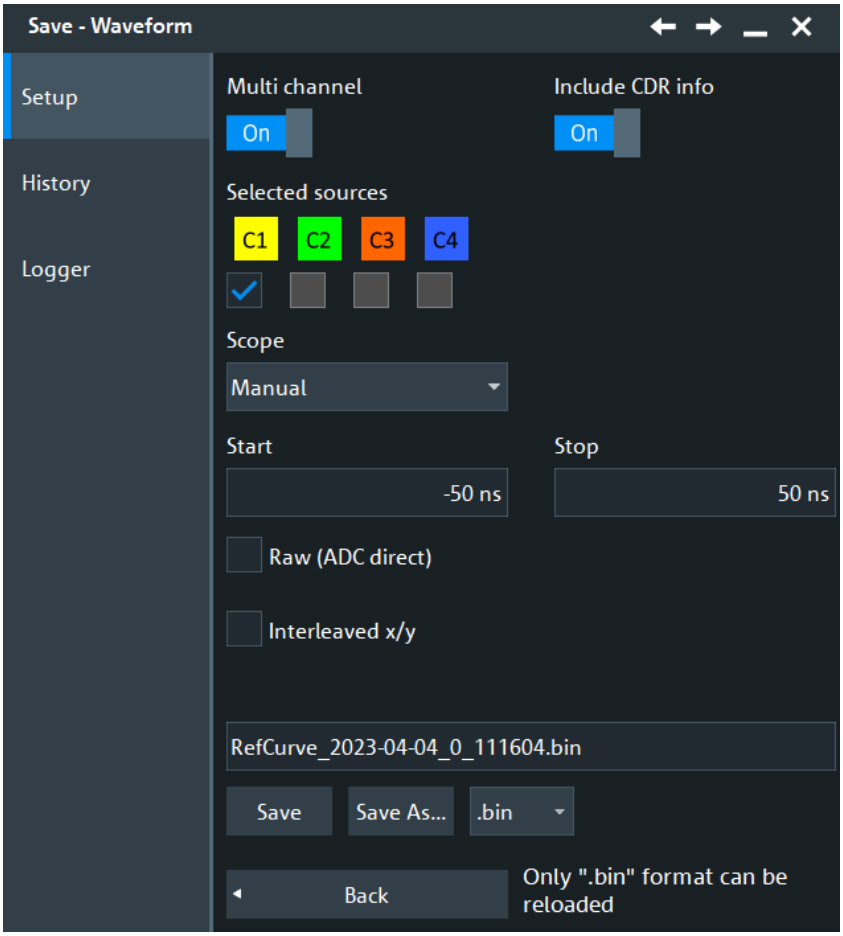
### History

Due to the 16-bit word length, the history depth is reduced, less waveforms are saved than in normal mode.

### Export

In high definition mode, waveform data in raw format is exported to file with 16-bit word length, except for peak detect decimation (2 values with 8 bit). In addition, you can define the byte order of the data words.

You can define additional export settings in the "Menu" key >"Save/Recall" key > "Save" tab > "Waveform" > "Setup" tab.




See:

- "Raw (ADC direct)" on page 493
- "Interleaved x/y" on page 494
- "Byte order" on page 1153

If you use remote control commands to transfer data to a controlling computer, set the data format to `INT,16` to transfer the complete data words (see `FORMat[:DATA]` on page 1201).

### 5.4.5 History settings

The history settings are described in Chapter 7.4.2, "History setup", on page 300.

## 5.5 Vertical setup

The "Vertical" dialog box contains all channel-dependent settings and information.

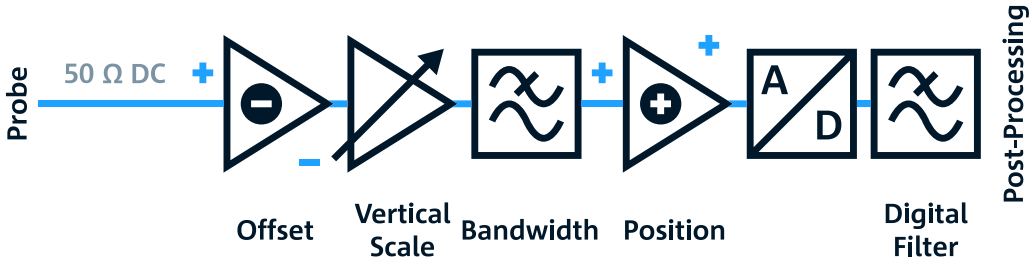
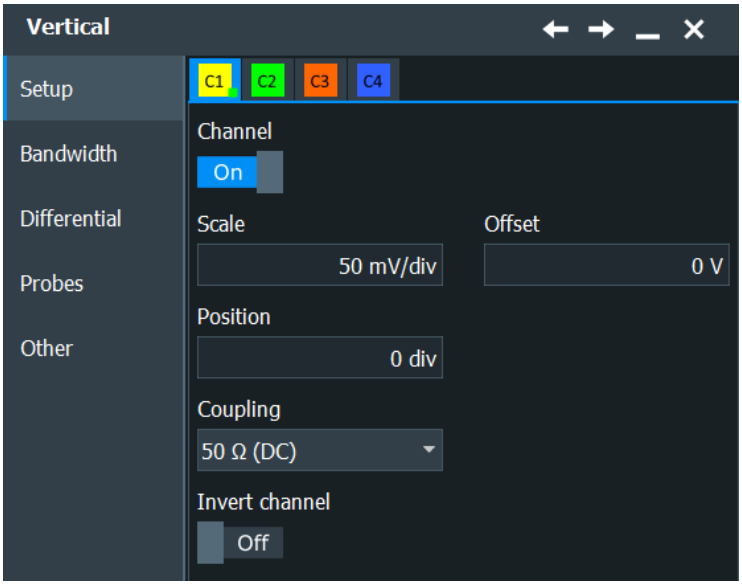



*Figure 5-6: Signal path in the vertical system of R&S RTP*

### 5.5.1 Setup settings

Access: "Menu" > "Vertical" > "Setup"

The "Setup" tab provides all basic vertical settings. The channels are listed in horizontal subtabs. Make sure to select the correct channel tab before you enter the settings.




Channel On/Off ..... 141
Scale ..... 141
Offset ..... 141

Position ..... 141
Coupling ..... 142
Invert channel ..... 142

**Channel On/Off**
Switches the channel signal on or off. The signal icon appears on the signal bar. The waveform of the last acquisition is displayed in the diagram.

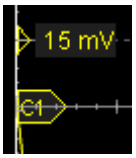
Remote command:
`CHANnel<m>:STATe` on page 1248

**Scale**
Defines the vertical scale in Volts per division. Increasing the scale compresses the display of the signal.

Remote command:
`CHANnel<m>:SCALe` on page 1249

**Offset**
The offset voltage is subtracted to correct a signal with DC offset. The vertical center of the selected channel is shifted by the offset value and the signal is repositioned within the diagram area. Negative offset values move up the waveform, positive values move it down.

The offset of a signal is determined and set by the autoset procedure. The current value is shown in the waveform label, and it is marked by a small triangle in the grid.




If a Rohde & Schwarz differential probe is connected, the offset is the differential offset.

If a Rohde & Schwarz modular probe is connected, the offset of the selected probe mode is used. For example, in CM mode, the offset is the common mode offset.

By default, the horizontal grid axis remains in the center when the offset is changed. To shift the axis together with the waveform, disable Keep Y-grid fixed in ""Settings" > Appearance > Grid".

Remote command:
`CHANnel<m>:OFFSet` on page 1250

**Position**
Moves the selected signal up or down in the diagram. The visual effect is the same as for Offset but the waveform is adjusted later in the signal flow. While the offset sets a voltage, position is a graphical setting given in divisions.

By default, the horizontal grid axis remains in the center when the offset is changed. To shift the axis together with the waveform, disable Keep Y-grid fixed in ""Settings" > Appearance > Grid".

Remote command:
`CHANnel<m>:POSition` on page 1250

**Coupling**
Selects the connection of the channel signal. The coupling determines what part of the signal is used for waveform analysis and triggering.

"DC 50 Ω" Connection with 50 Ω termination, passes both DC and AC components of the signal.

"DC 1 MΩ" Connection with 1 MΩ termination, passes both DC and AC components of the signal.

"AC" Connection with 1 MΩ termination through DC capacitor, removes DC and very low-frequency components.

If AC coupling is set, the attenuation of passive probes has no effect, and voltage is applied to the instrument with factor 1:1. Observe the voltage limits, otherwise you can damage the instrument.

Remote command:
`CHANnel<m>:COUPling` on page 1248

**Invert channel**
Turns the inversion of the signal amplitude on or off. To invert means to reflect the voltage values of all signal components against the ground level. If the inverted channel is the trigger source, the instrument triggers on the inverted signal.

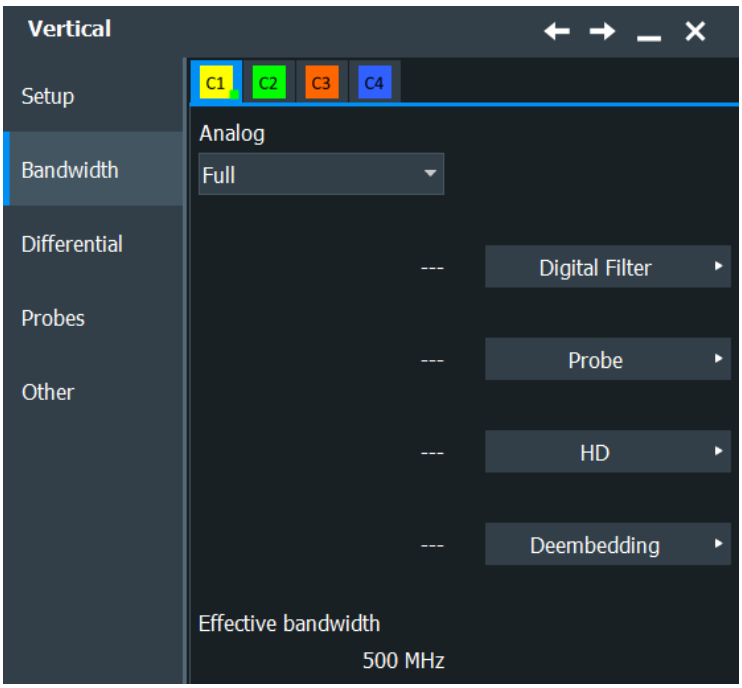
You can use inversion, for example, to switch the polarity of a differential signal without changing the probe connections.

Remote command:
`CHANnel<m>:INVert` on page 1250

### 5.5.2 Bandwidth settings

Access: "Menu" > "Vertical" > "Bandwidth"

The "Bandwidth" tab provides all settings that affect the bandwidth of the measurement system. The channels are listed in horizontal subtabs. Make sure to select the correct channel tab before you enter the settings.




**Analog**
Selects the bandwidth limit.

The specified bandwidth indicates the range of frequencies that the instrument can acquire and display accurately with less than 3dB attenuation. The probe has also a limited bandwidth and thus affects the resulting system bandwidth.

See also: Chapter 5.1.1.3, "Bandwidth", on page 114

"Full" At full bandwidth, all frequencies in the specified range are acquired and displayed. Full bandwidth is used for most applications.

"20 MHz, 200 MHz, 800 MHz"
Frequencies above the selected limit are removed to reduce noise at different levels.

Remote command:
`CHANnel<m>:BANDwidth` on page 1251

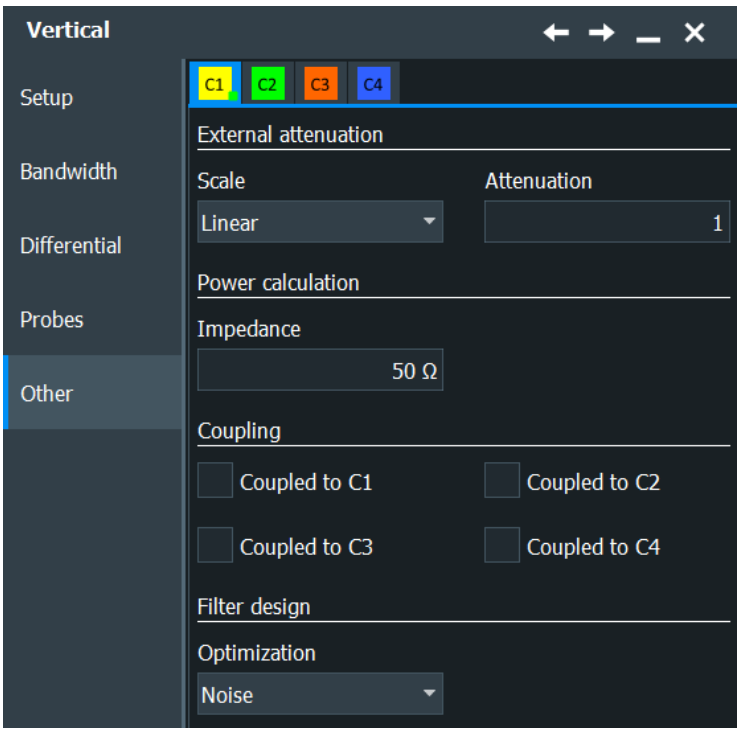
**Effective bandwidth**
Displays the effective bandwidth.

## 5.5.3 Probes settings

See Chapter 5.6, "Probes", on page 146.

## 5.5.4 Other settings

Access: "Menu" > "Vertical" > "Other"




**External attenuation: Scale, Attenuation**
Consider a voltage divider that is part of the DUT before the measuring point. The external attenuation is included in the measurement, and the instrument shows the results that would be measured before the divider. External attenuation can be used with all probes.

"Scale" Select linear or logarithmic attenuation scale.

"Attenuation" Enter the attenuation of the voltage divider according to the selected scale. The conversion from linear to logarithmic values depends on the "Vertical unit" of the probe:
For voltage-based unit (V and A):
*attenuation (dB) = 20 * $log_{10}$(attenuation factor)*
For power-based unit (W):
*attenuation (dB) = 10 * $log_{10}$(attenuation factor)*

Remote command:
`CHANnel<m>:EATScale` on page 1257
`CHANnel<m>:EATTenuation` on page 1258

**Impedance**
Sets the impedance of the channel for power calculations and measurements.

Remote command:
`CHANnel<m>:IMPedance` on page 1252

**Coupled to C1 / C2 / C3 / C4**
Channel coupling sets the vertical settings of the coupled channels to the values of the active channel. If you want to have the same vertical settings for two or more channels, you can set them at once by coupling these channels.

Channel coupling affects all vertical settings that are adjusted in the "Settings" tab: vertical scale, position, offset, coupling, and ground.

Remote command:
`CHANnel<m>:CPLing` on page 1251

### 5.5.5 Digital filter setup

Access: "Menu" > "Vertical" > "Bandwidth" tab > "Digital Filter"

After processing by the A/D converter, the channel and trigger signals are digitized signals. These digitized signals can be filtered to reject high frequency - also known as Digital Signal Processing (DSP). You can filter the acquisition channels.

One filter is applied to one channel.

In high definition mode, digital filter settings are enabled automatically. You can change the high definition bandwidth in the "Acquire" > "HD" tab. The value is assigned to all channels and shown in the "Digital Filter" dialog.

**BW limit**
Enables the digital filter for the correspondig input channel.

Remote command:
`CHANnel<m>:DIGFilter:STATe` on page 1298

**Acquisition BW**
Sets the limit frequency of the lowpass filter for each input channel individually.

Remote command:
`CHANnel<m>:DIGFilter:CUToff` on page 1298

**Trigger BW**
Shows the BW limit for the trigger signal, i.e. for the channel that is the trigger source. The trigger BW is set automatically.

Remote command:
`TRIGger<m>:RFReject?` on page 1299

## 5.6 Probes

With R&S RTP digital oscilloscopes, you can use various probe types. Mostly these probes are passive and active voltage probes. The "Probes" dialog provides all probe-relevant information.

The instrument can detect many probes and read out the probe-specific parameters, for example, bandwidth and attenuation.

In the "Probes" tab, you find all settings that are relevant for the connected probe.

Access: "Menu" > "Vertical" > "Probes"

The functionality on the "Probes" tab changes according to the type of the attached probe. Probes with Rohde & Schwarz probe interface (probe box), and also many other passive voltage probes, are recognized by the instrument. The R&S RTP reads out the main characteristics of the probe and displays them. Other probes cannot be detected, but their characteristics are known to the instrument. These known probes are called "Predefined probes". Probes that are not recognized automatically and not predefined are unknown probes, they require manual setting of measurement unit and attenuation.

Before you adjust the settings, select the correct channel tab.

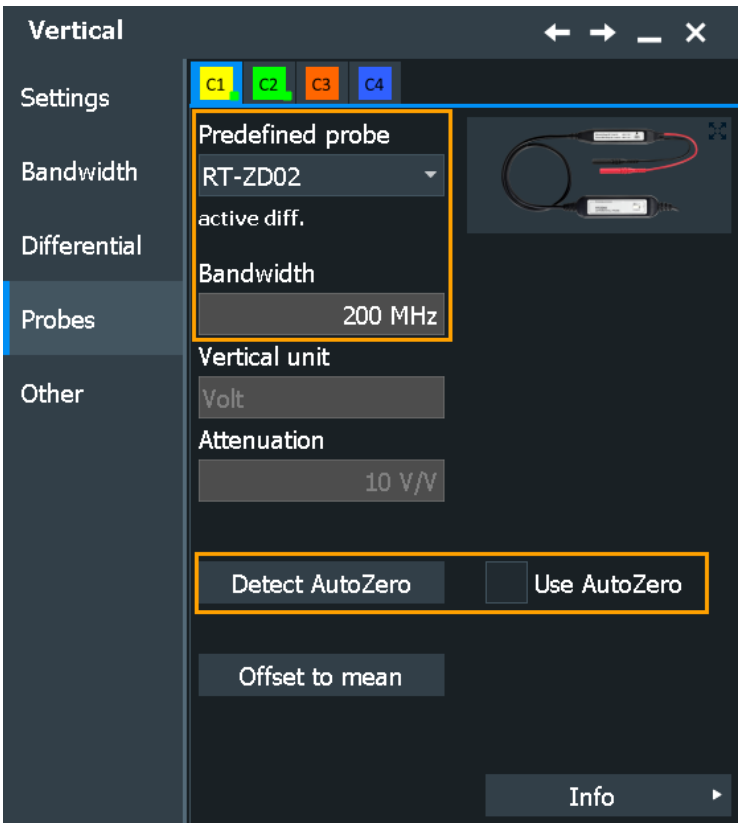
Additional information is given in the "Probe Attributes" and "Calibration Results" tabs. For background information, see Chapter 5.1.4, "Probes", on page 118.

This chapter has the following sections:

- Shared probe settings....146
- Settings for the R&S probe interface (voltage probes)....148
- Setup for passive probes....151
- Setup for active voltage probes....153
- Modular probes....158
- Setup for predefined probes....163
- Setup for current probes....165
- Setup for unknown probes....168
- Probe attributes....169
- Calibration results....170
- Probe adapter R&S RT-Z1M....170

### 5.6.1 Shared probe settings

Some of the settings in the "Probes" tab are available for all probes.




An external attenuation can be set on the "Other" tab, see "External attenuation: Scale, Attenuation" on page 144.

The shared probe settings are:

Type, Name, Bandwidth ..... 147
Detect AutoZero, Use AutoZero ..... 147

**Type, Name, Bandwidth**
The fields show the characteristics of a recognized or predefined probe for information. If the instrument cannot recognize the probe, and the probe is not known, the "Type" is "None", and the other fields are empty.

Remote command:
`PROBe<m>:SETup:TYPE?` on page 1255
`PROBe<m>:SETup:NAME?` on page 1256
`PROBe<m>:SETup:BANDwidth?` on page 1256

**Detect AutoZero, Use AutoZero**
Differences in DUT and oscilloscope ground levels can cause larger zero errors, which affect the waveform. If the DUT is ground-referenced, the AutoZero function corrects the zero error of the probe to optimize measurement results at small signal levels. The validation limit depends on the probe attenuation because probes with high attenuation often have to compensate high offsets. AutoZero detects offset values even when the signal is out of the current measurement range.

To correct the zero error of voltage probes, short the signal pin and the ground pin together and connect them to the ground of the DUT. Then tap "Detect AutoZero". While the alignment is running, the instrument switches to DC coupling to display the waveform correctly.

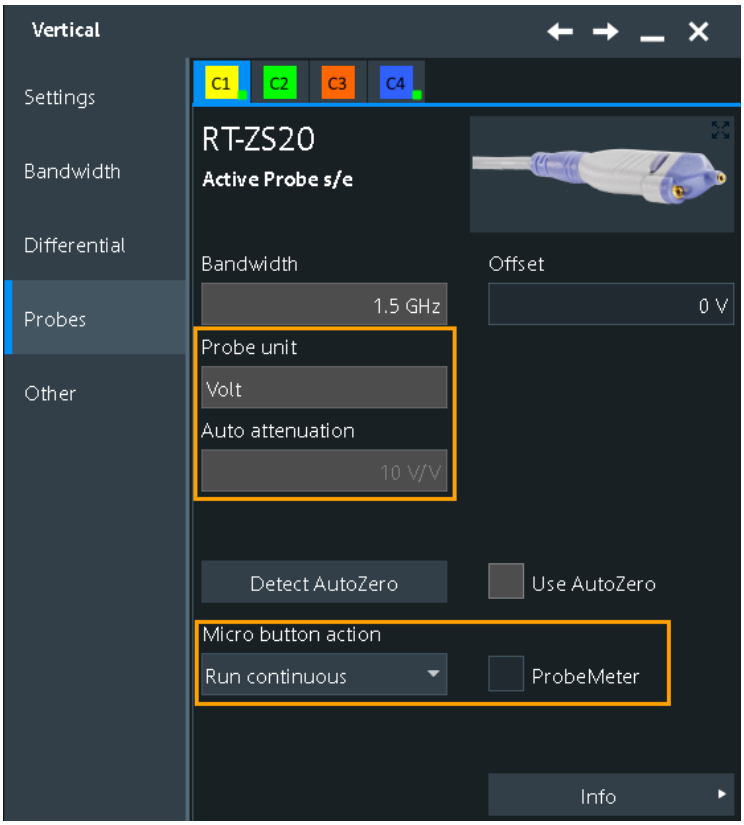
To include the measured offset in measurement results, enable "Use AutoZero".

If a current probe is connected, the function demagnetizes the probe's sensor head and sets the waveform to zero position. See "Detect AutoZero" on page 168.

Remote command:
`PROBe<m>:SETup:OFFSet:AZERo` on page 1257
`PROBe<m>:SETup:OFFSet:USEautozero` on page 1257

### 5.6.2 Settings for the R&S probe interface (voltage probes)

Active voltage probes with Rohde & Schwarz probe interface provide special features: the micro button and the ProbeMeter. Furthermore, the R&S RTP can read out the attenuation of the probe.



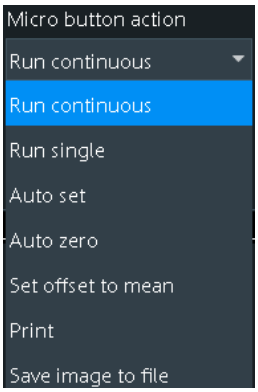

The settings for active voltage probes with Rohde & Schwarz probe interface are:

Probe unit, Auto attenuation ..... 149
Micro button action ..... 149
ProbeMeter ..... 150

**Probe unit, Auto attenuation**
If the probe is recognized by the R&S RTP, the instrument reads the attenuation unit and value from the probe an displays them.

Remote command:
`PROBe<m>:SETup:ATTenuation[:AUTO]?` on page 1256

**Micro button action**
Active voltage probes with Rohde & Schwarz probe interface have a configurable micro button on the probe head. Pressing this button, you start an action on the instrument directly from the probe. The button is disabled during internal automatic processes, for example, during self-alignment, autoset, and find level.




Select the action that you want to start from the probe:

"Run Continuous"
: Default assignment. Starts or stops the acquisition (same as Run / Stop key).

"Run single"
: Starts a defined number of acquisitions (same as [Single] key).

"Auto set"
: Starts the autoset procedure (same as Autoset key).

"AutoZero"
: Starts an auto zero measurement, see "Detect AutoZero, Use AutoZero" on page 147.

"Set offset to mean"
: Performs an automatic compensation for a DC component of the input signal using the result of a background mean measurement. See: "Offset to mean" on page 165.

"Save image to file"
: Saves the current display as image according to the image settings. See also Chapter 12.3, "Screenshots", on page 507.

"No action"
: Select this option to prevent unwanted actions due to unintended usage of the micro button.

"Find trigger level"
: Sets the trigger level automatically to *0.5 * (MaxPeak – MinPeak)*.

"Create report"
: Creates and saves a report using the report settings. See also Chapter 12.4.1, "Report settings", on page 512.

"Probe mode" Only available if a R&S RT-ZM modular probe is connected. Sets the measurement mode of the modular probe. See also "Probe mode" on page 159.

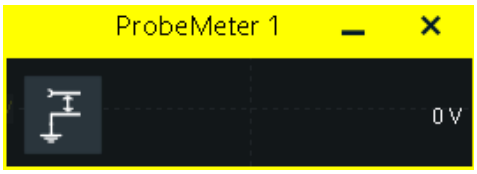
"Probe Setup" Opens the "Probes Setup" dialog box.

Remote command:
`PROBe<m>:SETup:MODE` on page 1258

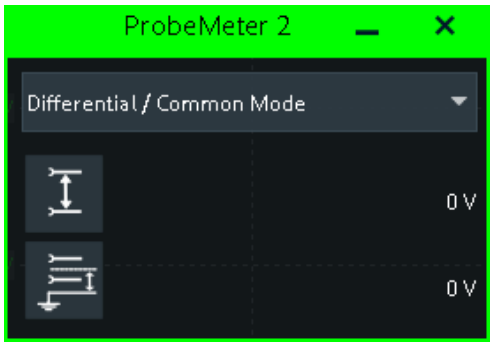
**ProbeMeter**

The integrated R&S ProbeMeter of active voltage probes with Rohde & Schwarz probe interface is a voltmeter. It measures DC voltages between the probe tip and ground connection or between the probe tips with very high precision. The R&S ProbeMeter enables ground-referenced measurements of voltages. The measurement is performed continuously and in parallel to the measurements of the oscilloscope.

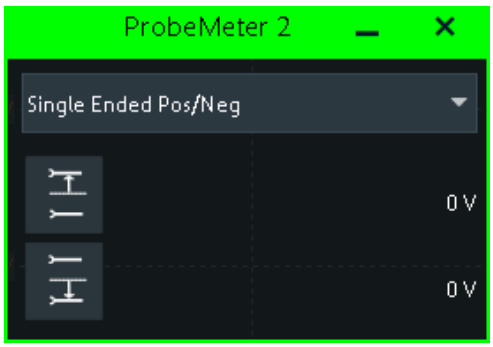
- **"Probemeter"**
  Select "Probemeter" to activate the integrated R&S ProbeMeter of active R&S probes. The measured voltages are displayed in the "ProbeMeter" result box on the screen.
- **ProbeMeter measurement results of single-ended active R&S probes**
  Measures the voltage between the probe tip and the ground.




- **ProbeMeter measurement results of differential and modular R&S probes**
  You can select the voltage to be measured by the differential active probe:
  – "Differential / Common Mode":
  Differential voltage is the voltage between the positive and negative signal sockets.
  Common mode voltage is the mean voltage between the signal sockets and the ground socket. It measures the voltage level relative to ground, for example, to check the operating voltage window.




  – "Single Ended Pos/Neg": Measures the voltage between the positive/negative signal socket and the ground.




The ProbeMeter always measures the common mode and differential voltages. Single-ended voltages are calculated values:
$V_p = V_{cm} + 0.5 * V_{in}$ and $V_n = V_{cm} - 0.5 * V_{in}$

Remote command:
PROBe<m>:PMETer:VISibility on page 1260
PROBe<m>:SETup:DISPlaydiff on page 1260
PROBe<m>:PMETer:RESults:SINGle? on page 1260
PROBe<m>:PMETer:RESults:POSitive? on page 1262
PROBe<m>:PMETer:RESults:NEGative? on page 1262
PROBe<m>:PMETer:RESults:DIFFerential? on page 1261
PROBe<m>:PMETer:RESults:COMMon? on page 1261

### 5.6.3 Setup for passive probes

Passive probes are the most widely used probes for oscilloscope measurements. Passive probes require compensation.

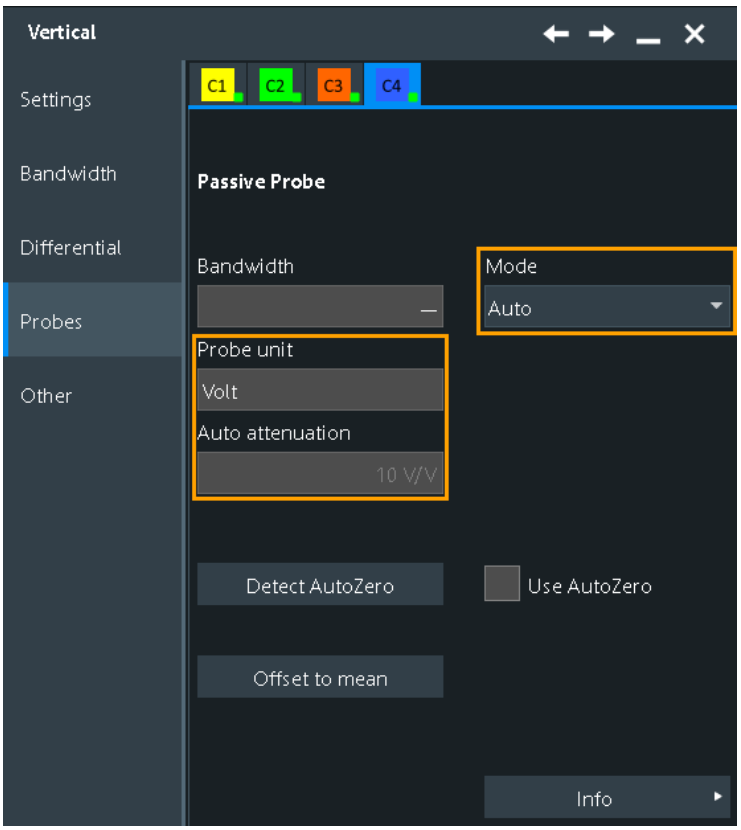



*Figure 5-7: Probe setup for passive probe*

The following shared probe settings are available:

- "Type, Name, Bandwidth" on page 147
- "Detect AutoZero, Use AutoZero" on page 147

If a passive probe is connected, the probe attenuation is read out and shown in the "Setup" tab:

- "Probe unit, Auto attenuation" on page 149

If you need to change the unit or attenuation, change the "Mode" to "Manual" and enter the correct values.

Mode..... 152
Vertical unit, Attenuation, Gain..... 153

**Mode**
Defines how the attenuation of a passive probe is set.

| | |
|---|---|
| "Auto" | The instrument uses the values that are read out from the probe. |
| "Manual" | You can define the attenuation unit and value.<br>See: "Vertical unit, Attenuation, Gain" on page 153 |

Remote command:
`PROBe<m>:SETup:ATTenuation:MODE` on page 1263

**Vertical unit, Attenuation, Gain**
If a predefined probe is connected and selected, the attenuation or gain values are shown.

For unknown probes and passive probes in manual mode, you can set user-defined values for unit, gain and attenuation.

If AC coupling is set, the attenuation of passive probes has no effect, and voltage is applied to the instrument with factor 1:1. Observe the voltage limits, otherwise you can damage the instrument.

Remote command:
`PROBe<m>:SETup:ATTenuation:UNIT` on page 1263
`PROBe<m>:SETup:ATTenuation:MANual` on page 1263
`PROBe<m>:SETup:GAIN:MANual` on page 1264

### 5.6.4 Setup for active voltage probes

Active voltage probes with Rohde & Schwarz probe interface have an integrated data memory that contains identification data and individual probe correction parameters. The R&S RTP can detect these probes and read out the data. Furthermore, these probes have a micro button and a ProbeMeter.

Active voltage probes that are offered by Rohde & Schwarz but not equipped with a Rohde & Schwarz probe interface are known to the R&S RTP as predefined probes, see Chapter 5.6.6, "Setup for predefined probes", on page 163.

The following shared probe settings are available:

- "Type, Name, Bandwidth" on page 147
- "Detect AutoZero, Use AutoZero" on page 147

Special features of the Rohde & Schwarz probe interface are described in these sections:

- "Probe unit, Auto attenuation" on page 149
- "Micro button action" on page 149
- "ProbeMeter" on page 150

The specific settings of Rohde & Schwarz active probes are described in the following chapters:

- R&S RT-ZS single-ended probes....154
- R&S RT-ZD differential probes....155
- R&S RT-ZPR power rail probes....156
- R&S RT-ZHD high-voltage differential probes....157

#### 5.6.4.1 R&S RT-ZS single-ended probes

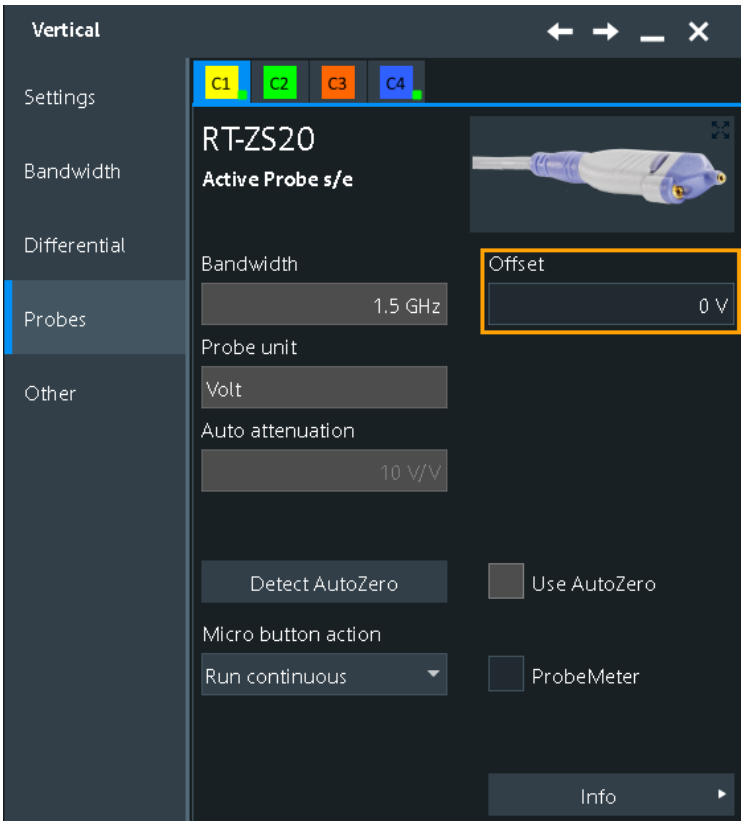



*Figure 5-8: Probe setup for active single-ended probe R&S RT-ZS20*

The only setting for R&S RT-ZS probes is the channel offset. See "Offset" on page 141.

### 5.6.4.2 R&S RT-ZD differential probes

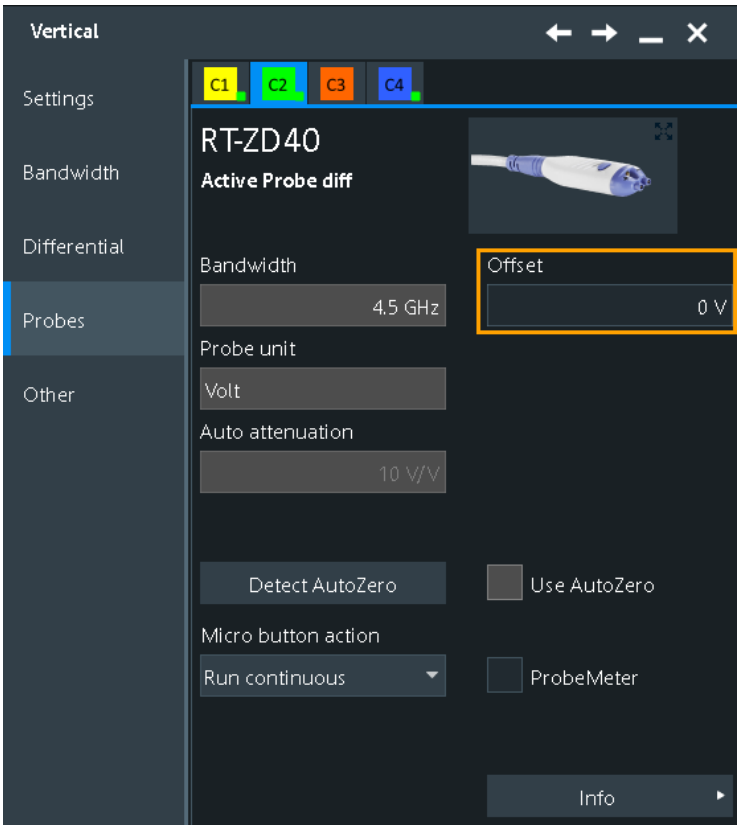



*Figure 5-9: Probe setup for active differential probe R&S RT-ZD40*

The offset is the differential offset. See "Offset" on page 141.

Specific settings for R&S RT-ZD probes are the following:

**CM offset**
Sets the common-mode offset to compensate for a common DC voltage applied to both input sockets (referenced to the ground socket). Offset compensation is particularly helpful for measurements on differential signals with high common mode levels, for example, current measurements using a shunt resistor. You can measure the common mode input voltage using the R&S ProbeMeter.

The setting is available for Rohde & Schwarz differential probes, and for modular probes in DM or CM mode (see "DM Offset, CM Offset, P Offset, N Offset" on page 160).

Remote command:
`PROBe<m>:SETup:CMOFfset` on page 1264

**Probe attenuator RT-ZA15**
If you use the external attenuator R&S RT-ZA15 together with one of the differential active probes R&S RT-ZD10/20/30, enable RT-ZA15 to include the external attenuation in the measurements.

Remote command:
PROBe<m>:SETup:ZAXV on page 1265

### 5.6.4.3 R&S RT-ZPR power rail probes

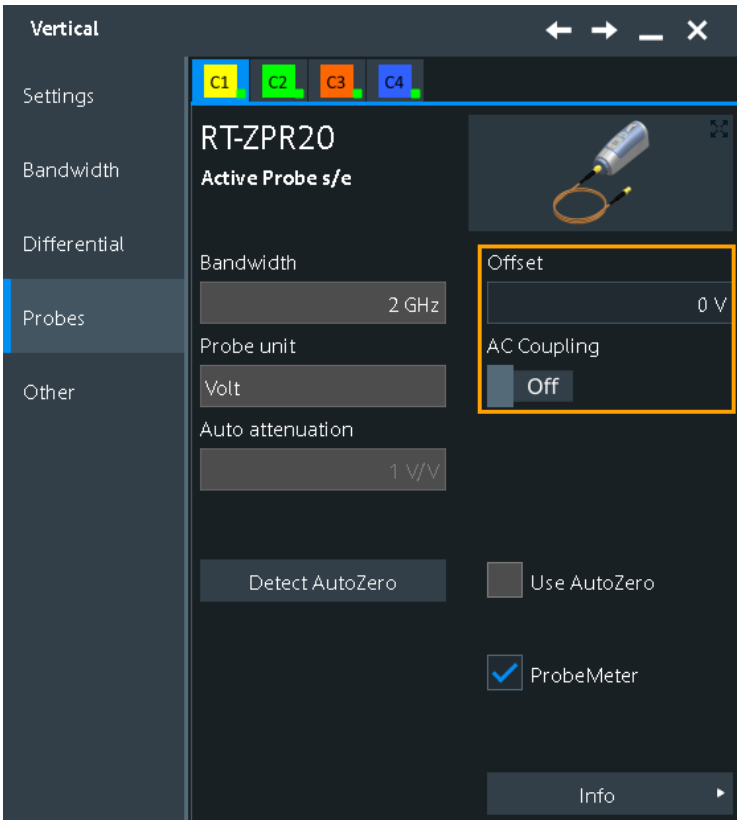



*Figure 5-10: Probe setup for power rail probe R&S RT-ZPR*

The offset is the channel offset. See "Offset" on page 141.

If the ProbeMeter is active, an additional function is provided in the "ProbeMeter" result box:

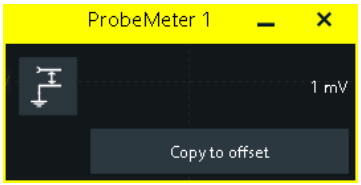



*Figure 5-11: ProbeMeter result box for*

Specific settings for R&S RT-ZPR probes are the following:

**AC Coupling**
Enables AC coupling in the R&S RT-ZPR power rail probes, which removes DC and very low-frequency components. The R&S RT-ZPR probe requires 50 Ω input termination, for which the channel AC coupling is not available. The probe setting allows AC coupling also at 50 Ω inputs.

Remote command:
`PROBe<m>:SETup:ACCoupling` on page 1265

**Copy to offset**
Sets the measured ProbeMeter value as offset. Thus, the value is considered in measurements.

Remote command:
`PROBe<m>:SETup:ADVanced:PMToffset` on page 1265

### 5.6.4.4 R&S RT-ZHD high-voltage differential probes

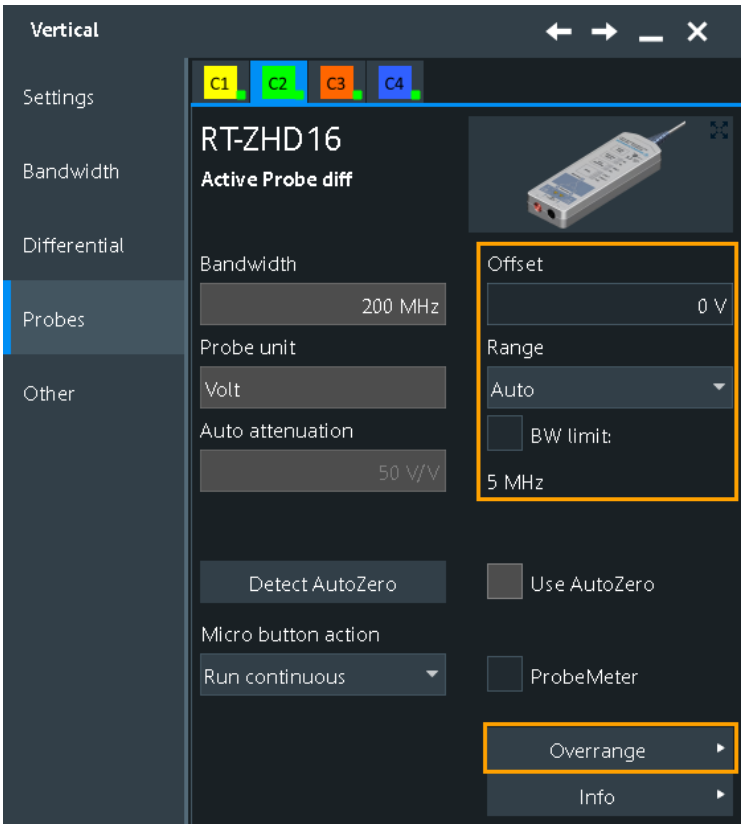



*Figure 5-12: Probe setup for R&S RT-ZHD probes*

The offset is the differential offset. See "Offset" on page 141.

Specific settings for R&S RT-ZHD probes are the following:

**Range**
Sets the voltage range of a R&S RT-ZHD probe. You can set the range on the probe control box or at the oscilloscope.

"Auto" The voltage range is set only at the oscilloscope with "Vertical scale".

"Low" Sets the lower voltage range of the connected probe. The selected value is shown in "Auto Attenuation".

"High" Sets the higher voltage range of the connected probe. The selected value is shown in "Auto Attenuation".

Remote command:
`PROBe<m>:SETup:ADVanced:RANGe` on page 1266

**BW limit**
Activates the lowpass filter in the probe control box and displays the used limit. You can also set the filter directly on the probe control box.

Remote command:
`PROBe<m>:SETup:ADVanced:FILTer` on page 1266

**Buzzer**
Activates the acoustic overrange warning in the probe control box. You can also activate the sound directly on the probe control box.

Remote command:
`PROBe<m>:SETup:ADVanced:AUDioverload` on page 1267

**Negative, Positive, Differential Overrange**
The color turns red if the voltage exceeds the probe range. The indicators are also available on the probe control box.

### 5.6.5 Modular probes

The probes of the R&S RT-ZM family are modular probes. They have a probe head and a probe amplifier connected by a cable, and various probe tip modules and tip cables for different applications. R&S RT-ZM probes are equipped with Rohde & Schwarz probe interface, and provide special features: ProbeMeter, micro button, and a wide offset compensation range.

If you connect the R&S RT-ZM probe to the external trigger input, you can adjust only the trigger level and the probe mode. The instrument sets the trigger level to the probe offset value (DM, CM, P or N, depending on the probe mode).

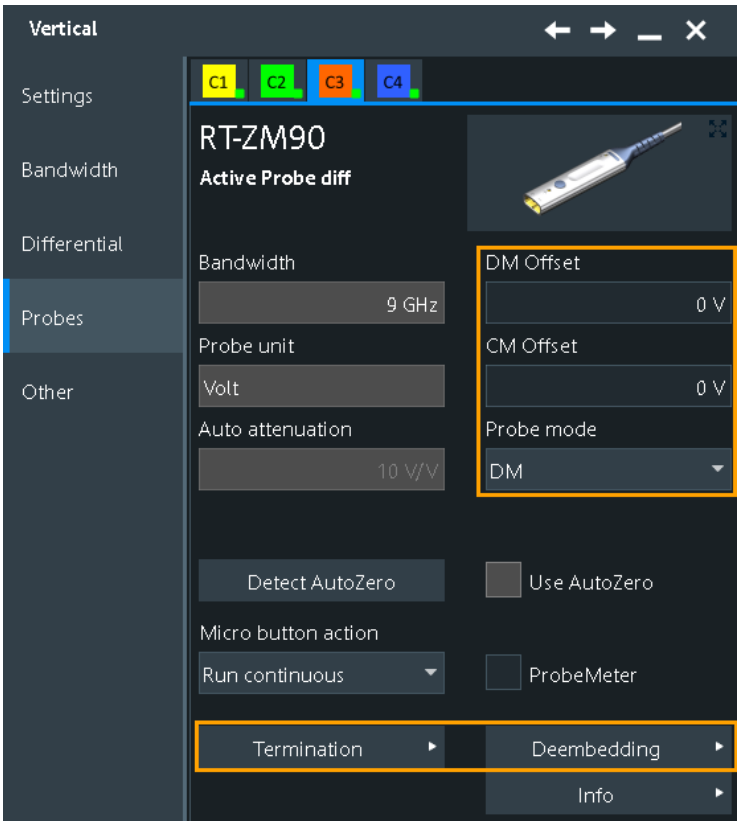
The following shared probe settings are available:

- "Type, Name, Bandwidth" on page 147
- "Detect AutoZero, Use AutoZero" on page 147

Special features of the Rohde & Schwarz probe interface are described in these sections:

- "Probe unit, Auto attenuation" on page 149
- "Micro button action" on page 149

- "ProbeMeter" on page 150




- Setup parameters of modular probes.....159
- Deembedding for modular probes.....160
- Termination voltage with R&S RT-ZMA40 SMA module.....162

### 5.6.5.1 Setup parameters of modular probes

The basic setup parameters of all modular probes are the measurement mode and the offset settings.

Access: "Menu" > "Vertical" > "Probes"

Make sure to adjust also the deembedding settings under "Probe Deembedding" (see Chapter 5.6.5.2, "Deembedding for modular probes", on page 160).

**Probe mode**
Sets the measurement mode of modular probes.

The modular probes of the R&S RT-ZM family have a multi-mode function. You can switch between single-ended, differential and common mode measurements without reconnecting or resoldering the probe. You can set the probe mode in the dialog box, and you can assign the probe mode setting to the micro button.

If you use the R&S RT-ZMA30 browser module, only DM measurements are possible because this module has no ground connector.

The measurement modes are:

"DM" Differential mode input voltage ($V_{dm}$), the voltage between the positive and negative input terminal.

$$V_{dm} = V_p - V_n$$

"CM" Common mode input voltage ($V_{cm}$), the mean voltage between the positive and negative input terminal vs. ground.

$$V_{cm} = \frac{V_p + V_n}{2}$$

"P" Positive single-ended input voltage ($V_p$). The voltage between the positive input terminal and ground.

"N" Negative single-ended input voltage ($V_N$). The voltage between the negative input terminal and ground.

Remote command:
`PROBe<m>:SETup:PRMode` on page 1267

**DM Offset, CM Offset, P Offset, N Offset**
Compensate offset voltages. Available offsets depend on the selected probe mode.

The offset of the selected probe mode is used as channel offset and considered automatically for correction. For example, in CM mode, the common mode offset is used as channel offset. See also: "Offset" on page 141.

"DM Offset" Compensates a DC voltage applied between the positive ($V_p$) and the negative ($V_n$) input terminal at the probe tip.

"CM Offset" Compensates a DC voltage applied to both input terminals referenced to ground. See also: "CM offset" on page 155.

"P Offset" Compensates a DC voltage applied to the positive input terminal ($V_p$) referenced to ground.

"N Offset" Compensates a DC voltage applied to the negative input terminal ($V_n$) referenced to ground.

Remote command:
`PROBe<m>:SETup:DMOFfset` on page 1268
`PROBe<m>:SETup:CMOFfset` on page 1264
`PROBe<m>:SETup:NOFFset` on page 1269
`PROBe<m>:SETup:POFFset` on page 1269

### 5.6.5.2 Deembedding for modular probes

Deembedding removes the parasitic effects of the measurement setup from the measured signal. These effects are typically increasing when signal frequency increases. Thus, deembedding is useful or even necessary when measuring signals of 4 GHz frequency or higher.

Access: "Vertical" menu > "Probe Setup" > "Probe Deembedding"

**Select probe tip module, Pin spacing**
Selects the tip module that is used for measurement.

If R&S RT-ZMA40 is selected, choose also the used submodule: semi-rigid cables or none.

For the browser module R&S RT-ZMA30, measure the space between the pins, and select the appropriate value.

Remote command:
`PROBe<m>:DEEMbedding:TIPModule[:SELect]` on page 1271
`PROBe<m>:DEEMbedding:TIPModule:ZMA<n>:SUBModule` on page 1272

**Probing, Terminating**
Sets the probing mode.

Select "Probing" if you use a high-ohmic probe and measure on an existing line in parallel to the load.

Select "Terminating" if the measuring equipment is the load of the line.

Remote command:
`PROBe<m>:DEEMbedding:MODE` on page 1273
`DEEMbedding<m>:COMPonent<n>:MODE` on page 2438

**Effective bandwidth**
Sets the maximum bandwidth for probe deembedding until which the signal is corrected. This maximum value is the minimum bandwidth value of the probe bandwidth, tip module bandwidth, and oscilloscope bandwidth. It cannot be higher than the highest frequency in a used S-parameter file.

Consider that most tip modules support the full bandwidth of the probe, but some tip modules have limited bandwidth.

If deembedding option R&S RTP-K121 is active, the instrument uses the effective bandwidth that is set in the "Deembedding" > "Setup" dialog box.

Remote command:
`PROBe<m>:DEEMbedding:BANDwidth` on page 1272

**Probe mode**
See "Probe mode" on page 159.

**Remove probe load**
If "Probing" is selected, you can remove the loading of the probe.

If the probe loading is removed, you see the signal that would be at the measurement point if the probe's input impedance is ideal (infinite impedance).

Without removing the loading, you see the real signal at the measurement point, including the probe loading.

Remote command:
`PROBe<m>:DEEMbedding:REMProbeload` on page 1273
`DEEMbedding<m>:COMPonent<n>:LOAD:REMProbeload` on page 2439

**Source impedance**
Shows the source impedance of the DUT. If an R&S RT-ZM probe is connected, the value depends on the selected probe mode: common mode, differential, or single-ended measurement.

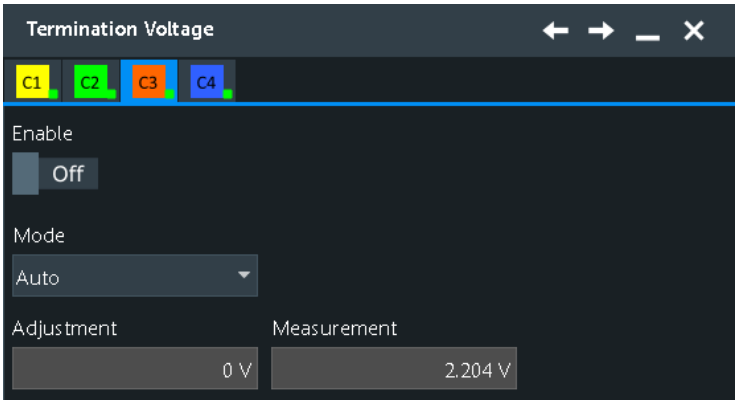
Remote command:
DEEMbedding<m>:COMPonent<n>:LOAD:IMPedance? on page 2439
PROBe<m>:DEEMbedding:LOAD:IMPedance on page 1274

### 5.6.5.3 Termination voltage with R&S RT-ZMA40 SMA module

Termination voltage is relevant if you use the R&S RT-ZMA40 SMA module. The SMA module applies a termination voltage (±4 V) to the DUT to enable measurements against a common mode DC voltage instead of ground. This measurement is required for many digital signal standards.

The termination voltage can be controlled by the oscilloscope. Therefore, connect the $V_T$ terminal of the R&S RT-ZM probe amplifier to the $V_T$ terminal of the R&S RT-ZMA40 SMA module using the red DC lead (see R&S RT-ZM User Manual). The required termination voltage is measured and adjusted automatically, but can also be set manually.

Access: "Menu" > "Vertical" > "Probes" > "Termination"




**Enable**
Activates control of the termination voltage.

Remote command:
PROBe<m>:SETup:TERM:STATe on page 1269

**Mode**
In "Auto" mode, the instrument uses the measured common mode voltage for termination.

In "Manual" mode, you can enter the voltage to be used for termination. Use the manual mode if you know the common mode voltage of the DUT.

Remote command:
PROBe<m>:SETup:TERM:MODE on page 1270

**Adjustment**
Sets the voltage to be used for termination to DC voltage.

Remote command:
PROBe<m>:SETup:TERM:ADJust on page 1270

**Measurement**
Shows the measured common mode voltage.

Remote command:
`PROBe<m>:SETup:TERM:MEASure?` on page 1270

### 5.6.6 Setup for predefined probes

Probes that cannot be detected, but their characteristics are known to the R&S RTP are called "Predefined probes".

The following shared probe settings are available:

- "Type, Name, Bandwidth" on page 147
- "Detect AutoZero, Use AutoZero" on page 147

The probe attenuation of the selected probe is also shown in the "Probes" tab:

- "Probe unit, Auto attenuation" on page 149

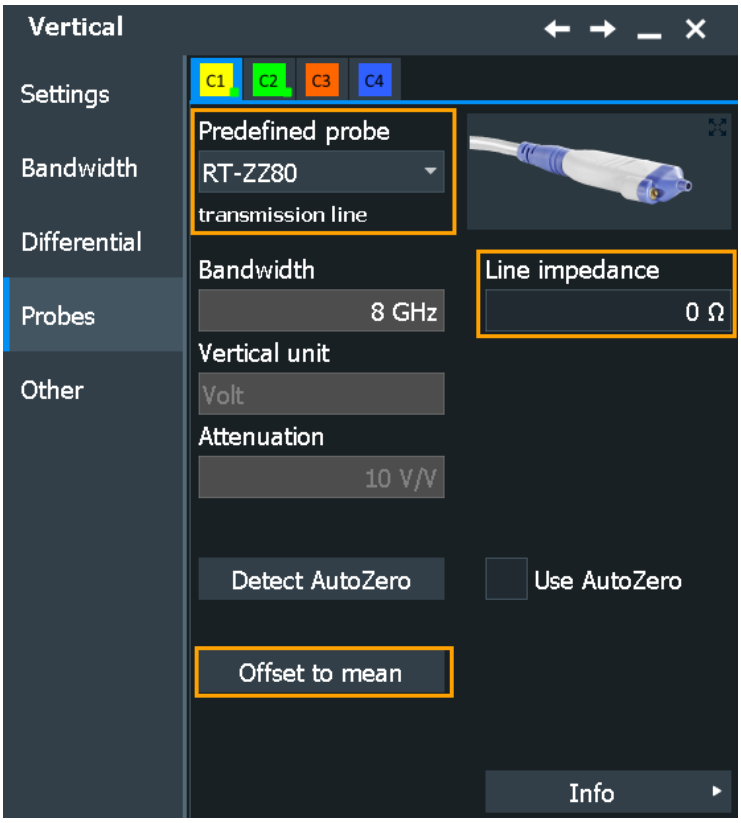



*Figure 5-13: Probe setup for transmission line probe R&S RT-ZZ80*

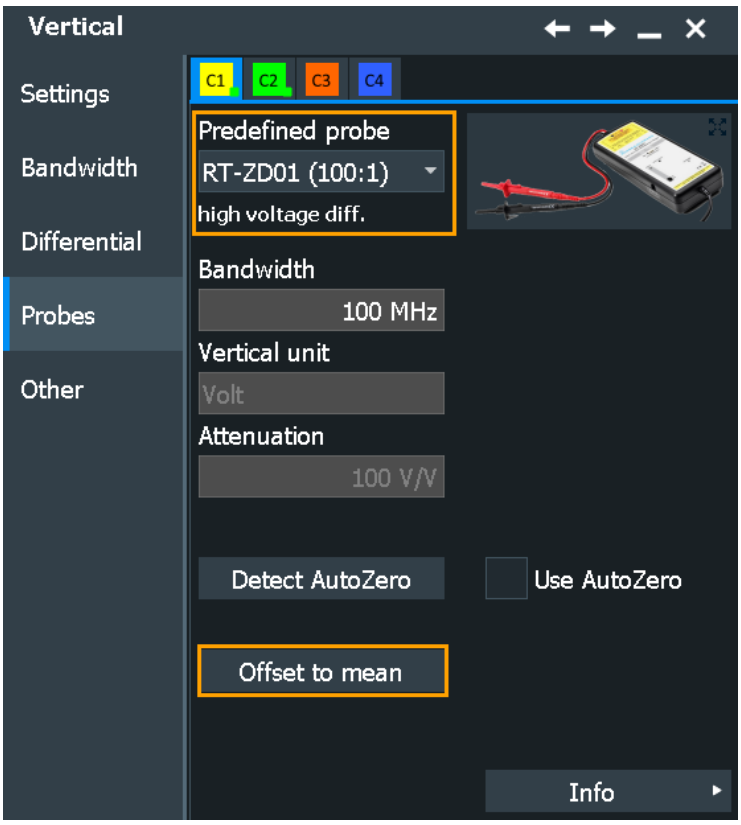



*Figure 5-14: Probe setup for R&S RT-ZD01*

Specific settings for predefined probes are the following:

Predefined probe....164
Offset to mean....165
Line impedance....165

**Predefined probe**
List of probes that are known to the instrument. The instrument lists only probes that match the instrument coupling and, if connected, also the used adapter.

Select the used probe on the list. The corresponding "Vertical unit" and the "Attenuation" or "Gain" are shown.

All other unrecognized probes that are not listed, are unknown probes. For these probes, set "Predefined probe" to "None". See Chapter 5.6.8, "Setup for unknown probes", on page 168.

Remote command:
`PROBe<m>:SETup:ATTenuation:DEFProbe` on page 1274

**Offset to mean**
Performs an automatic compensation for a DC component of the input signal using the result of a background mean measurement. The result is shown in "Offset". The function is probe-independent and supports quick and convenient measurements of input signals with different DC offsets. It detects offset values even when the signal is out of the current measurement range. It also sets the zero level to the determined DC offset in the middle of the screen and thus prevents clipping of the waveform.

Remote command:
`PROBe<m>:SETup:OFFSet:TOMean` on page 1275

**Line impedance**
If the transmission line probe R&S RT-ZZ80 is selected, enter the impedance of the measured line.

The actual attenuation of the transmission line probe depends on the impedance of the line $Z_0$:

*Attenuation = 10 + $Z_0$ / 100*

The instrument uses the actual attenuation to determine the measurement values.

## 5.6.7 Setup for current probes

The setup and adjustment of current probes depends on the output connector of the probe: BNC or Rohde & Schwarz probe box.

The following shared probe settings are available:

- "Type, Name, Bandwidth" on page 147
- "Detect AutoZero, Use AutoZero" on page 147

**Current probes R&S RT-ZCxx**

The current probes **R&S RT-ZCxx** have BNC connectors. They are known to the R&S RTP as predefined probes, see Chapter 5.6.6, "Setup for predefined probes", on page 163. Demagnetizing and zero adjustment is done on the probe, see the probe's User Manual for details. Make sure to demagnetize and adjust the probe before taking measurements.

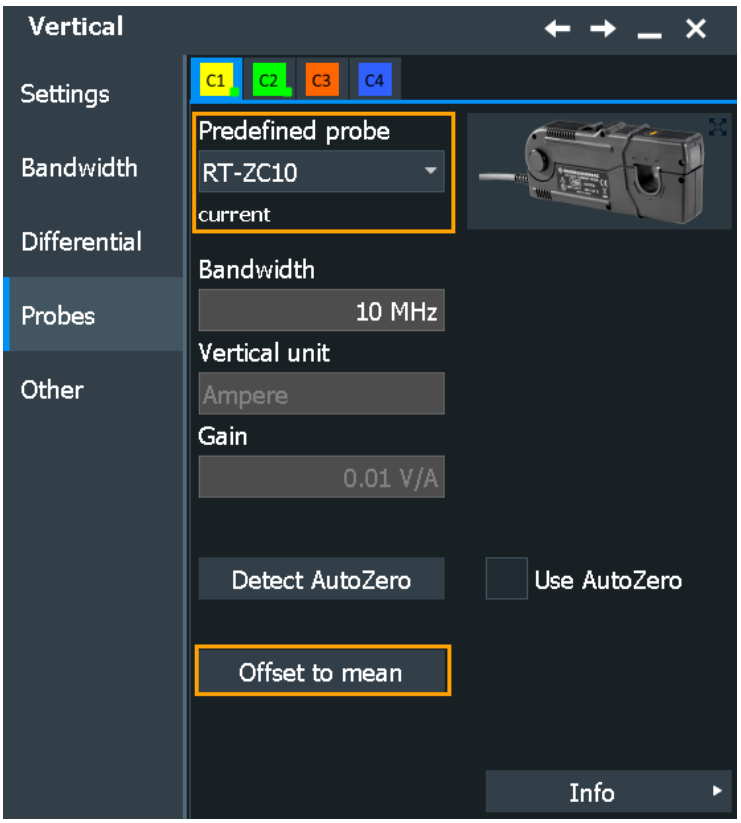



*Figure 5-15: Probe setup for current probes R&S RT-ZC10*

### Current probes R&S RT-ZCxxB

Current probes **R&S RT-ZCxxB** have a Rohde & Schwarz probe interface; they are powered and remotely controlled by the oscilloscope.

When the probe is connected, demagnetization is performed automatically.

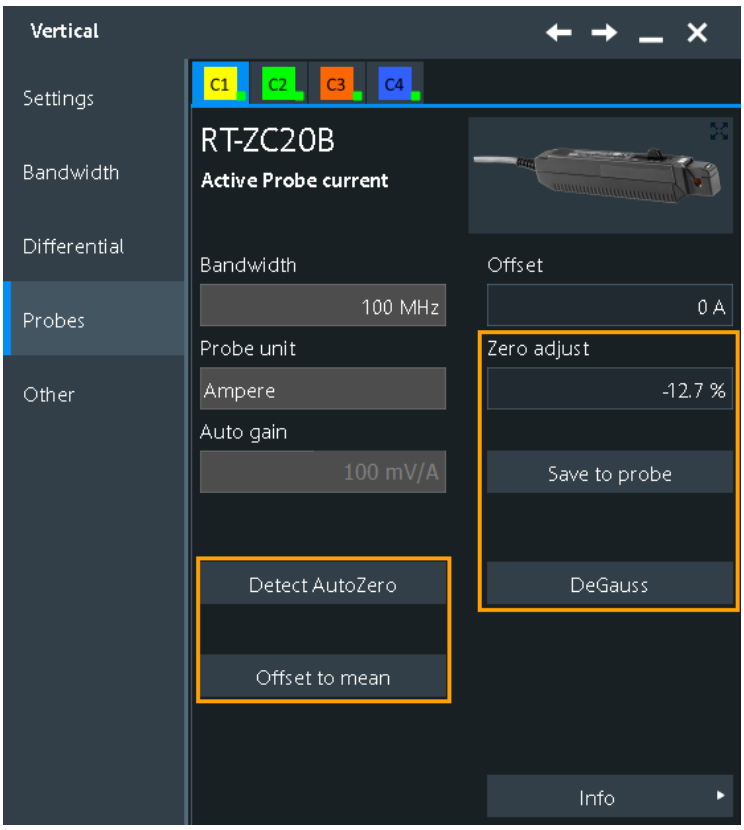



*Figure 5-16: Probe setup for current probes R&S RT-ZC20B*

For all current probes, attenuation or gain is shown in the "Setup" tab, and you can set the offset to mean:

- "Probe unit, Auto attenuation" on page 149
- "Offset to mean" on page 165

Current probes R&S RT-ZCxxB are adjusted by the following functions:

DeGauss ..... 167
Detect AutoZero ..... 168
Zero adjust ..... 168
Save to probe ..... 168

**DeGauss**
Demagnetizes the core if it has been magnetized by switching the power on and off, or by an excessive input. Always carry out demagnetizing before measurement.

The demagnetizing process takes about one second. During demagnetizing, a demagnetizing waveform is displayed.

Demagnetizing is done automatically when R&S RT-ZCxxB is connected to the oscilloscope, or when "Detect AutoZero" is performed.

Remote command:
`PROBe<m>:SETup:DEGauss` on page 1275

**Detect AutoZero**
If a current probe is connected, the function demagnetizes the probe's sensor head and sets the waveform to zero position to correct the error offset. Thus, it compensates for the remanence and offset caused by temperature drift.

For R&S RT-ZCxxB probes, the determined "Zero adjust" value is displayed and can be saved in the probe head.

See also "Detect AutoZero, Use AutoZero" on page 147.

Remote command:
`PROBe<m>:SETup:OFFSet:AZERo` on page 1257

**Zero adjust**
Zero adjust corrects the effect of an offset caused by temperature drift, and compensates for the remanence. The setting is only available if DC coupling is set.

To set the waveform to zero level by the instrument, use "Detect AutoZero". The detected value is displayed.

Alternatively, you can adjust the value manually until the waveform is set to zero level. Make sure to demagnetize the probe before zero adjustment.

The value is given in percent of the maximum range, which is internally defined. The actual setup range depends on the temperature drift, the measured current and other variables, and it can change over time. If you measure high currents, the probe core magnetizes, which impairs the measurement results. Therefore, repeat "Detect AutoZero" before the measurement.

Remote command:
`PROBe<m>:SETup:OFFSet:ZADJust` on page 1276

**Save to probe**
Saves the "Zero adjust" value in the probe box. If you connect the probe to another channel or to another R&S RTx oscilloscope, the value is read out again, and you can use the probe without further adjustment.

Remote command:
`PROBe<m>:SETup:OFFSet:STPRobe` on page 1276

### 5.6.8 Setup for unknown probes

If the R&S RTP cannot detect the probe, and the probe is not a predefined one, you can set the probe parameters manually.

The following shared probe settings are available:

- "Type, Name, Bandwidth" on page 147
- "Detect AutoZero, Use AutoZero" on page 147

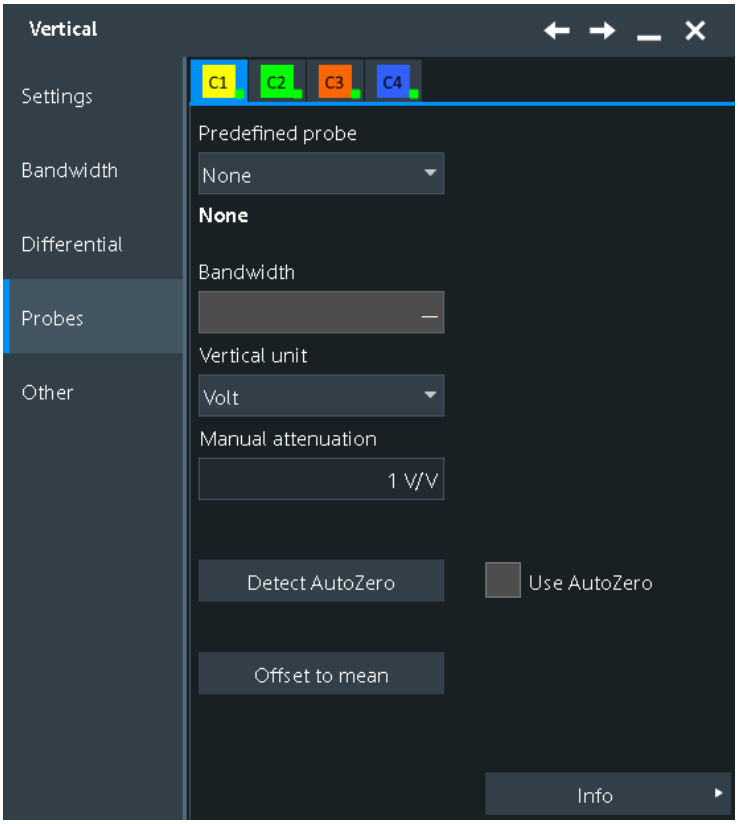


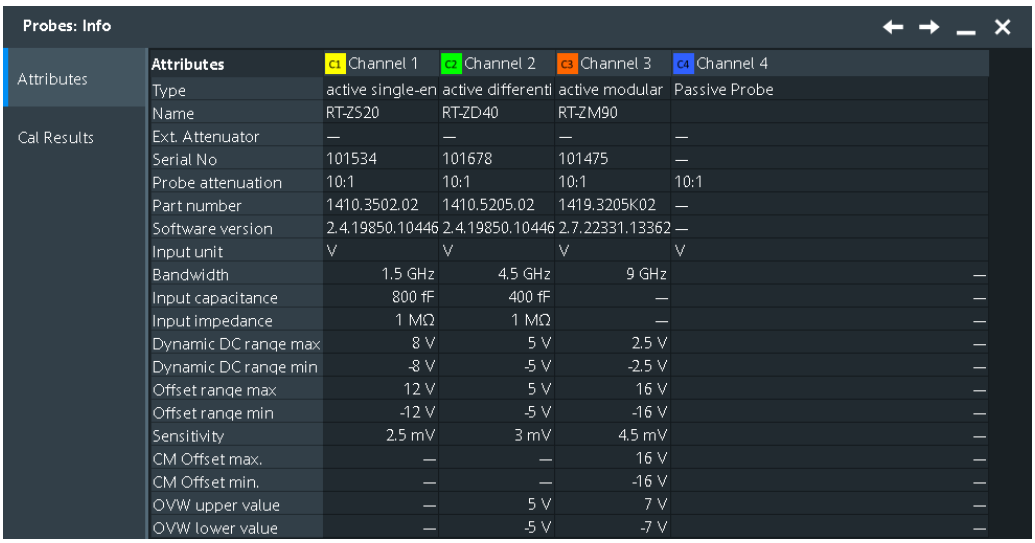

*Figure 5-17: Probe setup for an unknown probe*

Set the unit and the attenuation or gain of the probe: "Vertical unit, Attenuation, Gain" on page 153.

### 5.6.9 Probe attributes

To check the attributes of the connected probes, select "Info" in the "Vertical" > "Probes" tab.

For a specification of the probe parameters, refer to the data sheet.




Remote commands:

- `PROBe<m>:ID:SWVersion?` on page 1277
- `PROBe<m>:ID:PRDate?` on page 1277
- `PROBe<m>:ID:PARTnumber?` on page 1277
- `PROBe<m>:ID:SRNumber?` on page 1278
- `PROBe<m>:SETup:CAPacitance?` on page 1278
- `PROBe<m>:SETup:IMPedance?` on page 1278

## 5.6.10 Calibration results

To check the attributes of the connected probes, select "Info" > "Cal Results" in the "Probes" tab.

## 5.6.11 Probe adapter R&S RT-Z1M

Using the R&S RT-Z1M probe adapter, you can connect probes requiring 1 MΩ load impedance to the R&S RTP, which has inputs with 50 Ω impedance. The Rohde & Schwarz probe interface at the adapter input allows you to connect various probes, e.g. passive probes, current probes, and active probes with Rohde & Schwarz probe interface.

1. Connect the R&S RT-Z1M adapter to the channel input.

   The instrument identifies the adapter.

2. Connect the probe to the adapter.

   The oscilloscope reads the probe-specific parameters through the adapter.

3. Set up the probe as usual, depending on the type of the connected probe.

4. In the "Adapter Setup", start the self-alignment of the adapter.

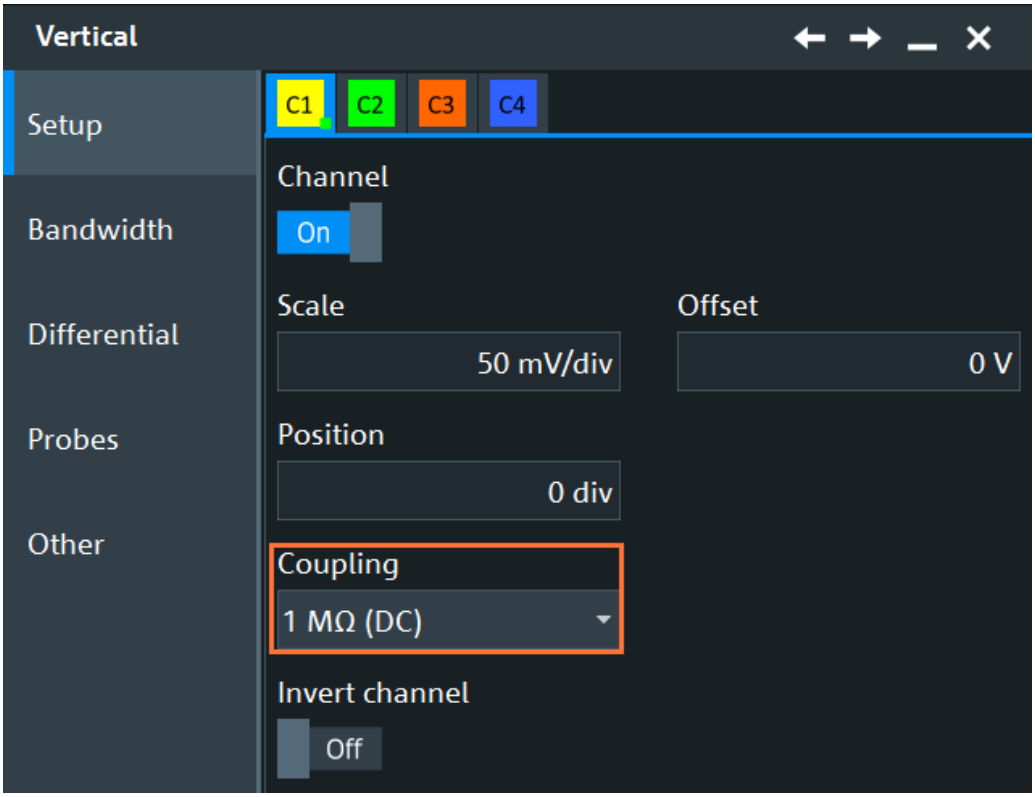
If the adapter is connected, additional settings are available in several dialog boxes:

- Channel settings with connected adapter ..... 171
- Probe settings with connected adapter ..... 172
- R&S RT-Z1M setup ..... 173

### 5.6.11.1 Channel settings with connected adapter

Access: [C<n>] > "Channels" tab

In the "Channels" tab, the adapter is shown, and you can set the coupling and the bandwidth limit here.




**Coupling**
Selects the coupling, which determines what part of the signal is used for waveform analysis and triggering.

In addition to coupling, the signal can be filtered for high frequency rejection, see Chapter 5.5.5, "Digital filter setup", on page 145.

"1 MΩ (DC)" Connection with 1 MΩ termination, passes both DC and AC components of the signal.

"1 MΩ (AC)" Connection with 1 MΩ termination through DC capacitor, removes DC and very low-frequency components.

If AC coupling is set, the attenuation of passive probes has no effect, and voltage is applied to the instrument with factor 1:1. Observe the voltage limits, otherwise you can damage the instrument.

Remote command:
`CHANnel<m>:COUPling` on page 1248

#### 5.6.11.2 Probe settings with connected adapter

Access: "Menu" > "Vertical" > "Probes"

In the "Probes" tab, you can switch on the R&S ProbeMeter of the adapter, and navigate to the adapter settings.

The R&S ProbeMeter of the adapter takes effect if the connected probe does not have an own R&S ProbeMeter. Probes with an own R&S ProbeMeter use their own one.

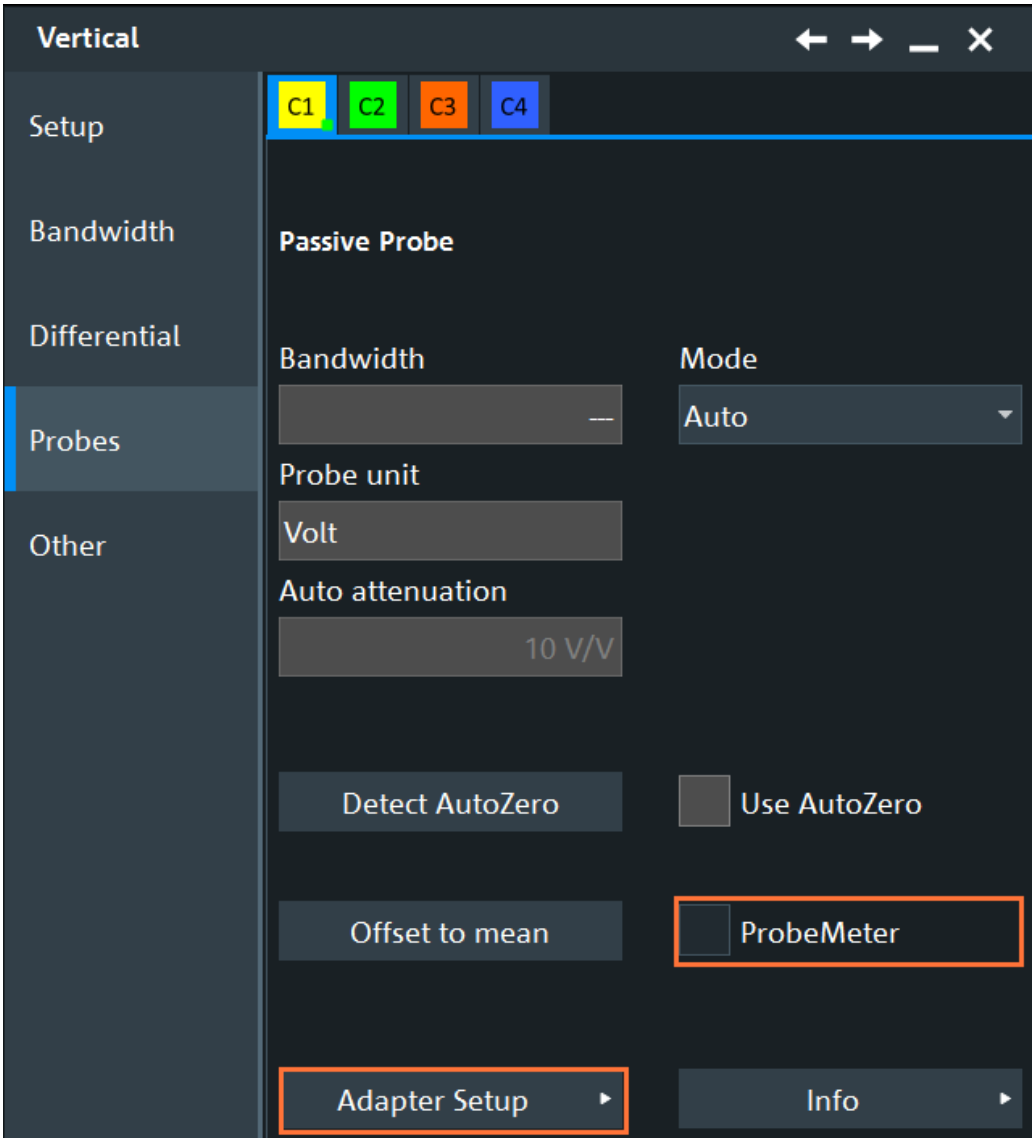
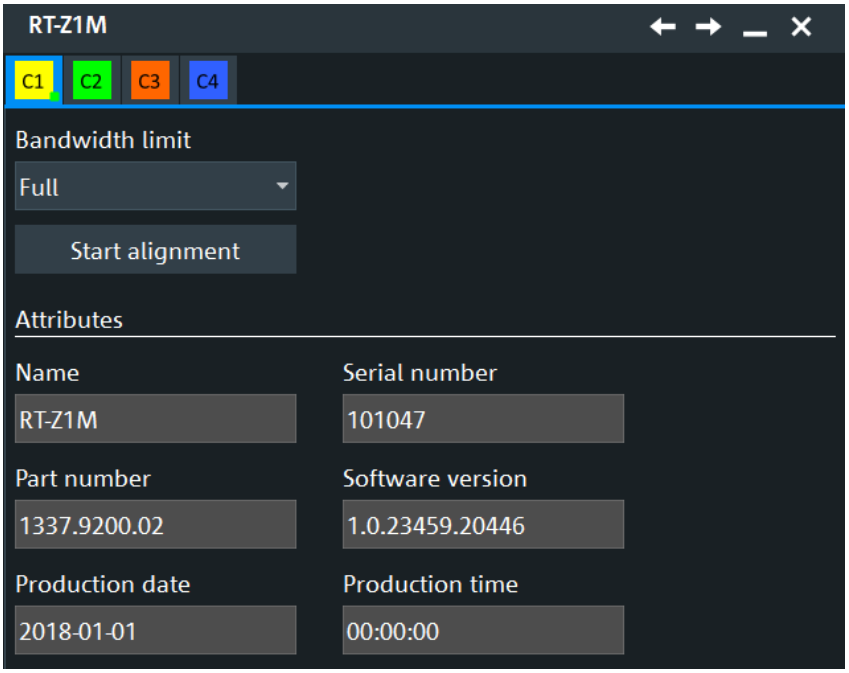



*Figure 5-18: Probe settings of passive probe connected via R&S RT-Z1M adapter, R&S ProbeMeter of the adapter is available*

### 5.6.11.3 R&S RT-Z1M setup

Access: "Menu" > "Vertical" > "Probes" > "Adapter Setup"

In the "RT-Z1M" dialog, you can change the bandwidth limit, and start the self-alignment. You also can see information on the adapter: name, serial number, part number, software version, production date and time.




**Bandwidth limit**
Selects the bandwidth limit of the adapter. The specified bandwidth indicates the range of frequencies that the instrument can acquire and display accurately with less than 3dB attenuation.

| | |
|---|---|
| "Full" | At full bandwidth, all frequencies in the specified range are acquired and displayed. Full bandwidth is used for most applications. |
| 200 MHz, 20 MHz | Frequencies above the selected limit are removed to reduce noise at different levels. |

Remote command:
`PROBe<m>:CONVerter:BANDwidth` on page 1279

**Selfalignment**
Self-alignment compensates measurement errors caused by a difference of termination impedance, or an ambient temperature different from the one at which the factory alignment was done. When the adapter is detached from the oscilloscope, the values of the self-alignment are discarded, and the original factory-alignment is reloaded. Thus, repeat the selfalignment when you reattach the probe.

Remote command:
`PROBe<m>:CONVerter:SELFalign` on page 1279

## 5.7 R&S RT-ZVC probe

With the R&S RTP and option R&S RTP-B1E, you can use the R&S RT-ZVC multichannel power probe. It has an integrated 2- or 4-channel amperemeter and 2- or 4-channel voltmeter. The probe provides parallel measurements of analog or digital, voltage/current signals with excellent 18-bit resolution.

For more information on the R&S RT-ZVC probe, see also its user manual.

**Source Channels**

You can simultaneously connect a R&S RT-ZVC and a R&S RT-ZL04 to the R&S RTP, but no parallel operation on screen is possible.

You can acquire and measure the R&S RT-ZVC or R&S RT-ZL04 together with the analog input channels. They are running on the same horizontal scale.

If an amperemeter or voltmeter channel is activated, it can be displayed on the screen and used as a source for:

- Cursor measurements
- Automatic measurements, and also histogram, limit checks, longterm, track, quick measurements
- Mask testing
- Basic and advanced maths
- Reference waveform
- Trigger type "Edge"
- Search: all search events
- XY diagrams
- Data export

All features of the base unit, the R&S RTP-K37 option and R&S RTP-K19 option are supported.

**Data export**

You can save the data of the amperemeter and voltmeter channels to an XML, CSV, or BIN file. One channel per file can be saved. Files in BIN format can be reloaded to the R&S RTP as reference waveforms.

See also:

- Chapter 12.2.7, "Saving and loading waveform data", on page 505
- Chapter 12.2.1, "Waveform export files", on page 483

Remote commands for export to file:

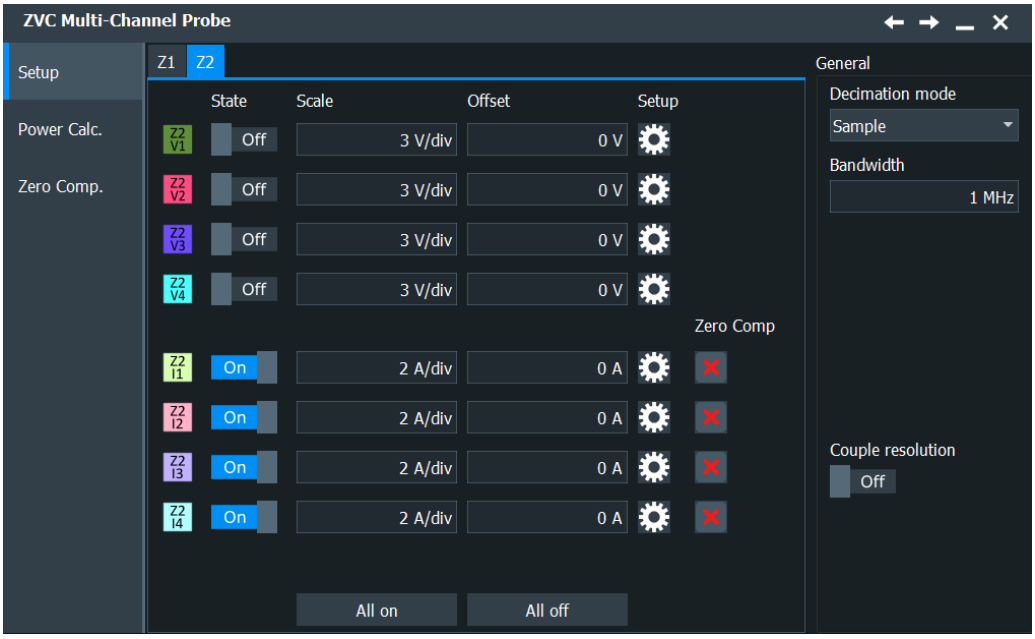
- `EXPort:WAVeform:SOURce` on page 1614
- `EXPort:WAVeform:NAME` on page 1616
- `EXPort:WAVeform:SAVE` on page 1617

## 5.7.1 R&S RT-ZVC overview

### 5.7.1.1 Setup

Access: [Apps]> "Analysis" > "Multi-channel probe" > "Setup" tab

Make sure that the tab of the correct probe is selected on the left side.




**State**
Enables the corresponding channel of the probe. The number of available channels depend on the characteristics of your multi-channel power probe.

Remote command:
`ZVC:Z<m>:V<n>[:STATe]` on page 1290
`ZVC:Z<m>:I<n>[:STATe]` on page 1287

**Scale**
Defines the vertical scale for the channel in Volts per division. Increasing the scale compresses the display of the signal. Within a given operation range, modifying the scale is equivalent to scaling a display range.

For the voltmeter channels, the Scale, the Offset and the "Position" specify the operating range of the voltmeter.

Remote command:
`ZVC:Z<m>:V<n>:SCALe` on page 1289
`ZVC:Z<m>:I<n>:SCALe` on page 1284

**Offset**
The vertical center of the selected channel is shifted by the offset value and the signal is repositioned within the diagram area. Negative offset values move up the waveform, positive values move it down.

Within a given operation range, modifying the offset is equivalent to moving vertically the display range. The offset can only be modified such that the display range reaches at most the limits of the operation range.

For the voltmeter channels, the Scale, the Offset and the "Position" specify the operating range of the voltmeter.

Remote command:
ZVC:Z<m>:V<n>:OFFSet on page 1288
ZVC:Z<m>:I<n>:OFFSet on page 1282

**Setup**
Opens the voltage or current settings for the selected channel. See Chapter 5.7.2, "ZVC voltage settings", on page 179 and Chapter 5.7.3, "ZVC current settings", on page 181.

**Zero Comp**
Shows the status of the zero compensation of each current channel. The green checkmark indicates that the zero offset is compensated automatically. To adjust the settings, tap the icon. For details, see Chapter 5.7.1.3, "Zero compensation", on page 179.

Remote command:
ZVC:Z<m>:I<n>:ZERComp:STATe? on page 1292

**All on**
Enables all available channels.

**All off**
Disables all available channels.

**Decimation mode**
Selects the decimation mode for all R&S RT-ZVC probes. Decimation reduces the data stream of the ADC to a stream of waveform points with lower sample rate and a less precise time resolution.

| | |
|---|---|
| "Sample" | One of n samples in a sample interval of the ADC is recorded as waveform point, the other samples are discarded. The time between the two adjacent waveform points is exactly the resolution. Very short glitches might remain undiscovered by this method. |
| "Peak detect" | The minimum and the maximum of n samples in a sample interval are recorded as waveform points, the other samples are discarded. |
| "High res" | The average of n sample points is recorded as one waveform sample. Averaging reduces the noise, the result is a more precise waveform with higher vertical resolution. The high measurement resolution is suitable for high accuracy measurements of instantaneous values. |

Remote command:
ZVC:TYPE on page 1281

**Bandwidth**
Sets the bandwidth limit of all R&S RT-ZVC probes. The bandwidth specifies the maximum frequency at which a purely sinusoidal signal is still transferred at 89 % (1 dB) of its amplitude.

The bandwidth of some current channels is restricted to 300 KHz due to their vertical settings.

Remote command:
ZVC:BANDwidth on page 1281

**Couple resolution**
Sets the resolution of all R&S RT-ZVC channels.

"On" The resolution of the analog channels is applied to R&S RT-ZVC channels. The signal is automatically interpolated or decimated to get the analog resolution.

"Off" The resolution of R&S RT-ZVC channels is set in a way so that the record length of the waveforms is minimum 1000 samples.

Remote command:
ZVC:RESCoupled on page 1283

### 5.7.1.2 Power calculation

Access: [Apps]> "Analysis" > "Multi-channel probe" > "Power Cal." tab

**State**
Enables the corresponding channel of the probe. The number of available channels depend on the characteristics of your multi-channel power probe.

Remote command:
ZVC:Z<m>:V<n>[:STATe] on page 1290
ZVC:Z<m>:I<n>[:STATe] on page 1287

**Measurement Impedance**
Sets the impedance of the probe channel for power calculations and measurements.

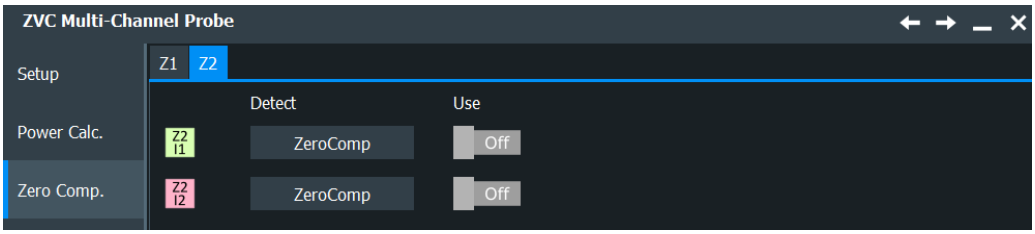
Remote command:
ZVC:Z<m>:V<n>:IMPedance on page 1288
ZVC:Z<m>:I<n>:IMPedance on page 1282

#### 5.7.1.3 Zero compensation

Access: [Apps]> "Analysis" > "Multi-channel probe"> "ZeroComp" tab

Zero compensation is used to avoid negative currents in measurement results, and to improve the measurement accuracy. The determined compensation is valid as long as the temperature is constant and the probe settings are unchanged.

1. Disconnect the DUT from the power supply.
2. Adjust the current settings, in particular, the shunt settings. See Chapter 5.7.3, "ZVC current settings", on page 181.
3. Connect one probe tip of the R&S RT-ZVC probe to the high voltage pin of the DUT. Disconnect the other probe tip from the DUT.
4. Open the "ZeroComp" settings: [Apps]> "Analysis" > "Multi-channel probe"> "ZeroComp" tab
5. Tap "ZeroComp" to determine the zero offset.
6. Enable "Use" to compensate for the zero offset.




**ZeroComp**
Measures the zero offset, the mean value on a currentless DUT. If temperature changes, or if you change the shunt or other probe settings, repeat the measurement.

Remote command:
`ZVC:Z<m>:I<n>:ZERComp:DETect` on page 1291

**Use**
If enabled, the measured "ZeroComp" offset value is compensated automatically.

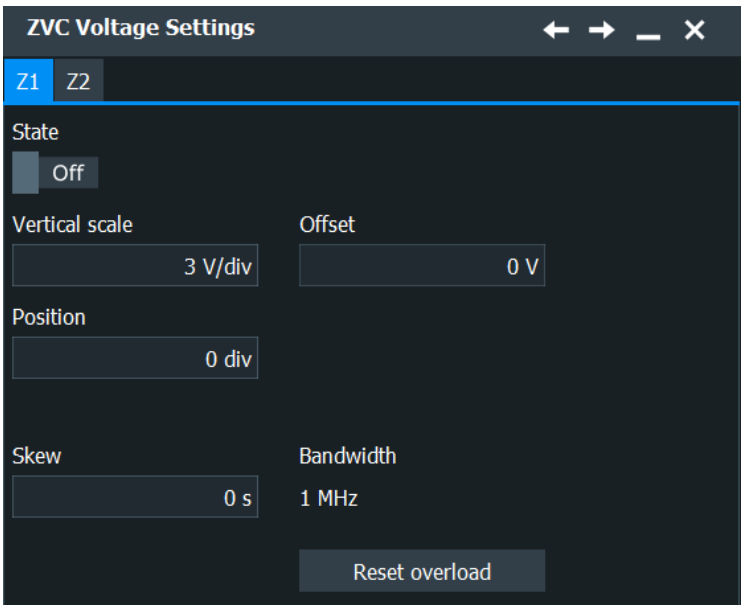
Remote command:
`ZVC:Z<m>:I<n>:ZERComp:USE` on page 1291

### 5.7.2 ZVC voltage settings

Access: [Apps]> "Analysis" > "Multi-channel probe"> "Setup" tab > "Setup" icon of voltage channel

See also:

- Scale
- Offset




**Channel**
Selects the voltmeter channel to be configured.

**State**
Enables the corresponding voltage channel of the probe. The number of available channels depend on the characteristics of your multi-channel power probe.

Remote command:
`ZVC:Z<m>:V<n>[:STATe]` on page 1290

**Position**
Moves the selected signal up or down in the diagram. The visual effect is the same as for Offset. While the offset sets a voltage, position is a graphical setting given in divisions. Within a given operation range, modifying the position is equivalent to moving vertically the display range. The position can only be modified such that the display range reaches at most the limits of the operation range.

For the voltmeter channels, the Scale, the Offset and the "Position" specify the operating range of the voltmeter.

Remote command:
`ZVC:Z<m>:V<n>:POSition` on page 1289

**Bandwidth**
Displays the bandwidth of the current channel. You can set the probe bandwidth in the "Setup" dialog.

Remote command:
`ZVC:Z<m>:V<n>:BANDwidth?` on page 1287

**Skew**
Sets the skew, a delay value, that is known from the circuit specifics but cannot be compensated by the instrument automatically.

Remote command:
ZVC:Z<m>:V<n>:SKEW on page 1290

**Reset Overload**
Resets the overload indication at the probe.

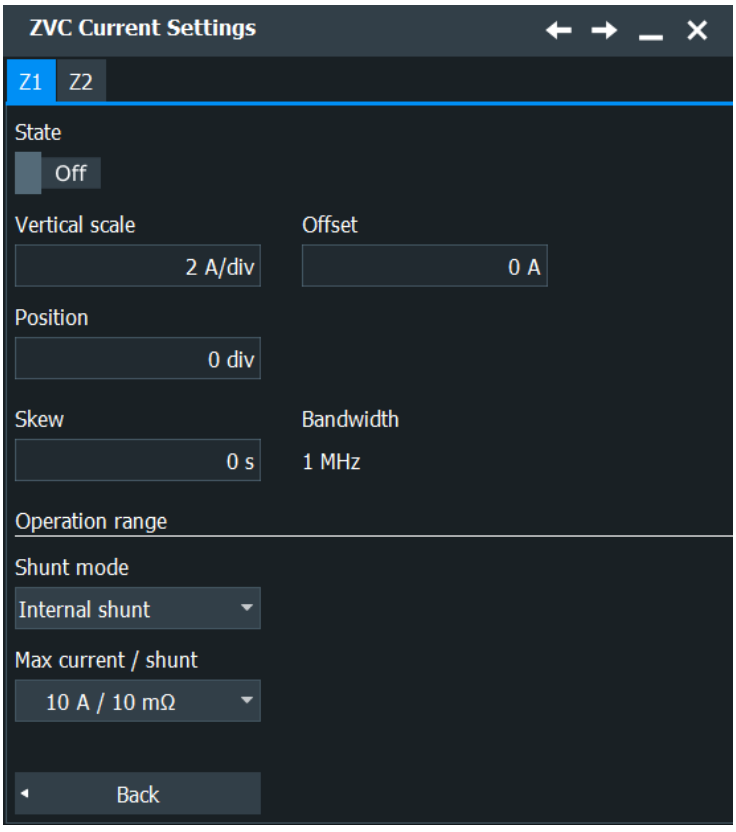
Remote command:
ZVC:Z<m>:V<n>:OVERload:RSTO on page 1288

### 5.7.3 ZVC current settings

Access: [Apps]> "Analysis" > "Multi-channel probe"> "Setup" tab > "Setup" icon of current channel

See also:

- Scale
- Offset




**Channel**
Selects the amperemeter channel to be configured.

**State**
Enables the corresponding current channel of the probe. The number of available channels depend on the characteristics of your multi-channel power probe.

Remote command:
ZVC:Z<m>:I<n>[:STATe] on page 1287

**Bandwidth**
Displays the bandwidth of the current channel. You can set the probe bandwidth in the "Setup" dialog.

Remote command:
ZVC:Z<m>:I<n>:BANDwidth? on page 1281

**Position**
Moves the selected signal up or down in the diagram. The visual effect is the same as for Offset. While the offset sets a current, position is a graphical setting given in divisions. Within a given operation range, modifying the position is equivalent to moving vertically the display range. The position can only be modified such that the display range reaches at most the limits of the operation range.

Remote command:
ZVC:Z<m>:I<n>:POSition on page 1284

**Skew**
Sets the skew, a delay value, that is known from the circuit specifics but cannot be compensated by the instrument automatically.

Remote command:
ZVC:Z<m>:I<n>:SKEW on page 1286

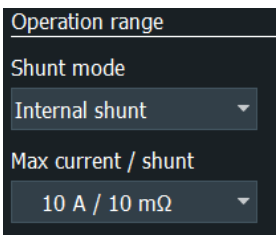
**Shunt mode**
Selects the internal or external shunt mode.

Regarding the shunt selection, i.e. the burden voltage level, there is a trade-off between the burden of the circuit under test and the SNR at the frontend input. The burden voltage is the DUT circuit loading caused by leads, connectors and the amperemeter circuit.

From the DUT perspective, the burden voltage has to be kept low not to distort the device operation. In contrast, from the probe's view the voltage has to be as large as possible to obtain a good SNR. For that reason, the external shunt can be applied to get the best compromise of both for a specific measurement range.

Remote command:
ZVC:Z<m>:I<n>:SHUNt:MODE on page 1285

**Internal Shunt Mode**
If Shunt mode is set to "Internal shunt", include the settings for the internal shunt mode.



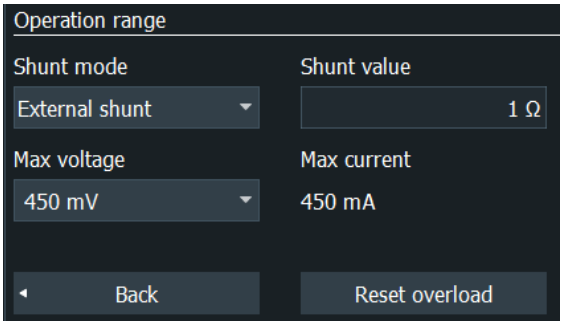
**Max current / shunt ← Internal Shunt Mode**
Selects the maximum current and the internal shunt value.

With the maximum current and the internal shunt selection, the operating range of the amperemeter is specified. At the same time, the burden voltage at the amperemeter input can be estimated. For values of the total round-trip resistance that can be seen at the test lead ends, consider the data sheet.

For using internal shunts, the circuit under test needs to be interrupted so that the current can flow through the probe.

Remote command:
`ZVC:Z<m>:I<n>:SHUNt:MAXCurrent` on page 1285

**External shunt mode**
If Shunt mode is set to "External shunt", include the settings for the external shunt mode.




**Max voltage ← External shunt mode**
Selects the maximum voltage at the external shunt.

Remote command:
`ZVC:Z<m>:I<n>:SHUNt:MAXVoltage` on page 1286

**Shunt value ← External shunt mode**
Sets the shunt value of the external shunt resistor.

Remote command:
`ZVC:Z<m>:I<n>:SHUNt:EVALue` on page 1285

**Max current ← External shunt mode**
Displays the maximum current.

Remote command:
`ZVC:Z<m>:I<n>:SHUNt:MXCValue?` on page 1286

**Reset Overload ← External shunt mode**
Resets the overload indication at the probe.

Remote command:
`ZVC:Z<m>:I<n>:OVERload:RSTO` on page 1283

## 5.8 Differential signals

With R&S RTP, you can easily analyze differential signals using single-ended probes, or even cables. The instrument processes the input of single-ended probes on analog channels and creates the differential and common mode waveforms. Similar to Rohde & Schwarz modular probes, you can display the waveforms of differential voltage, common mode voltage, positive single-ended voltage and negative single-ended voltage. Triggering on differential signals is also possible.

Deembedding is possible if option R&S RTP-K121 is installed. With option R&S RTP-K122, real-time deembedding is also available.

Deembedding is applied to the two single-ended captured signals, resulting in deembedded differential waveform, common mode waveform, positive single-ended waveform and negative single-ended waveform. Deembedding also considers cross-couplings.

### 5.8.1 Settings of differential signals

Access: "Menu" > "Vertical" > "Differential"

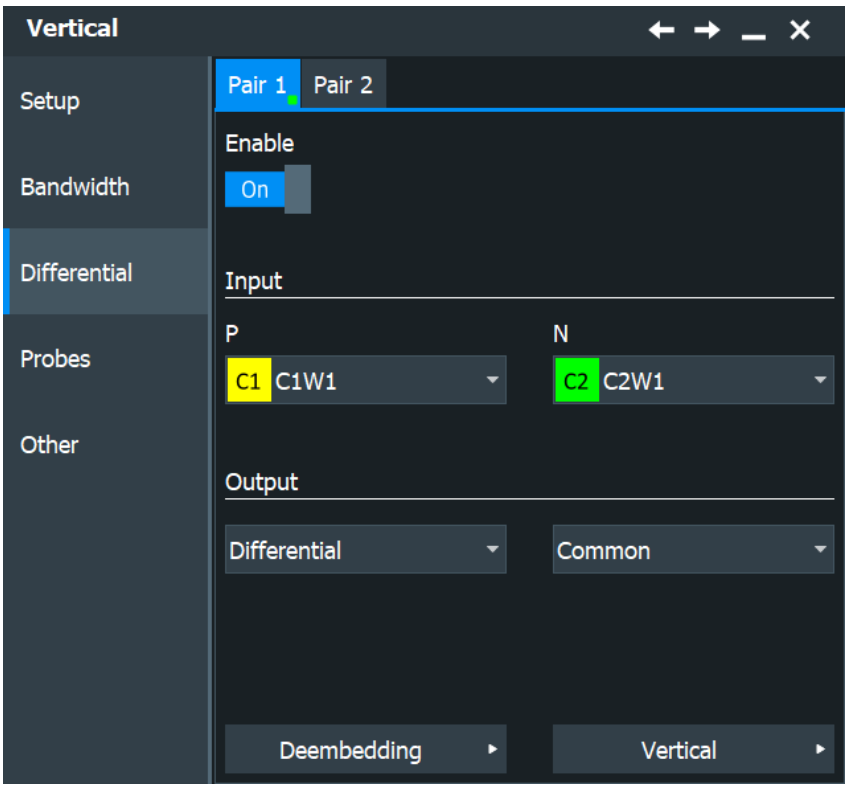



*Figure 5-19: Setup of differential signals without deembedding*

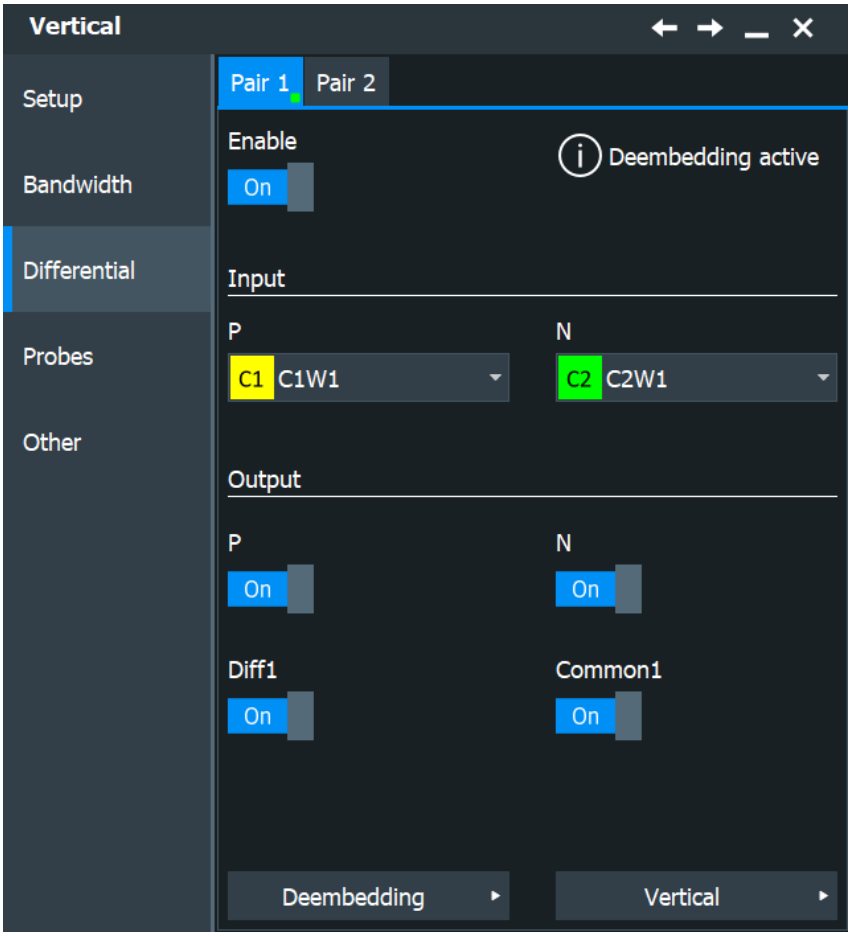



*Figure 5-20: Setup of deembedded differential signals*

**Enable**
Activates the differential signal.

Remote command:
`DIFFerential<m>:STATe` on page 1295

**P, N**
Select the analog input channel of the positive and negative signal.

Remote command:
`DIFFerential<m>:PSIGnal[:SELect]` on page 1295
`DIFFerential<m>:NSIGnal[:SELect]` on page 1295

**Output: P, N, Diff, Common**
Select the waveforms that are displayed as result of differential processing.

The number of output waveforms per differential signal depends on the deembedding:
- No deembedding: 2 output waveforms.
- Deembedding by software: up to 4 output waveforms. All output waveforms can be displayed at the same time.

- Real-time deembedding: 1 output waveform. It can either be the deembedded differential, common-mode, positive single-ended or negative single-ended waveform.

In the diagram, you see the output waveforms, and the signal icons of the input channels are grayed out. The input channels are only visible if "P" or "N" are selected as output.

"P", "N" Displays the waveform of the positive or negative input signal. Without deembedding, the input channel is shown directly, no additional waveform is created. If differential deembedding is active, you always see the deembedded waveforms, but never the captured input signal.

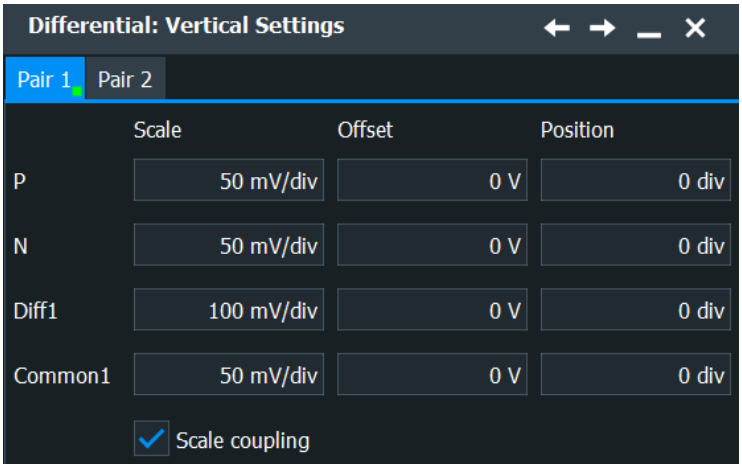
"Differential" Displays the differential signal as a new waveform.

"Common" Displays the common mode signal as a new waveform.

Remote command:
`DIFFerential<m>:AOUTput` on page 1295
`DIFFerential<m>:BOUTput` on page 1295
If deembedding by software is active, use the following commands:
`DIFFerential<m>:COMMon:STATe` on page 1296
`DIFFerential<m>:DIFFerential:STATe` on page 1296
`DIFFerential<m>:NSIGnal:STATe` on page 1296
`DIFFerential<m>:PSIGnal:STATe` on page 1296

## 5.8.2 Vertical setup of differential signals

Access: "Menu" > "Vertical" > "Differential" > "Vertical Setup"




**Scale**
Sets the vertical scale of differential and common mode waveforms.

Vertical settings of P and N output are the vertical settings of the input channels.

Remote command:
`DIFFerential<m>:COMMon:SCALe` on page 1296
`DIFFerential<m>:DIFFerential:SCALe` on page 1296

**Scale coupling**
If enabled, the vertical scales of P, N, differential and common mode waveforms are coupled. The scales are related as follows:

$VertScale_P = VertScale_N = VertScale_{CM} = VertScale_{Diff}/2$

You can disable the scale coupling and set an individual scale for each waveform.

Remote command:
`DIFFerential<m>:COUPling` on page 1297

**Offset**
Sets the offset of differential and common mode waveforms.

Vertical settings of P and N output are the vertical settings of the input channels.

Remote command:
`DIFFerential<m>:COMMon:OFFSet` on page 1297
`DIFFerential<m>:DIFFerential:OFFSet` on page 1297

**Position**
Sets the vertical position of differential and common mode waveforms.

Vertical settings of P and N output are the vertical settings of the input channels.

Remote command:
`DIFFerential<m>:COMMon:POSition` on page 1297
`DIFFerential<m>:DIFFerential:POSition` on page 1297

### 5.8.3 Deembedding of differential signals

To remove the parasitic effects of the measurement setup from the measured signal, you can use deembedding for differential signals. Deembedding requires option R&S RTP-K121, and real-time deembedding requires option R&S RTP-K122 in addition.

The deembedding of differential signals is represented by one or more 4-port input components. You can activate deskewing, and set a skew time.

See also: Chapter 20, "Deembedding (option R&S RTP-K121/-K122)", on page 1087.

1. On the "Vertical" menu, select "Differential Signals".
2. Select the input channels of the positive and negative signal.
3. Enable the differential signal.
4. Select "Deembedding".
5. Select the channel subtab of the positive input.
6. On the "Setup" tab, enable deembedding for the channel.
7. Optionally, you can enable real-time deembedding if available.
8. On the "Input" component, select "Configure".

9. If necessary, activate "Use skew offset". Use the deskewing if the P and N waveforms are not correctly aligned.

When you disable a deembedded differential signal in "Vertical" > "Differential Signals", deembedding is also disabled. When you enable the differential signal again, repeat the deembedding and output settings.

### 5.8.4 Analysis of differential signals

Before analyzing the output waveforms of differential signals, make sure that the P and N signals are correctly aligned. Without deembedding, use "Horizontal" > "Skew" to deskew the input waveforms if necessary. For deembedded waveforms, the skew setting is in the deembedding configuration.

For all output waveforms of differential signals (differential, common mode, P, N), the usual analysis methods are available:

- Trigger on differential, common mode, P, and N waveforms.
  Without deembedding, and with software deembedding, all possible output waveforms are available as trigger source, even if they are not selected and not displayed. Therefore, the differential signal must be enabled ("Enable" = on).
  With real-time deembedding, you can trigger only on the selected output waveform. Real-time deembedding and triggering is fast. You trigger on the corrected waveform, and you see this corrected waveform.
- Zoom
- Cursor measurements
- Automatic measurements
- Histogram
- Mathematics. In formulas, use DIFF1, DIFF2, COMMON1, COMMON2.
- FFT, spectrogram
- Mask test
- Export of waveform data. You can export the differential and common waveforms if they are selected as output and displayed. Raw data is not available.
  Without deembedding, P and N waveforms can be exported if they are selected as output and displayed (source = input channel). With software deembedding, only differential and common waveforms are available for export.

Search on differential signals and their input channels is not supported.

## 5.9 Setting up the waveform

This chapter contains the fundamental procedures for setting up the acquisition and adjusting the channel waveforms.

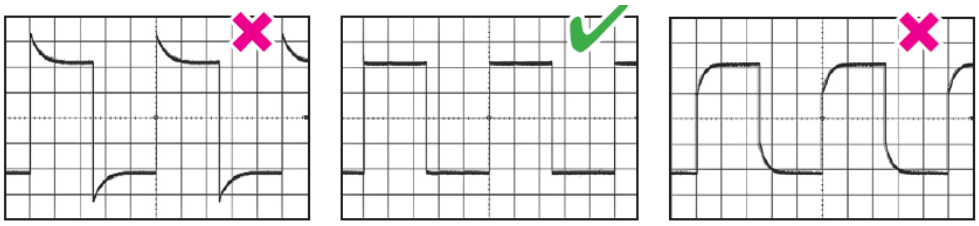
### 5.9.1 Adjusting passive probes

When using a passive probe, you have to compensate it when you connect it to the instrument the first time. Compensation matches the probe cable capacitance to the oscilloscope input capacitance to assure good amplitude accuracy from DC to upper bandwidth limit frequencies. A poorly compensated probe reduces the performance of the probe-oscilloscope system and introduces measurement errors resulting in distorted waveforms and inaccurate results.

Two connector pins are located on the front panel. The ⊥ pin is on ground level. The ⎍ pin supplies a square wave signal with 1 kHz for low frequency probe compensation.

1. Connect the BNC connector of the probe to one of the channel inputs.
2. Connect the probe's ground connector to the ground compensation pin, and the probe tip to the signal pin.
3. Press [Autoset].

   A square wave appears on the display.
4. Adjust the compensation trimmer of the probe to optimum square wave response. For details, refer to the documentation of your probe.



### 5.9.2 Setting up the signal input with autoset

Autoset is the solution for the major part of routine test-setup. It is also a good start if you need to use more complex trigger settings. Autoset finds appropriate horizontal and vertical scales, vertical offset, and trigger conditions to present a stable waveform.

1. Connect the probe to the input connector [C<n>].

   The instrument recognizes the probe and turns on the channel.
2. Press the [Autoset] key.

### 5.9.3 Adjusting the signal input manually

1. Connect the probe to the input connector [C<n>].

   The instrument recognizes the probe and turns on the channel.

2. Open "Menu" > "Horizontal" > "Setup" tab.
3. Set the "Time scale".
4. If you want to analyze the signal some time before or after the trigger, use the "Position" and "Reference point" to adjust the visible section of the waveform.
5. Press the channel button corresponding to the input channel. It is illuminated with the color of the channel waveform.
6. Open the "Menu" > "Vertical" > "Setup" tab.
7. Select the "Coupling".
8. Adjust the vertical "Scale", and the vertical "Position".
9. Proceed with the acquisition setup, see Chapter 5.9.4, "Setting the acquisition", on page 190.

### 5.9.4 Setting the acquisition

Prerequisites:

- Probes are connected.
- Vertical and horizontal settings are adjusted.

The settings are described in Chapter 5.4.2, "Mode settings", on page 131.

1. Open "Menu" > "Acquire" > "Mode" tab.
2. To configure the waveform-specific acquisition settings, enable "Multi waveform". Activate the waveforms "W1"/"W2"/"W3" you need.
   You can set up and display up to three waveforms per channel.
3. Select the "Decimation" - for example, Peak detect or High res.
4. Select the "Arithmetic" - for example, Average or Envelope.
   The instrument precludes incompatible combinations, like "Peak detect" with "Average".
5. If "Arithmetic"="Average" is selected for a waveform, enter the "Average count", that is the number of waveforms used for average calculation.
6. Set the aritmetic reset condition for the average and envelope calculation:
   a) If "Arithmetic Reset > Mode"="Time" is selected, enter the "Time".
   b) If "Arithmetic Reset > Mode"="Waveforms" is selected, enter the "Count".

### 5.9.5 Starting and stopping acquisition

You can control the acquisition in two ways:

- Running continuous acquisition until you stop it.
- Running one acquisition or a given number of acquisitions.

If "Envelope" or "Average" is selected in the "Acquisition" tab, one acquisition means a cycle containing as many acquired waveforms as required to satisfy the reset conditions.

Prerequisites:

- Probes are connected.
- Vertical and horizontal settings are adjusted.
- Triggering is set.
- Channels to be acquired are turned on.

**To start and stop continuous acquisition**

1. Check if the trigger mode is set to "Norm". The trigger mode is shown in the trigger label in the upper right corner of the screen.
   If not, press the trigger [Auto Norm] key on the front panel to toggle the setting.
2. Press the [Run / Stop] key to start acquisition.
   The acquisition starts if a trigger occurs.
3. To stop , press the [Run / Stop] key again.
   The acquisition stops immediately.

**To acquire a limited number of acquisitions**

1. Check if the trigger mode is set to "Norm". The trigger mode is shown in the trigger label in the upper right corner of the screen.
   If not, press the trigger [Auto Norm] key on the front panel to toggle the setting.
2. Open "Menu" > "Acquire" > "Mode" tab.
3. Enter the number of acquisitions in the "Average count" field.
4. Press the [Single] key on the front panel.
   You can stop the running acquisition before it is finished by pressing the key again.

### 5.9.6 Using the roll mode

The roll mode can be used if the acquisition process is slow - that is if the time scale is large. In roll mode, the instrument shows the waveform immediately and saves waiting for the waveform display. The roll mode can be activated by the instrument if several conditions are fulfilled.

**To set the roll mode manually**

1. Make sure that all requirements for the roll mode are fulfilled: see Chapter 5.2, "Horizontal", on page 122.
2. Open "Menu" > "Horizontal" > "Roll" tab.
3. Set "Mode" to "Auto".

4. Set the "Start roll time", the acquisition time at which the instrument starts the roll mode.

### 5.9.7 Using fast segmentation

Fast Segmentation reduces the dead time between two waveform acquisition cycles.

The settings are described in Chapter 5.4.3, "Segmented settings", on page 134.

1. Open "Menu" > "Acquire" > "Segmented" tab.
2. Tap "Fast segmentation" to activate the fast segmentation mode.
3. If you want to sample the maximum number of acquisitions in a series, enable "Acquire maximum".
   If you want to capture a defined number of acquisitions, disable "Acquire maximum" and enter the "Required" number of acquisitions.
4. Set the "Display time".

# 6 Triggers

## 6.1 Basics of triggering

Triggering means to capture the interesting part of the relevant waveforms. Choosing the right trigger type and configuring all trigger settings correctly allows you to detect various incidents in analog, digital, and protocol signals.

### Trigger

A trigger occurs if the complete set of trigger conditions is fulfilled. The trigger is the determining point in the waveform record. The instrument acquires continuously and keeps the sample points to fill the pre-trigger part of the waveform record. When the trigger occurs, the instrument continues acquisition until the post-trigger part of the waveform record is filled. Then it stops acquiring and waits for the next trigger. When a trigger is recognized, the instrument does not accept another trigger until the acquisition is complete, and the holdoff time has expired.

### Trigger setup

A simple trigger setup includes:

- Source of the trigger signal, its coupling and filtering
- Trigger type selection and setup
- Horizontal position of the trigger: see: Chapter 5.1.3.2, "Horizontal position", on page 118
- Trigger mode

The R&S RTP provides various trigger types for troubleshooting and signal analysis, for example, edge trigger, glitch trigger, interval trigger, pattern trigger, and much more.

For complex tasks like verifying and debugging designs, advanced trigger settings are available:

- Hysteresis: rejection of noise to avoid unwanted trigger events caused by noise
- Holdoff to define exactly which trigger event causes the trigger
- Qualification to consider the states of digital signals on other input channels and their logical combination
- Trigger sequences to combine two trigger type conditions

### Action on trigger

A trigger can initiate one or more actions, for example, saving a screenshot or saving waveform data. All available actions can be initiated at the same time.

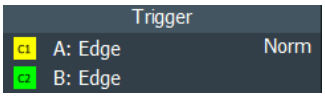
### Trigger sequence

A complex trigger sequence joins two or more separate trigger conditions with an optional delay time and an optional reset time or reset condition. Similar setups are also known as multi-step trigger or A/B trigger.

**Trigger information**

Information on the most important trigger settings is shown in the trigger label on top of the signal bar. If you double-tap the trigger label, the "Trigger" dialog box opens. The label shows:

- Trigger mode
- Trigger type, edge/polarity and trigger source, for A- and B-trigger




## 6.2 Setting up the trigger

This chapter provides step-by-step procedures for the important stages of trigger setup.

### 6.2.1 Configuring a simple trigger

Prerequisites:

- Horizontal and vertical settings are set appropriately to the signals.
- The acquisition is running, the [Run / Stop] key lights green.

For details on settings, see Chapter 6.3, "Trigger types", on page 196.

Proceed as follows:

1. Open the "Menu">"Trigger" dialog.

   The "Trigger" dialog box opens with the "Setup" tab.
2. Tap the "Source" button and select the trigger source.
3. Tap the "Type" button and select the trigger type.
4. Under "Trigger type dependent settings", configure the settings for the selected trigger type.
   See: Chapter 6.3, "Trigger types", on page 196
5. To set the trigger level automatically, tap "Find level".
6. Set the normal trigger mode. Do one of the following:
   - Press the [Auto Norm] key on the front panel until "Normal" is shown in the trigger label.
   - In the "Mode / Holdoff" tab, set the "Trigger mode" to "Normal".

### 6.2.2 Positioning the trigger

By positioning the trigger on the time axis, you define which part of the waveform is displayed: mainly the pre-trigger part, or the post-trigger part, or the part around the trigger point.

For details on position settings, see Chapter 5.4, "Acquire settings", on page 128.

1. Open the "Menu"> "Horizontal" dialog.
2. Set the "Reference point" and the "Position".
   If you want to set the trigger position outside the waveform display, make sure that "Restrict horizontal position to acquisition range" is disabled.

### 6.2.3 Using holdoff

For details on holdoff settings, see Chapter 6.4, "Mode / Holdoff", on page 225.

1. Open the "Menu">"Trigger" dialog.
2. Select the "Mode / Holdoff" tab.
3. Select the "Holdoff mode".
4. Enter the "Holdoff settings" belonging to the selected mode.

### 6.2.4 Setting up an A → B → R trigger sequence

The complete configuration of a complex "A → B → R" trigger sequence consists of:

- A-trigger condition
- B-trigger condition in the same way as for the A-trigger, and optional delay time between the two triggers
- Optional reset by timeout and/or R-trigger

For details on sequence settings, see Chapter 6.8, "Sequence", on page 234.

1. Open the "Menu" > "Trigger" dialog.
2. Set "Trigger on" = "Sequence".
3. Select the type of "Sequence": "A → B → R".
4. Tap "Setup A trigger".

   A dialog opens where you can configure the first event.
   See: Chapter 6.2.1, "Configuring a simple trigger", on page 194.
5. Tap "Setup B trigger" and configure the B-trigger condition.

   A dialog opens where you can configure the second event.

6. Optionally, set the "After 1st event wait" that the instrument waits after an A-trigger until it recognizes B-triggers.
7. Set the "2nd event count". The last B-trigger causes the trigger.
8. You can also define a reset condition. If no B-trigger occurs and the reset condition is fulfilled, the sequence restarts with the A-trigger.
   a) To specify a reset by timeout, enable "Reset on timeout", and enter the "Rest timeout".
   b) To specify a reset trigger type condition, enable "Reset event" and configure the reset trigger type.
   The trigger types and settings depend on the A and B trigger settings. The instrument provides only possible, reasonable combinations.

### 6.2.5 Qualifying the trigger

Qualification considers the states of digital signals on other input channels and their logical combination as an additional trigger condition. For example, an edge trigger is configured for channel 1, and the instrument triggers only if the signal on channel 2 is high.

For details on qualification settings and restrictions, see Chapter 6.7, "Qualification", on page 231.

1. On the "Menu", select "Trigger" > "Qualify" tab.
2. Set one or more channels with the input signal to be used as qualifying signals to "On".
   The channel that is the trigger source is not available for qualification.
3. Check and set the threshold levels for all used channels, that is, the thresholds for digitization of analog signals.
4. Set the signal state for each channel: high or low.
5. If more than one channel is selected, set the logical combination of the channel states.
6. If not yet enabled, tap "Qualify" to enable the qualification.

## 6.3 Trigger types

The setup of the trigger type is the most important part of the trigger definition. It determines the method to identify specific signal phenomena. Almost all trigger types are available for all conditions in a trigger sequence, that is, you can combine different types in the sequence. The instrument checks the trigger settings for compatibility and feasibility, and disables settings that do not fit the previous settings in the sequence.

The settings in the tab are:

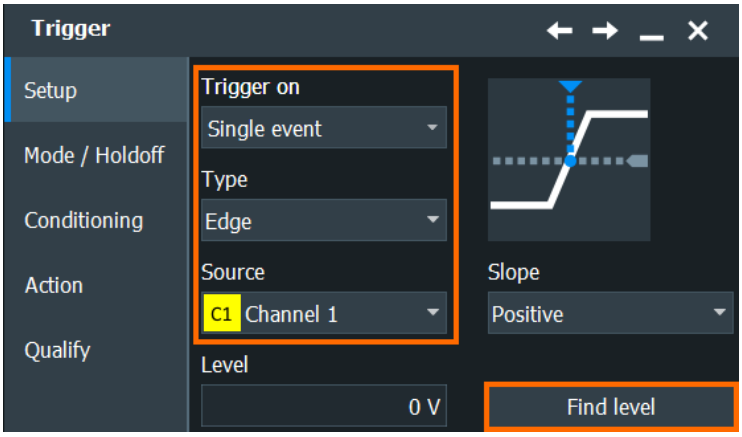
- Basic trigger settings....197
- Edge....199
- Glitch....200
- Width....201
- Runt....202
- Window....204
- Timeout....206
- Interval....207
- Slew rate....208
- Setup & Hold....210
- State....212
- Pattern....213
- Serial pattern....216
- TV/Video trigger....216
- Line trigger....220
- Hardware CDR trigger....220
- Triggering on serial buses....225
- Triggering on parallel buses and digital channels....225


### 6.3.1 Basic trigger settings

Access: "Menu" > "Trigger" > "Setup" tab

The basic trigger settings are the trigger source and the trigger type, including the trigger level. These settings are specific for each condition in a trigger sequence. To set the trigger level automatically, use "Find level".

Depending on the trigger type, additional settings are available. They are explained in the type-specific section.




**Trigger on**
Selects, if you want to trigger on a single event, or on a series of events.

Remote command:
`TRIGger<m>:EVENt[:EVENt]` on page 1367

**Source**
Selects the source of the trigger signal for the current trigger condition. The trigger source works even if it is not displayed in a diagram. It must be synchronized to the signal to be displayed and analyzed.

The trigger source can be:
- Channel 1...4: An analog input channel
- Extern: External analog signal connected to the external trigger input
  The external trigger source is supported for the "Single event" sequence. It is not available if a longer trigger sequence is selected, or if qualification is enabled.
- Line: The instrument generates the trigger from the AC power input and synchronizes the signal to the AC power frequency. Use this source if you want to analyze signals related to the power line frequency, such as lighting equipment and power supply devices.
- Serial bus, D0...D15, Logic, Parallel bus 1...4:
  If options with trigger functionality are installed, the variety of trigger sources is enhanced with specific trigger sources.

Available sources depend on the trigger sequence setting. If "Single event" is selected, all inputs (analog input channels, serial and parallel buses, digital channels) can be used as trigger source. If any other trigger sequence is selected, only channel inputs Ch1...4 can be set as trigger source, and all other input sources are disabled.

See also: Chapter 6.8, "Sequence", on page 234

Remote command:
`TRIGger<m>:SOURce[:SELect]` on page 1305

**Type**
Selects the trigger type specific for each condition in a trigger sequence. The current trigger type is shown on the button.

The following trigger types are available:
- Edge, see page 199
- Glitch, see page 200
- Width, see page 201
- Runt, see page 202
- Window, see page 204
- Timeout, see page 206
- Interval, see page 207
- Slew rate, see page 208
- Setup & Hold, see page 210
- State, see page 212
- Pattern, see page 213
- High speed serial pattern trigger (option R&S RTP-K140/K141), see page 241
- Line trigger, see page 220

Restrictions:
- If the external trigger input is used as trigger source, the analog edge trigger is the only available trigger type.
- For the R-trigger (reset), the trigger types and settings are restricted dependent on the A and B trigger settings. The instrument provides only possible, reasonable combinations.

Remote command:
TRIGger<m>:TYPE on page 1306

**Find level**
Sets the trigger level automatically to *0.5 * (MaxPeak – MinPeak)*. The function is not available for an external trigger source and the TV trigger.

Remote command:
TRIGger<m>:FINDlevel on page 1307

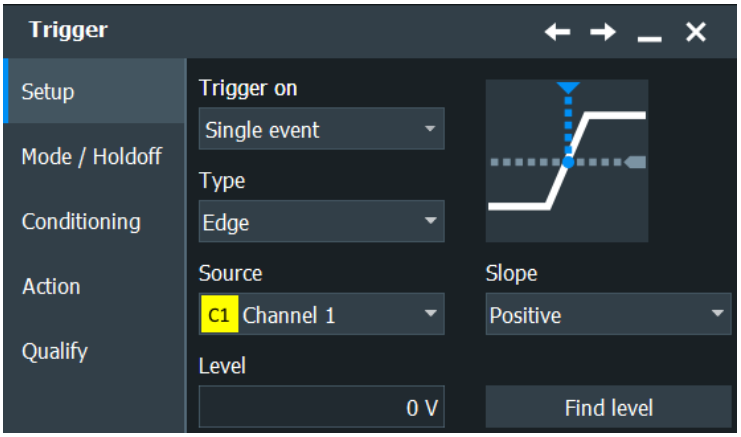
## 6.3.2 Edge

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Edge"

The edge trigger is the most common trigger type. It is well known from analog oscilloscopes; and you can use it for analog and digital signals.

The trigger condition is fulfilled when the signal from the trigger source passes the specified threshold voltage in the specified direction (slope).

If the trigger source is a channel signal, the edge trigger uses the digitized trigger signal. This signal can be qualified and filtered with the DSP filter. If the trigger source is the external trigger input, the coupling and filter for this signal is set directly in the trigger setup.




**Slope**
Sets the edge type for the trigger condition.

| | |
|---|---|
| "Positive" | Selects the rising edge, which is a positive voltage change. |
| "Negative" | Selects the falling edge, which is a negative voltage change. |
| "Both" | Selects the rising as well as the falling edge. This option is not available if the trigger source is the external trigger input. |

Remote command:
TRIGger<m>:EDGE:SLOPe on page 1308
TRIGger<m>:ANEDge:SLOPe on page 1377
TRIGger<m>:SLEW:SLOPe on page 1319

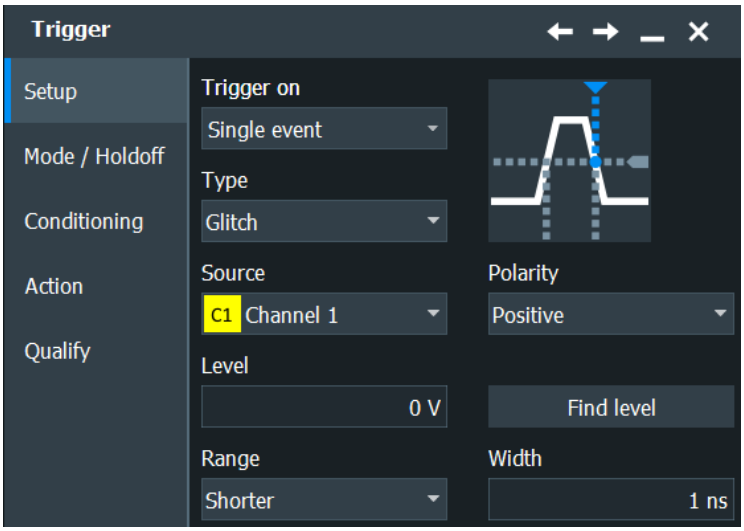
**Level**
Sets the voltage level for the trigger condition. You can also drag the trigger level marker on the display (TA or TB on the right edge of the display).

Remote command:
`TRIGger<m>:LEVel<n>[:VALue]` on page 1306

## 6.3.3 Glitch

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Glitch"

The glitch trigger detects pulses shorter or longer than a specified time. It identifies deviation from the nominal data rate and helps to analyze causes of even rare glitches and their effects on other signals.




**Polarity**
Indicates the polarity of a pulse, which is the direction of the first pulse slope.

"Positive" Selects positive-going pulses.

"Negative" Selects negative-going pulses.

"Either" Selects both positive-going and negative-going pulses.

Remote command:
`TRIGger<m>:GLITch:POLarity` on page 1308
`TRIGger<m>:RUNT:POLarity` on page 1311

**Range**
Selects which glitches are identified: shorter or longer than the specified "Width".

Remote command:
`TRIGger<m>:GLITch:RANGe` on page 1308

**Width**
Sets the length of a glitch. The instrument triggers on pulses shorter or longer than this value. The minimum width is 100 ps.

You need to know the expected pulse widths of the circuit to set the glitch width correctly.

Remote command:
TRIGger<m>:GLITch:WIDTh on page 1309

**Trigger level**
Sets the voltage level for the trigger condition. You can also drag the trigger level marker on the display (TA or TB on the right edge of the display). The range of the trigger level is limited in a way so that always a hysteresis for stable trigger conditions is available.

Remote command:
TRIGger<m>:LEVel<n>[:VALue] on page 1306

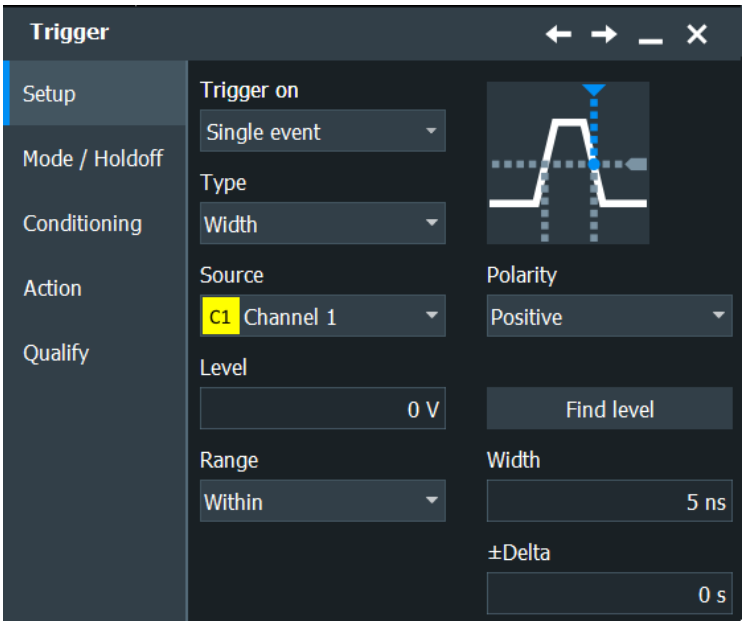
## 6.3.4 Width

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Width"

The width trigger compares the pulse width (duration of a pulse) with a given time limit. It detects pulses with an exact pulse width, pulses shorter or longer than a given time, and pulses inside or outside the allowable time range. The pulse width is measured at the trigger level.

Using the width trigger, you can define the pulse width more precisely than with the glitch trigger. However, using the range settings "Shorter" and "Longer", you can also trigger on glitches.

The width trigger can only analyze **either** positive **or** negative polarity, but searching for a width is also possible for both polarities at the same time ("Either").




**Polarity**
Indicates the polarity of a pulse, which is the direction of the first pulse slope.

"Positive" Triggers on positive-going pulses.

"Negative" Triggers on negative-going pulses.

Remote command:
`TRIGger<m>:WIDTh:POLarity` on page 1309

**Range**
Selects how the range of a pulse width is defined:

"Within" Triggers on pulses inside a given range. The range of the pulse width is defined by "±Delta" related to "Width".

"Outside" Triggers on pulses outside a given range. The range definition is the same as for "Within" range.

"Shorter" Triggers on pulses shorter than the given "Width".

"Longer" Triggers on pulses longer than the given "Width".

Remote command:
`TRIGger<m>:WIDTh:RANGe` on page 1310

**Width**
For the ranges "Within" and "Outside", the width defines the center of a range which is defined by the limits ±Delta.

For the ranges "Shorter" and "Longer", the width defines the maximum and minimum pulse width, respectively.

Remote command:
`TRIGger<m>:WIDTh:WIDTh` on page 1310

**±Delta**
Defines a range around the given width value.

The combination "Range = Within" and "±Delta" = 0 triggers on pulses with a pulse width that equals "Width".

The combination "Range = Outside" and "±Delta" = 0 means to trigger on pulse widths ≠ "Width".

**Trigger level**
Sets the voltage level for the trigger condition. You can also drag the trigger level marker on the display (TA or TB on the right edge of the display). The range of the trigger level is limited in a way so that always a hysteresis for stable trigger conditions is available.

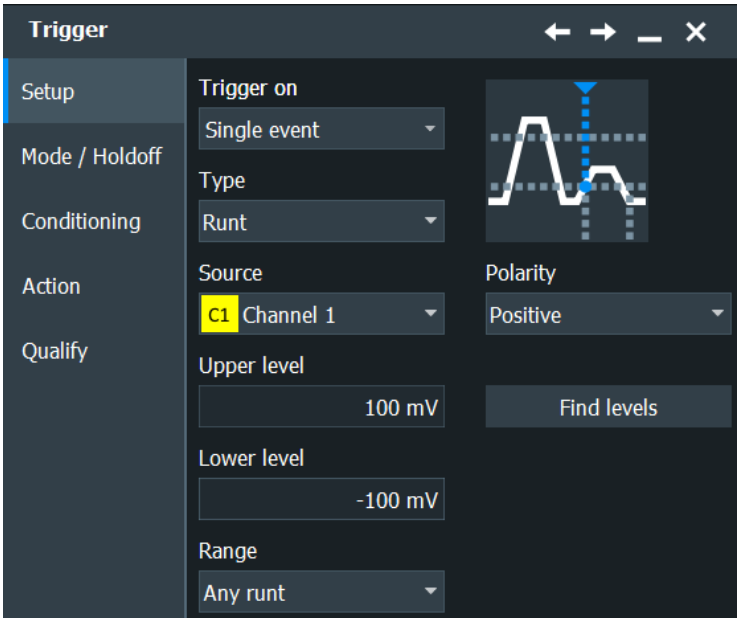
Remote command:
`TRIGger<m>:LEVel<n>[:VALue]` on page 1306

### 6.3.5 Runt

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Runt"

A runt is a pulse lower than normal in amplitude. The amplitude crosses the first threshold twice in succession without crossing the second one. In addition to the

threshold amplitudes, you can define a time limit for the runt in the same way as for width triggers. For example, this trigger can detect logic, digital, and analog signals remaining below a specified threshold amplitude because I/O ports are in undefined state.




**Polarity**
Indicates the polarity of a pulse, which is the direction of the first pulse slope.

"Positive" Selects positive-going pulses.

"Negative" Selects negative-going pulses.

"Either" Selects both positive-going and negative-going pulses.

Remote command:
`TRIGger<m>:GLITch:POLarity` on page 1308
`TRIGger<m>:RUNT:POLarity` on page 1311

**Upper level**
Sets the upper voltage threshold.

Remote command:
`TRIGger<m>:LEVel<n>:RUNT:UPPer` on page 1312

**Lower level**
Sets the lower voltage threshold.

Remote command:
`TRIGger<m>:LEVel<n>:RUNT:LOWer` on page 1312

**Range**
Selects how the time limit of the runt pulse is defined:

"Any runt" Triggers on all runts fulfilling the level condition, without time limitation.

| | |
|---|---|
| "Longer" | Triggers on runts longer than the given "Runt width". |
| "Shorter" | Triggers on runts shorter than the given "Runt width". |
| "Within" | Triggers if the runt length is inside a given time range. The range is defined by "Runt width" and "±Delta". |
| "Outside" | Triggers if the runt length is outside a given time range. The range definition is the same as for "Within" range. |

Remote command:
`TRIGger<m>:RUNT:RANGe` on page 1312

**Runt width**
For the ranges "Shorter" and "Longer", the runt width defines the maximum and minimum pulse width, respectively.

For the ranges "Within" and "Outside", the runt width defines the center of a range which is defined by "±Delta".

Remote command:
`TRIGger<m>:RUNT:WIDTh` on page 1313

**±Delta**
Defines a range around the given runt width.

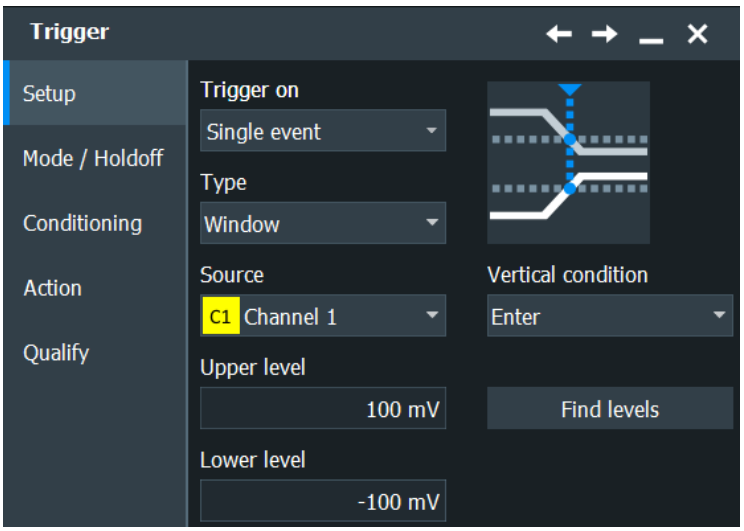
Remote command:
`TRIGger<m>:RUNT:DELTa` on page 1313

### 6.3.6 Window

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Window"

The window trigger checks the signal run in relation to a "window". The window is formed by the upper and lower voltage levels. The trigger condition is fulfilled, if the waveform enters or leaves the window, or if the waveform stays inside or outside for a time longer or shorter than specified.

With the window trigger, you can display longer transient effects.




**Vertical condition**
Selects how the signal run is compared with the window:

"Enter" Triggers when the signal crosses the upper or lower level and thus enters the window made up of these two levels.

"Exit" Triggers when the signal leaves the window.

"Stay within" Triggers if the signal stays between the upper and lower level for a specified time. The time is defined in various ways by the Time condition.

"Stay outside" Triggers if the signal stays above the upper level or below the lower level for a specified time. The time is also defined by the "Time condition".

Remote command:
`TRIGger<m>:WINDow:RANGe` on page 1314

**Upper level**
Sets the upper voltage limit for the window.

Remote command:
`TRIGger<m>:LEVel<n>:WINDow:UPPer` on page 1314

**Lower level**
Sets the lower voltage limit for the window.

Remote command:
`TRIGger<m>:LEVel<n>:WINDow:LOWer` on page 1314

**Time condition**
Selects how the time limit of the window is defined. Time conditioning is available for the vertical conditions "Stay within" and "Stay outside".

"Within" Triggers if the signal stays inside or outside the vertical window limits at least for the time *Width* - *Delta* and for *Width* + *Delta* at the most.

| | |
|---|---|
| "Outside" | "Outside" is the opposite definition of "Within". The instrument triggers if the signal stays inside or outside the vertical window limits for a time shorter than *Width - Delta* or longer than *Width + Delta*. |
| "Shorter" | Triggers if the signal crosses vertical limits before the specified "Width" time is reached. |
| "Longer" | Triggers if the signal crosses vertical limits after the specified "Width" time is reached. |

Remote command:
`TRIGger<m>:WINDow:TIME` on page 1315

**Width**
For the ranges "Within" and "Outside", the width defines the center of a time range which is defined by the limits "±Delta".

For the ranges "Shorter" and "Longer", it defines the maximum and minimum time lapse, respectively.

Remote command:
`TRIGger<m>:WINDow:WIDTh` on page 1315

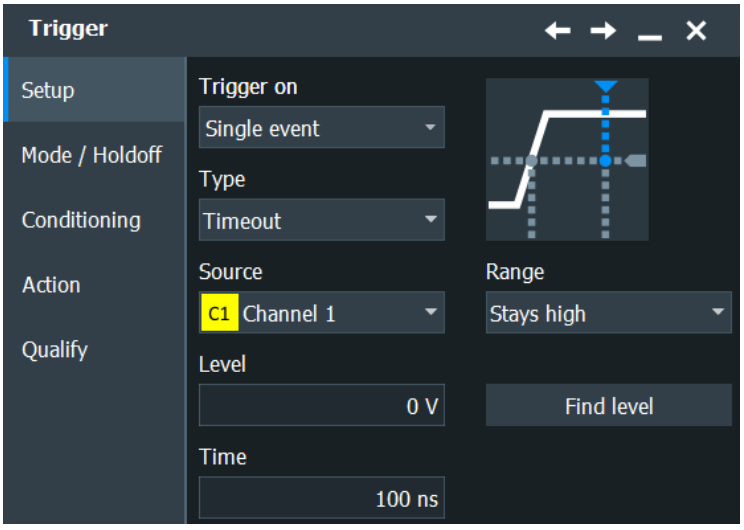
**±Delta**
Defines a range around the "Width" value.

Remote command:
`TRIGger<m>:WINDow:DELTa` on page 1316

### 6.3.7 Timeout

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Timeout"

The timeout trigger checks if the signal stays above or below the threshold voltage for a specified time lapse. In other words, the trigger occurs if the trigger source does not have the expected transition within the specified time.



**Trigger level**
Sets the voltage level for the trigger condition. You can also drag the trigger level marker on the display (TA or TB on the right edge of the display). The range of the trigger level is limited in a way so that always a hysteresis for stable trigger conditions is available.

Remote command:
`TRIGger<m>:LEVel<n>[:VALue]` on page 1306

**Range**
Selects the relation of the signal level to the trigger level:

"Stays high" The signal level stays above the trigger level.

"Stays low" The signal level stays below the trigger level.

"High or low" The signal level stays above or below the trigger level.

Remote command:
`TRIGger<m>:TIMeout:RANGe` on page 1316

**Time**
Defines the time limit for the timeout at which the instrument triggers.

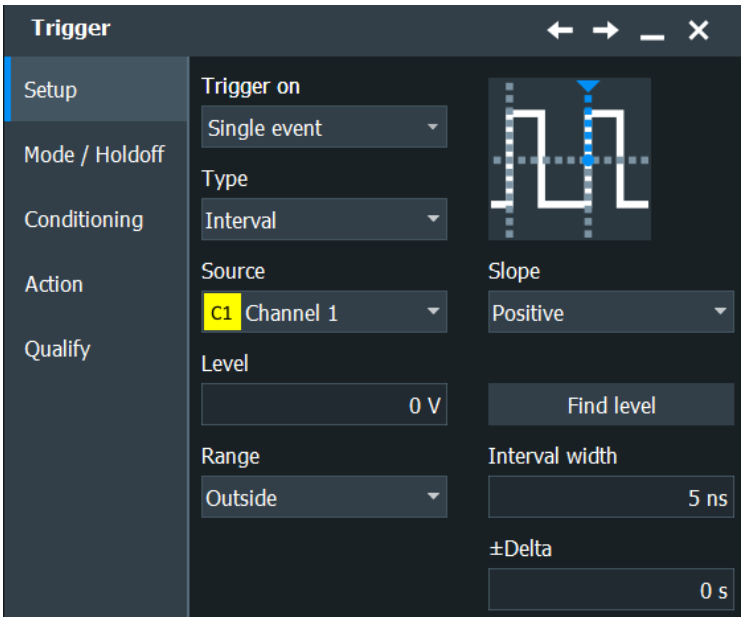
Remote command:
`TRIGger<m>:TIMeout:TIME` on page 1317

## 6.3.8 Interval

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Interval"

The interval trigger analyzes the time between two pulses.

The interval trigger can analyze either rising or falling edges, but searching for an interval is also possible for both edges at the same time ("Either").



**Slope**
Sets the edge for the trigger. You can analyze the interval between positive edges or between negative edges.

Remote command:
TRIGger<m>:INTerval:SLOPe on page 1317

**Trigger level**
Sets the voltage level for the trigger condition. You can also drag the trigger level marker on the display (TA or TB on the right edge of the display). The range of the trigger level is limited in a way so that always a hysteresis for stable trigger conditions is available.

Remote command:
TRIGger<m>:LEVel<n>[:VALue] on page 1306

**Range**
Selects how the range of an interval is defined:

| | |
|---|---|
| "Within" | Triggers on pulse intervals inside a given range. The range is defined by "Interv. width" and "±Delta". |
| "Outside" | Triggers on intervals outside a given range. The range definition is the same as for "Within" range. |
| "Shorter" | Triggers on intervals shorter than the given "Interv. width". |
| "Longer" | Triggers on intervals longer than the given "Interv. width". |

Remote command:
TRIGger<m>:INTerval:RANGe on page 1317

**Interv. width**
Defines the time between two pulses.

Remote command:
TRIGger<m>:INTerval:WIDTh on page 1318

**±Delta**
Defines a range around the "Interval width" value.

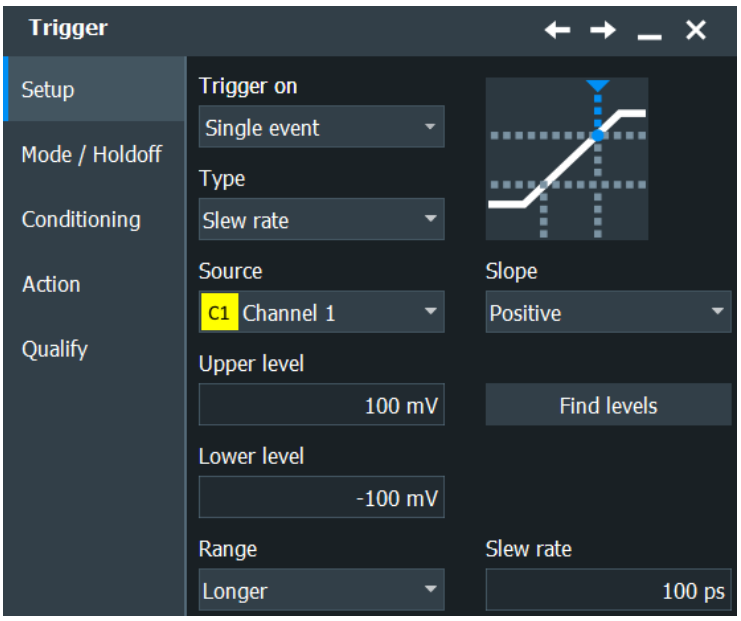
Remote command:
TRIGger<m>:INTerval:DELTa on page 1318

### 6.3.9 Slew rate

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Slew rate"

The slew rate trigger is also known as transition trigger. It triggers if the transition time from the lower to higher voltage level (or vice versa) is shorter or longer as defined, or outside or inside a specified time range.

The slew rate trigger finds slew rates faster than expected or permissible to avoid overshooting and other interfering effects. It also detects slow edges violating the timing in pulse series.




**Slope**
Sets the edge type for the trigger condition.

"Positive" Selects the rising edge, which is a positive voltage change.

"Negative" Selects the falling edge, which is a negative voltage change.

"Both" Selects the rising as well as the falling edge. This option is not available if the trigger source is the external trigger input.

Remote command:
`TRIGger<m>:EDGE:SLOPe` on page 1308
`TRIGger<m>:ANEDge:SLOPe` on page 1377
`TRIGger<m>:SLEW:SLOPe` on page 1319

**Upper level**
Sets the upper voltage threshold. When the signal crosses this level, the slew rate measurement starts or stops depending on the selected slope.

Remote command:
`TRIGger<m>:LEVel<n>:SLEW:UPPer` on page 1319

**Lower level**
Sets the lower voltage threshold. When the signal crosses this level, the slew rate measurement starts or stops depending on the selected slope.

Remote command:
`TRIGger<m>:LEVel<n>:SLEW:LOWer` on page 1319

**Range**
Selects how the time limit for the slew rate is defined. The time measurement starts when the signal crosses the first trigger level - the upper or lower level depending on the selected slope. The measurement stops when the signal crosses the second level.

"Within" Triggers on slew rates inside a given time range. The range is defined by "Slew rate" and "±Delta".

"Outside" Triggers on slew rates outside a given time range. The range definition is the same as for "Within" range.

"Shorter" Triggers on slew rates shorter than the given "Slew rate" limit.

"Longer" Triggers on slew rates longer than the given "Slew rate" limit.

Remote command:
TRIGger<m>:SLEW:RANGe on page 1320

**Slew rate**
For the ranges "Within" and "Outside", the slew rate defines the center of a range which is defined by the limits "±Delta".

For the ranges "Shorter" and "Longer", the slew rate defines the maximum and minimum slew rate limits, respectively.

Remote command:
TRIGger<m>:SLEW:RATE on page 1320

**±Delta**
Defines a time range around the given slew rate.

Remote command:
TRIGger<m>:SLEW:DELTa on page 1321

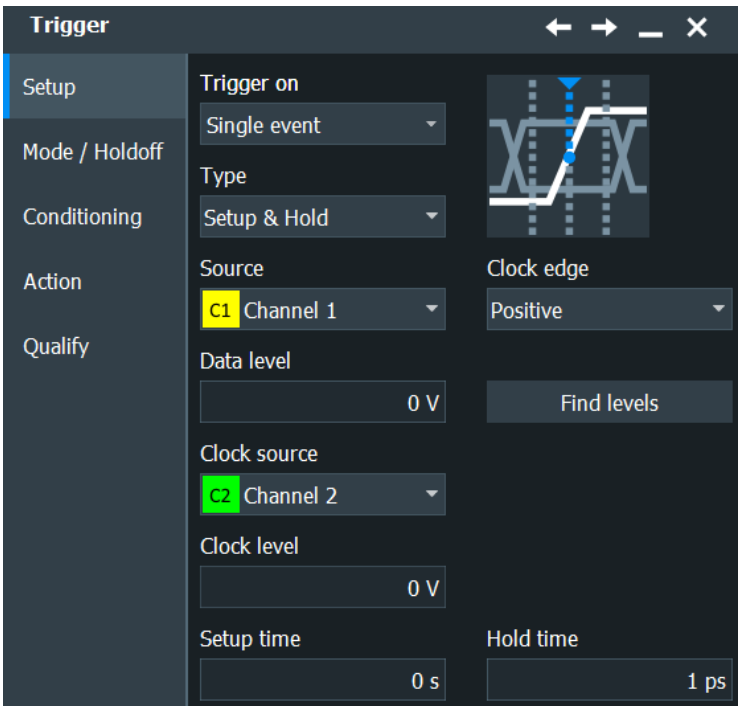
### 6.3.10 Setup & Hold

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Setup & Hold"

With the Setup & Hold trigger you can analyze the relative timing between two signals: a data signal and the synchronous clock signal. Many systems require, that the data signal must be steady for some time before and after the clock edge, for example, the data transmission on parallel interfaces. With this trigger type, you can also test the time correlation of sideband and in-band signals.

In firmware versions < 5.00, the Setup & Hold was called Data2Clock trigger.

The trigger occurs if the data signal crosses the data level during the setup and hold time. The reference point for the time measurement is defined by clock level and clock edge.




**Clock source**
Selects the input channel of the clock signal.

Remote command:
`TRIGger<m>:SETHold:CSOurce[:VALue]` on page 1322

**Clock edge**
Sets the edge of the clock signal to define the time reference point for the setup and hold time:

"Positive" Rising edge, a positive voltage change.

"Negative" Falling edge, a negative voltage change.

"Both" Both the rising and the falling edge.

Remote command:
`TRIGger<m>:SETHold:CSOurce:EDGE` on page 1321

**Clock level**
Sets the voltage level for the clock signal. Both "Clock level" and "Clock edge" define the starting point for calculation of the setup and hold time.

Remote command:
`TRIGger<m>:SETHold:CSOurce:LEVel` on page 1322

**Data level**
Sets the voltage level for the data signal. At this level, the setup time and hold time are measured.

Remote command:
`TRIGger<m>:LEVel<n>[:VALue]` on page 1306

**Use data level for clock level**
Sets the trigger levels and hysteresis values for all channels to the values of the currently selected trigger source. The function affects only the levels defined for the selected condition. The hysteresis of the external trigger input is an independent value, and it is not affected by level coupling.

In trigger sequences, another coupling of trigger levels is possible: "Couple sequence thresholds" on page 236.

Remote command:
`TRIGger<m>:SCOupling` on page 1323

**Setup time**
Sets the minimum time **before** the clock edge while the data signal must stay steady above or below the data level.

The setup time can be negative. In this case, the hold time is always positive. If you set a negative setup time, the hold time is adjusted by the instrument.

Remote command:
`TRIGger<m>:SETHold:STIMe` on page 1323

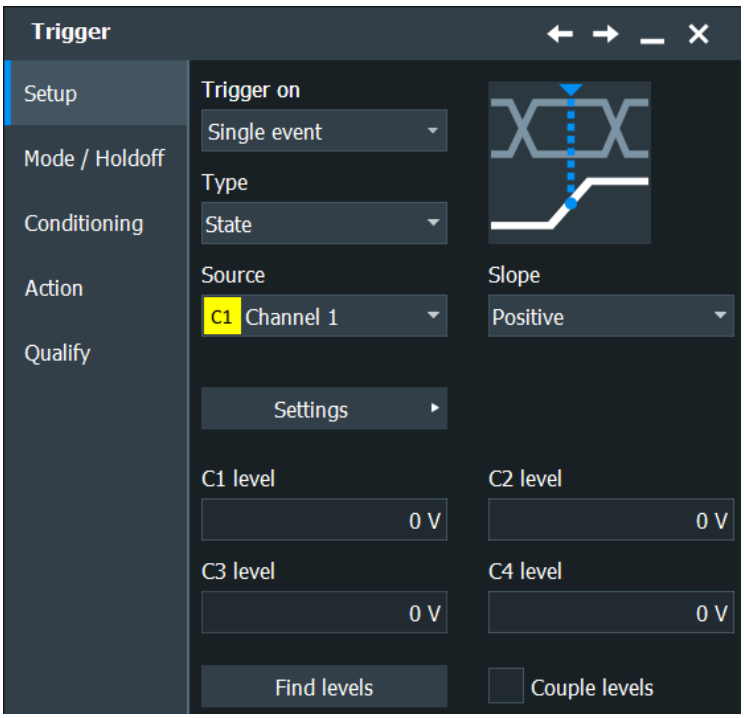
**Hold time**
Sets the minimum time **after** the clock edge while the data signal must stay steady above or below the data level.

The hold time can be negative. In this case, the setup time is always positive. If you set a negative hold time, the setup time is adjusted by the instrument.

Remote command:
`TRIGger<m>:SETHold:HTIMe` on page 1323

### 6.3.11 State

The state trigger is a qualified edge trigger. It combines the edge trigger settings with trigger qualification.




The individual settings are:

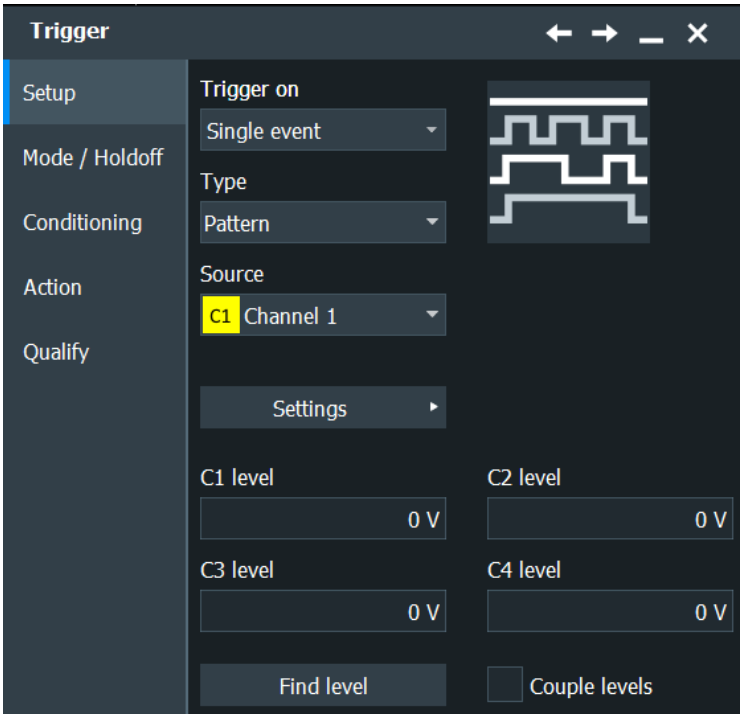
- "Slope" on page 199
- "Pattern" on page 214
- "Trigger Levels" on page 215
- "Find level" on page 199
- "Use data level for clock level" on page 212

### 6.3.12 Pattern

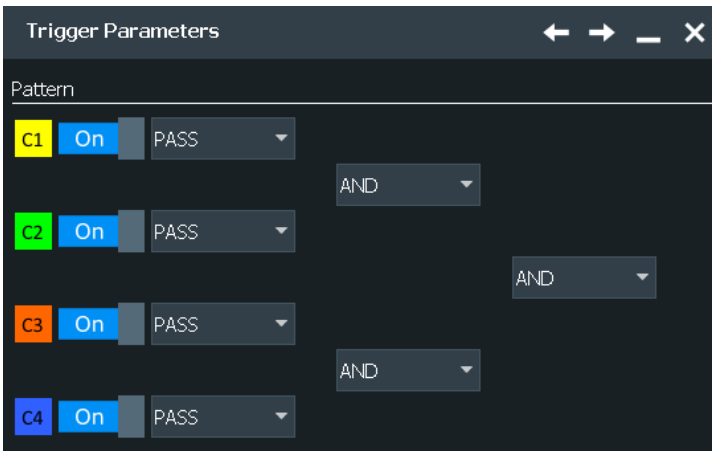
Access: "Menu" > "Trigger" > "Setup" tab > "Type = Pattern"

The pattern trigger is a logic trigger. It provides logical combination of the input channels and supports you in verifying the operation of digital logic. In addition to the pattern and the trigger levels, you can define a timing condition.

The setup of the pattern trigger is similar to trigger qualification. The complete settings for the pattern trigger are provided in the "Qualification" tab.




**Pattern**
The pattern contains the channel selection, and the logical operations structure of hardware-based Boolean logic.




| | |
|---|---|
| "Channel" | Select the channels to be considered. For qualification, you can select all channel signals except for the trigger source. In pattern trigger setup, the trigger source channel is selected by default, and you can select all other channel signals. |
| "Boolean operator" | Defines the logical operation on the digital signal resulting from the comparison with the trigger level.<br>• "PASS": leaves the input value unchanged<br>• "NOT": inverts the input value |

"Logical operator" — Defines the logic combination of two sources. The sources are channel 1/2 and channel 3/4 on the first step, and in the second step the logical combination resulting from the first step.

- "AND": logical AND, conjunctive combination
- "NAND": logical NOT AND
- "OR": logical OR, disjunctive combination
- "NOR": logical NOT OR

Remote command:
`TRIGger<m>:QUALify<n>:A:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:A[:ENABle]` on page 1358
`TRIGger<m>:QUALify<n>:AB:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:ABCD:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:B:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:B[:ENABle]` on page 1358
`TRIGger<m>:QUALify<n>:C:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:C[:ENABle]` on page 1358
`TRIGger<m>:QUALify<n>:CD:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:D:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:D[:ENABle]` on page 1358

**Trigger Levels**
Define the trigger level for each input channel. For state and pattern trigger, the trigger level is a decision threshold: If the signal value is higher than the trigger level, the signal state is high (1 or true for the Boolean logic). Otherwise, the signal state is considered low (0 or false) if the signal value is below the trigger level.

These trigger levels are also used in qualification setup.

You can set the trigger levels for all channels to the same value, see "Use data level for clock level" on page 212.

**Additional settings: Timing**
"Timing" adds time limitation to the pattern condition.

You find this setting in the "Qualification" tab.

"Off" — No time limitation. The trigger occurs if the pattern condition is fulfilled.

"Timeout" — Defines how long the result of the pattern condition must be true or false.

"Width" — Defines a time range for keeping up the true result of the pattern condition. The range is defined in the same way as for width and interval triggers, see "Range" on page 202.

Remote command:
`TRIGger<m>:PATTern:MODE` on page 1324
`TRIGger<m>:PATTern:TIMeout:MODE` on page 1325
`TRIGger<m>:PATTern:TIMeout[:TIME]` on page 1325
`TRIGger<m>:PATTern:WIDTh:DELTa` on page 1327
`TRIGger<m>:PATTern:WIDTh:RANGe` on page 1326
`TRIGger<m>:PATTern:WIDTh[:WIDTh]` on page 1326

### 6.3.13 Serial pattern

Serial pattern is available, if option R&S RTP-K140/K141 is installed.

See Chapter 6.10, "High speed serial pattern trigger (option R&S RTP-K140/K141)", on page 241.

### 6.3.14 TV/Video trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Type = TV"

The TV or video trigger is used to analyze analog baseband video signals. You can trigger on baseband video signals from standard definition and high definition standards, and also on user-defined signals.

The instrument triggers on the line start - the horizontal sync pulse. You can trigger on all lines, or specify a line number. You can also trigger on the field or frame start.

Also, a delay can be set: Set the "Holdoff events" in the "Mode / Holdoff" tab to the number of fields to be skipped.

For details on holdoff settings, see Chapter 6.4, "Mode / Holdoff", on page 225.

Make sure that the trigger level crosses the synchronizing pulses of the video signal, see "Trigger level" on page 218.

Most video signals have an output impedance of 75 Ω. The channel inputs of the R&S RTP have an input impedance of 50 Ω. Make sure to provide the adequate matching to ensure amplitude fidelity. Using the R&S RT-Z1M adapter and a simple 75 Ω feed-through termination is suitable for most applications.

Once the trigger is set correctly, you can use cursor and automatic measurements to perform amplitude and timing measurements.

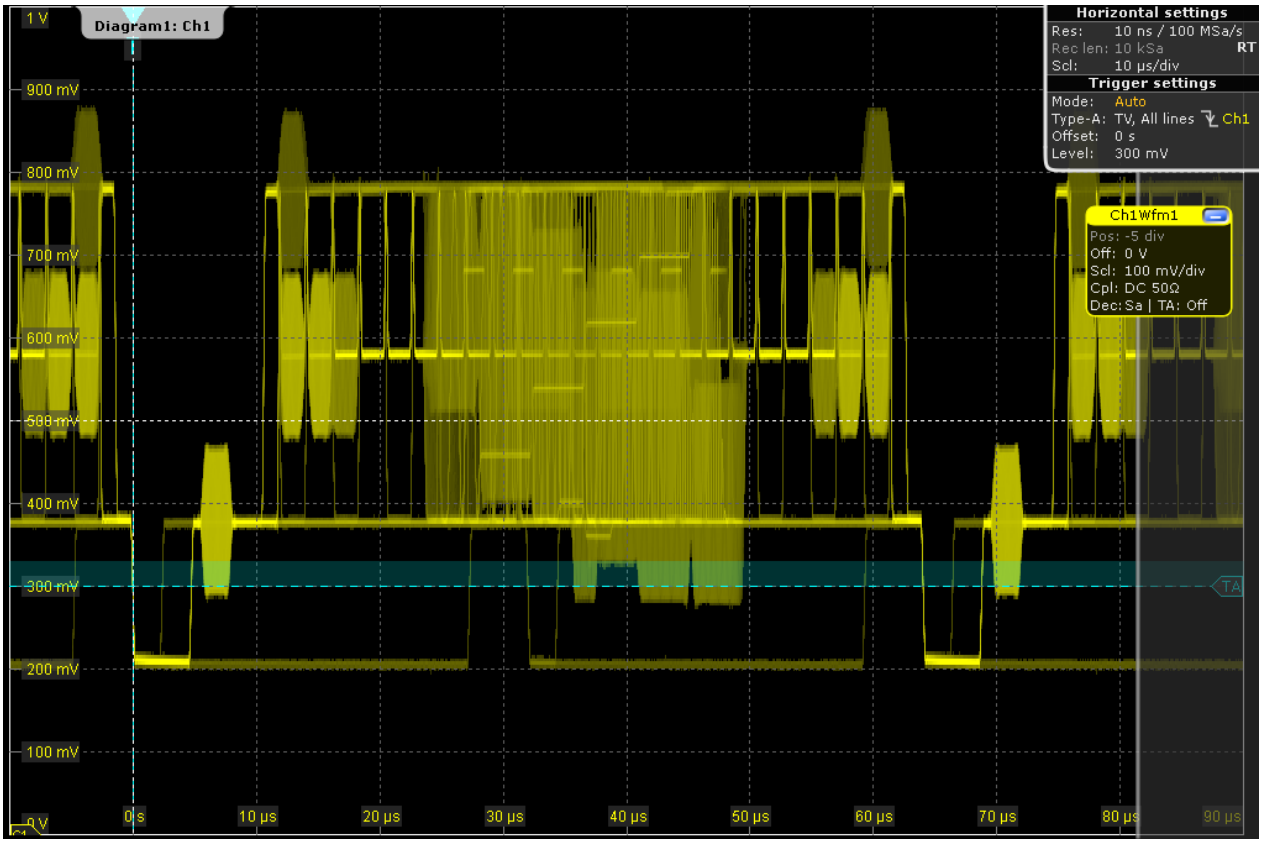



*Figure 6-1: Trigger on all lines of a PAL signal with positive signal polarity, trigger level = 300 mV*

**Standard**
Selects the TV standard or "Custom" for user-defined signals.

HDTV standards are indicated by the number of active lines, the scanning system (p for progressive scanning, i for interlaced scanning) and the frame rate. For interlaced scanning, the field rate is used instead of the frame rate. 1080p/24sF is an HDTV standard using progressive segmented frame scanning.

"Custom" can be used for signals of other video systems, for example, medical displays, video monitors, and security cameras. To trigger on these signals, you have to define the pulse type and length of the sync pulse, the scanning system and the line period.

Remote command:
`TRIGger<m>:TV:STANdard` on page 1352

**Mode**
Selects the lines or fields on which the instrument triggers. Available modes depend on the scanning system of the selected standard.

| | |
|---|---|
| "All fields" | Triggers on the first video line of the frame (progressive scanning) or field (interlaced scanning), for example, to find amplitude differences between the fields. |
| "Odd fields / Even fields" | Triggers on the first video line of the odd or even field. These modes are available for interlaced scanning (PAL, PAL-M, SECAM, NTSC, 1080i) and progressive segmented frame scanning (1080p/24sF). They can be used, for example, to analyze the components of a video signal. |
| "All lines" | Triggers on the line start of all video lines, for example, to find maximum video levels. |
| "Line number" | Triggers on a specified line. Enter the line number in "Line #". |

Remote command:
`TRIGger<m>:TV:MODE` on page 1352

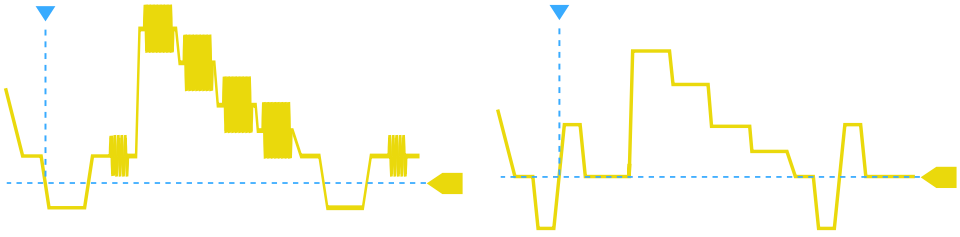
**Line #**
Sets the number of the line to be triggered on if "Mode" is set to "Line number". Usually the lines of the frame are counted, beginning from the frame start.

For NTSC signals, the lines are counted per field, not per frame. Therefore, you have to set the "Field" (odd or even), and the line number in the field.

Remote command:
`TRIGger<m>:TV:LINE` on page 1353
`TRIGger<m>:TV:LFIeld` on page 1354

**Trigger level**
Sets the trigger level as threshold for the sync pulse. Make sure that the trigger level crosses the synchronizing pulses of the video signal.

The hysteresis is set according to the settings in the "Noise Reject" tab.



Remote command:
`TRIGger<m>:LEVel<n>[:VALue]` on page 1306

**Signal polarity**
Sets the polarity of the signal. Note that the sync pulse has the opposite polarity, for example, a positive signal has a negative sync pulse.

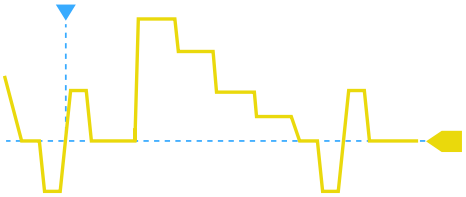


*Figure 6-2: Signal with positive polarity and tri-level sync pulse*

Remote command:
`TRIGger<m>:TV:POLarity` on page 1353

**Pulse type**
Sets the type of the sync pulse, either bi-level sync pulse (used in SDTV signals), or tri-level sync pulse (used in HDTV signals).

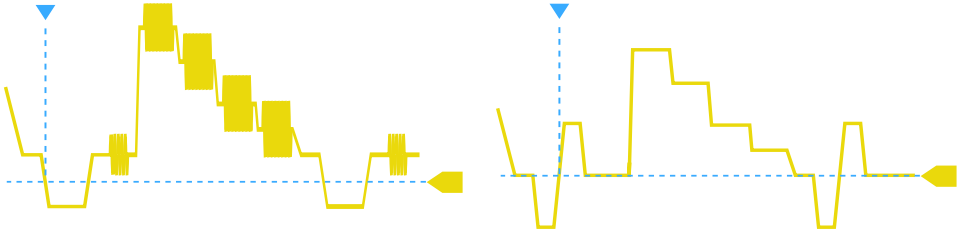
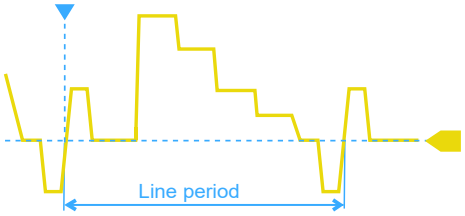


*Figure 6-3: Bi-level (left) and tri-level (right) sync pulses*

This setting is available for user-defined video signals if "Standard" is set to "Custom".

Remote command:
`TRIGger<m>:TV:CUSTom:STYPe` on page 1355

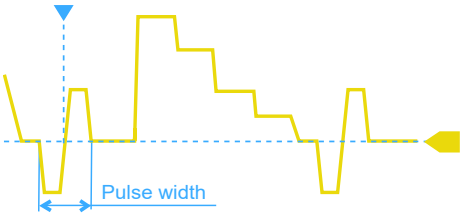
**Line period**
Sets the duration of a single video line, the time between two successive sync pulses.




This setting is available for user-defined video signals if "Standard" is set to "Custom".

Remote command:
`TRIGger<m>:TV:CUSTom:LDURation` on page 1355

**Pulse width**
Sets the width of the sync pulse.




This setting is available for user-defined video signals if "Standard" is set to "Custom".

Remote command:
`TRIGger<m>:TV:CUSTom:SDURation` on page 1356

**Scan**
Sets the scanning system.

This setting is available for user-defined video signals if "Standard" is set to "Custom".

"Interlaced" Interlace scanning uses two fields to create a frame. One field contains all the odd lines (odd, first, or upper filed), the other contains all the even lines of the image (even, second, or lower field). First the lines of the odd filed are processed, then the lines of the even field.

"Progressive" Progressive scanning is a method to capture, transmit and display all lines of a frame in sequence.

"Segmented" Progressive segmented frame uses progressive scanning to capture the frame, and interlaced scanning for transmission and display.

Remote command:
`TRIGger<m>:TV:CUSTom:SCANmode` on page 1355

### 6.3.15 Line trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Source = Line"

The line trigger triggers on the AC power input and synchronizes the signal to the AC power frequency. It is not a trigger type but rather a special trigger source. Use the line source if you want to analyze signals related to the power line frequency, such as lighting equipment and power supply devices.

**Slope**
Selects the rising or falling edges of the AC power input for the trigger condition.

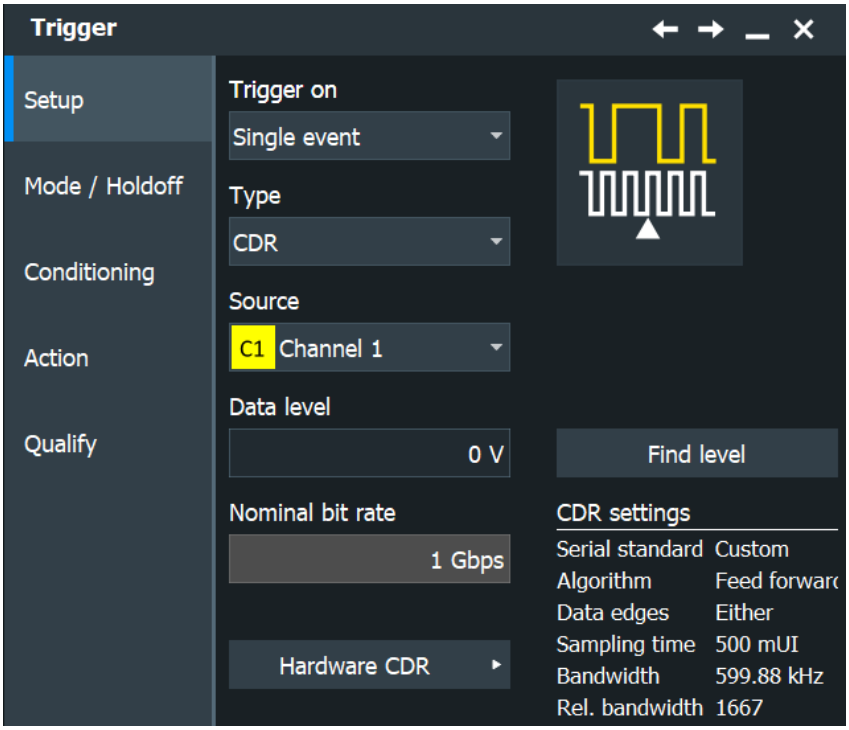
Remote command:
`TRIGger<m>:POWerline:SLOPe` on page 1356

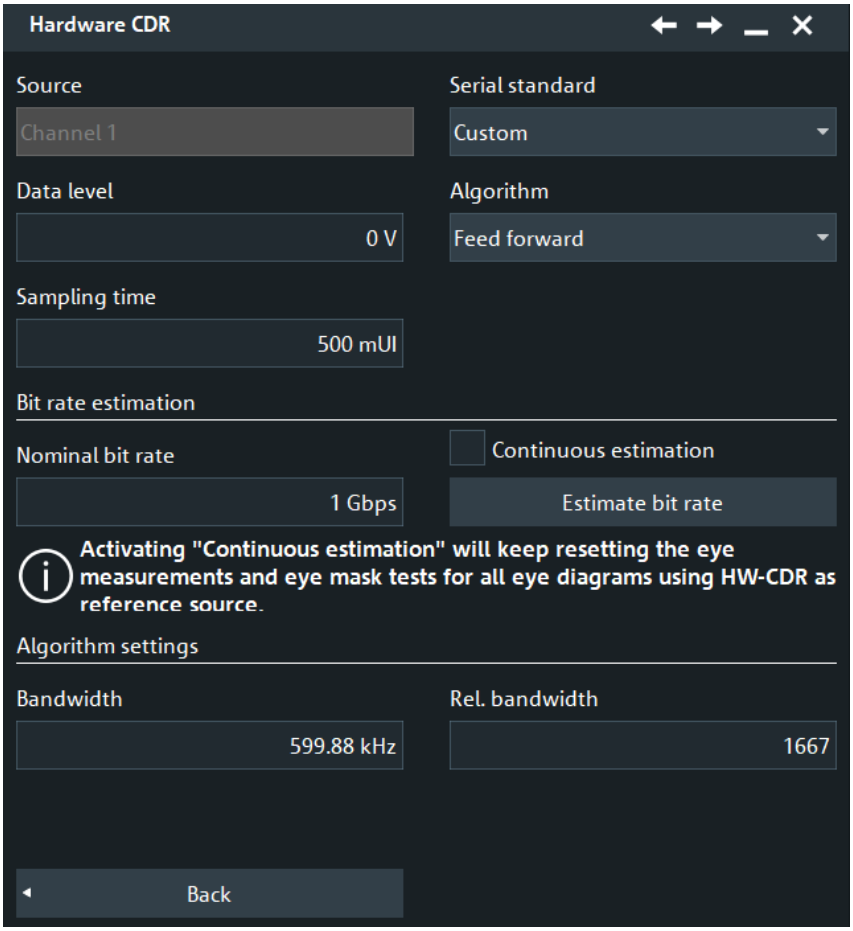
### 6.3.16 Hardware CDR trigger

Access: "Menu" > "Trigger" > "Setup" tab > "Type = CDR" > "Hardware CDR".

The clock data recovery (CDR) trigger triggers on the edges of a clock edge stream that is recovered from a data signal (source) using the hardware CDR.

Hardware CDR is a continuous process, the recovered clock is always synchronized. HW CDR is the best way to analyze the jitter and eyes of continuous signals, and of buses that use Spread Spectrum Clocking. The complete data can be used, there is no synchronization time needed as with SW CDR, and no data must be discarded. Even short acquisitions are usable with HW CDR. For burst signals, HW CDR can be used for decoding, but it is inapplicable for eye and jitter measurement.






**Source**
Selects the channel signal that is used for clock recovery.

The source cannot be changed if the CDR trigger is selected in the trigger setup. In this case, the instrument triggers on the recovered clock; trigger source and CDR source are the same.

Remote command:
`TRIGger<m>:CDR:EYE:SOURce?` on page 1351

**Serial standard**
For some serial standards, the bit rate and the bandwidth are known to the instrument. Select the standard, and the "Nominal bit rate" and bandwidth settings are set automatically, no estimation is required.

| Instruments bandwidth | Number of active channels | Maximum acquisition mode | Maximum nominal bit rate | |
|---|---|---|---|---|
| | | | R&S RTP-K136 | R&S RTP-K137 |
| up to 8 GHz | 1 channel | 40 GSa/s | 8 Gbit/s | 8 Gbit/s |
| | 2 channels:<br>CH1 and CH3 or<br>CH1 and CH4 or<br>CH2 and CH3 or<br>CH2 and CH4 | If serial number > 103000:<br>40 GSa/s<br>If serial number < 103000:<br>20 GSa/s | | |
| | 2 channels:<br>CH1 and CH2 or<br>CH3 and CH4 | 20 GSa/s | | |
| | 3 or 4 channels | 20 GSa/s | | |
| 13 GHz, 16 GHz | 1 channel | 40 GSa/s | 8 Gbit/s | 16 Gbit/s |
| | 2 channels:<br>CH1 and CH3 or<br>CH1 and CH4 or<br>CH2 and CH3 or<br>CH2 and CH4 | 40 GSa/s | | |
| | 2 channels:<br>CH1 and CH2 or<br>CH3 and CH4 | 20 GSa/s | | 8 Gbit/s |
| | 3 or 4 channels | 20 GSa/s | | |

The following predefined standards are currently available:

| Serial standard | Nom. bit rate | Pattern type |
|---|---|---|
| USB 2.0 | 480 Mbit/s | 8b/10b |
| USB 3.2 Gen 1 | 5 Gbit/s | 8b/10b |
| USB 3.1 Gen 2 | 10 Gbit/s | 128b/132b |
| DisplayPort 1.1 | 1.296 Gbit/s | 8b/10b |
| DisplayPort 1.2 | 4.32 Gbit/s | 8b/10b |
| Fibre Channel 1x | 1 Gbit/s | 8b/10b |
| Fibre Channel 2x | 2 Gbit/s | 8b/10b |
| Fibre Channel 4x | 4 Gbit/s | 8b/10b |
| Fibre Channel 8x | 8 Gbit/s | 8b/10b |
| HDMI 1.2 | 1.32 Gbit/s | 8b/10b |
| HDMI 1.4 | 3.4 Gbit/s | 8b/10b |
| HDMI 2.0 | 4.8 Gbit/s | 8b/10b |

| Serial standard | Nom. bit rate | Pattern type |
|---|---|---|
| PCIe Gen 1 | 2.5 Gbit/s | 8b/10b |
| PCIe Gen 2 | 5 Gbit/s | 8b/10b |
| SATA Gen 1 | 1.5 Gbit/s | 8b/10b |
| SATA Gen 2 | 3 Gbit/s | 8b/10b |
| SATA Gen 3 | 6 Gbit/s | 8b/10b |
| SAS Gen1 | 3 Gbit/s | 8b/10b |
| SAS Gen 2 | 6 Gbit/s | 8b/10b |
| SAS Gen 3 | 12 Gbit/s | 8b/10b |
| Thunderbolt 10G | 10.3125 Gbit/s | Single bit pattern |
| XAUI | 3.75 Gbit/s | 8b/10b |

Remote command:
`TRIGger<m>:CDR:EYE:SERStandard` on page 1350

**Data level**
Sets the edge detection threshold for hardware CDR. The threshold is linked to the trigger level of the CDR source.

Remote command:
`TRIGger<m>:CDR:EYE:LEVel` on page 1349

**Algorithm**
Sets the software algorithm that is used for the hardware clock data recovery.

"Constant frequency"
CDR uses the "Nominal bit rate" to generate the clock signal. The method assumes that the frequency of the signal is constant during the complete acquisition.

"Feed forward"
The feed forward algorithm is a mathematical optimization method used for clock data recovery. To set up the algorithm, set the bandwidth and the relative bandwidth.

Remote command:
`TRIGger<m>:CDR:EYE:ALGorithm` on page 1348

**Sampling time**
Sets the normalized offset of the clock edge within a CDR period.

Remote command:
`TRIGger<m>:CDR:EYE:SAMTime` on page 1350

**Bit rate estimation**
Bit rate estimation analyzes the source signal based on the given "Nominal bit rate" and corrects the value.

"Continuous estimation" — Enables ongoing bit rate correction. Activating the "Continuous estimation" keeps resetting the eye measurements and eye mask tests for all eye diagrams using the "Hardware CDR" as a reference source.

"Nominal bit rate" — Sets the frequency of the feed forward CDR. It corresponds to the data rate of the data stream from which the clock is to be recovered.

"Estimate bit rate" — Estimates the bit rate once, for the current acquisition.

Remote command:
`TRIGger<m>:CDR:EYE:BITRate` on page 1348
`TRIGger<m>:CDR:EYE:BREStimate:CONTinuous` on page 1349
`TRIGger<m>:CDR:EYE:BREStimate` on page 1348

**Algorithm settings**
Define the bandwidth settings for "Algorithm" = "Feed forward".

"Bandwidth" — Sets the bandwidth for the feed forward algorithm. It defines the part of the spectrum that the feed forward can follow during synchronization. The bandwidth is usually defined by the serial standard.

"Rel. bandwidth" — Sets the relative bandwidth, which is the ratio of the nominal bit rate to the PLL bandwidth (NBR/BW).

Remote command:
`TRIGger<m>:CDR:EYE:BWIDth` on page 1349
`TRIGger<m>:CDR:EYE:RELBwidth` on page 1350

### 6.3.17 Triggering on serial buses

Protocol analysis including configuration, triggering, and decoding is described in Chapter 13, "Protocol analysis", on page 515.

For information on triggering on serial buses, see the "Trigger" chapter of the relevant protocol.

### 6.3.18 Triggering on parallel buses and digital channels

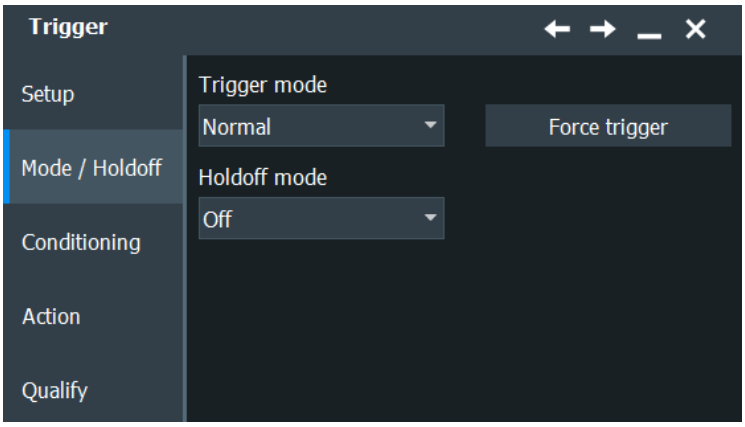
Triggering on digital signals requires the mixed signal option. The option is described in Chapter 14, "Mixed signal option (MSO, R&S RTP-B1)", on page 938.

For information on triggering, see Chapter 14.3.1, "Trigger settings for digital signals and parallel buses", on page 950.

## 6.4 Mode / Holdoff

Access: "Trigger" menu > "Mode / Holdoff" tab

Holdoff conditions define a waiting time after the current trigger until the next trigger can be recognized.




**[Run / Stop]/[Single]**
Front panel keys to start and stop a continuous acquisition or a defined number of acquisition cycles, respectively. The number of acquisitions is set with "Average count".

Remote command:
`RUN` on page 1235
`SINGle` on page 1235
`STOP` on page 1235

**Trigger mode**
Sets the trigger mode which determines the behavior of the instrument if no trigger occurs. The current setting is shown on the trigger label on top of the signal bar.

To toggle quickly between "Auto" and "Normal" mode, use the [Auto Norm] key on the front panel (in "Trigger" section).

"Auto" The instrument triggers repeatedly after a time interval if the trigger conditions are not fulfilled. If a real trigger occurs, it takes precedence. This mode helps to see the waveform even before the trigger conditions are set correctly. The waveform on the screen is not synchronized, and successive waveforms are not triggered at the same point of the waveform. The time interval depends on the timebase settings.

"Normal" The instrument acquires a waveform only if a trigger occurs, that is, if all trigger conditions are fulfilled. If no trigger occurs, no waveform is acquired and the last acquired waveform is displayed. If no waveform was captured before, none is displayed.
When no trigger has been found for longer than one second, a message box appears that shows the time elapsed since the last trigger.

"Free run" The instrument starts acquisition immediately and triggers after a short time interval independent of the timebase settings and faster than in "Auto" mode. Real triggers are ignored. Use this mode if the "Auto" mode is too slow.

Remote command:
`TRIGger<m>:MODE` on page 1370

**Force Trigger**
If the acquisition is running in normal mode and no valid trigger occurs, forcing the trigger provokes an immediate single acquisition. Thus you can confirm that a signal is available and use the waveform display to determine how to trigger on it.

If you need this function frequently, you can add the "Force Trigger" icon to the toolbar.

Remote command:
`TRIGger<m>:FORCe` on page 1371

**Holdoff mode**
Selects the method to define the holdoff condition.

The trigger holdoff defines when the next trigger after the current will be recognized. Thus, it affects the next trigger to occur after the current one. Holdoff helps to obtain stable triggering when the oscilloscope is triggering on undesired events.

Holdoff settings are not available if the trigger source is an external trigger input or serial bus. For the TV trigger, only the "Events" mode is useful.

**Example:**

You want to analyze the first pulse in a burst of several pulses. At first, you select a sufficiently slow timebase to display the entire burst. Then, you set the holdoff time a little longer than the length of the burst. Now, each trigger corresponds to the first pulse in successive bursts, and you can change the timebase to display the waveform in more detail.

The following methods are available:

| | |
|---|---|
| "Time" | Defines the holdoff directly as a time period. The next trigger occurs only after the "Holdoff time" has passed. |
| "Events" | Defines the holdoff as a number of trigger events. The next trigger only occurs when this number of events is reached. The number of triggers to be skipped is defined in "Holdoff events". |
| "Random" | Defines the holdoff as a random time limited by "Minimum time" and "Maximum time". For each acquisition cycle, the instrument selects a new random holdoff time from the specified range.<br>Random holdoff prevents synchronization to discover effects invisible with synchronized triggering, for example, the features of a pulse train. |
| "Auto" | The holdoff time is calculated automatically based on the current horizontal scale.<br>"Auto time scaling" defines the factor that the horizontal scale is multiplied with.<br>"Auto time" shows the resulting holdoff time: *Auto time = Auto time scaling * Horizontal scale*. |
| "Off" | No holdoff |

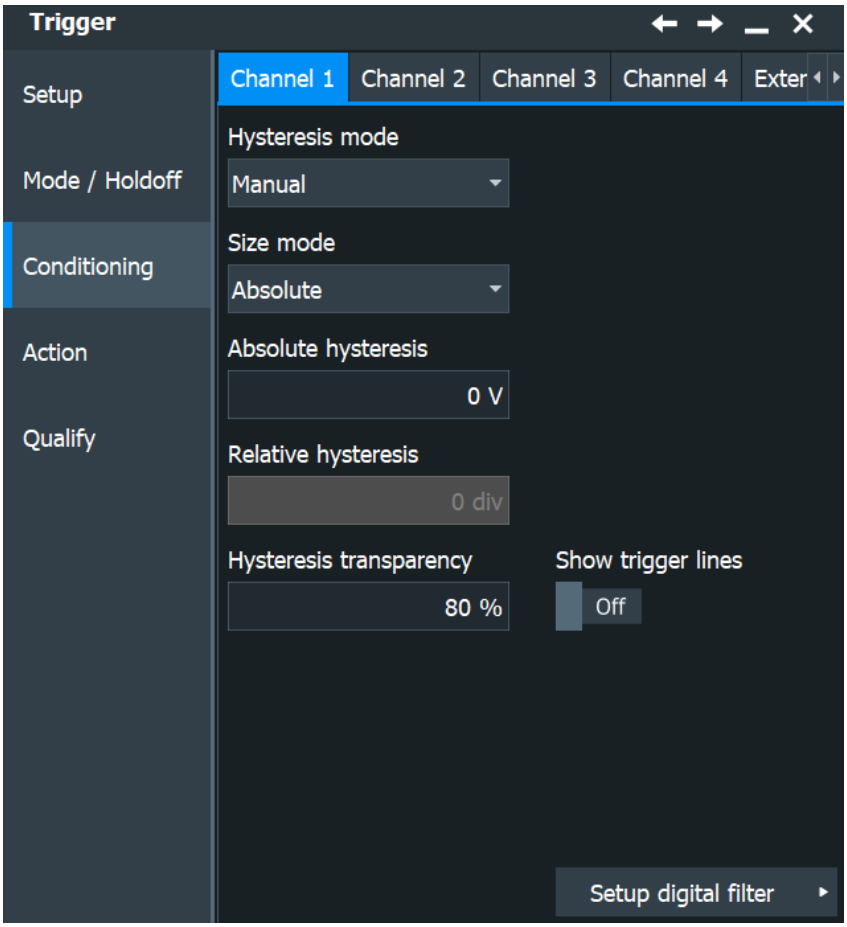
Remote command:
`TRIGger<m>:HOLDoff:MODE` on page 1360
`TRIGger<m>:HOLDoff:TIME` on page 1361
`TRIGger<m>:HOLDoff:EVENts` on page 1362

TRIGger<m>:HOLDoff:MAX on page 1363
TRIGger<m>:HOLDoff:MIN on page 1362
TRIGger<m>:HOLDoff:AUTotime? on page 1363
TRIGger<m>:HOLDoff:SCALing on page 1364

## 6.5 Conditioning

The rejection of noise by setting a hysteresis avoids unwanted trigger events caused by noise oscillation around the trigger level.

You can select the hysteresis mode and value for each channel separately, or couple the trigger levels and set the same hysteresis for channels. The hysteresis of the external trigger input is an independent value, and it is not affected by level coupling.




**Hysteresis mode**
Selects how the hysteresis is set.

"Auto" Automatic hysteresis is the recommended mode. The hysteresis is set by the instrument to reject the internal noise of the instrument.

"Manual" The hysteresis is defined directly in absolute or relative values.

Remote command:
TRIGger<m>:LEVel<n>:NOISe[:STATe] on page 1364

**Size mode**
Selects whether the hysteresis is defined in absolute or relative values. The setting is available only in manual hysteresis mode.

Remote command:
TRIGger<m>:LEVel<n>:NOISe:MODE on page 1365

**Absolute hysteresis**
Defines a range in absolute values around the trigger level. If the signal jitters inside this range and crosses the trigger level, no trigger event occurs.

Remote command:
TRIGger<m>:LEVel<n>:NOISe:ABSolute on page 1365

**Relative hysteresis**
Defines a range in divisions around the trigger level. If the signal oscillates inside this range and crosses the trigger level, no trigger event occurs.

Remote command:
TRIGger<m>:LEVel<n>:NOISe:PERDivision on page 1366
TRIGger<m>:LEVel<n>:NOISe:RELative on page 1366

**Show trigger lines**
Displays the trigger levels and the hysteresis in the diagrams until you disable this option.

Remote command:
DISPlay:TRIGger:LINes on page 1307

**Hysteresis transparency**
Defines the transparency of the hysteresis area above or below the trigger level.

Only visible if Show trigger lines is enabled.

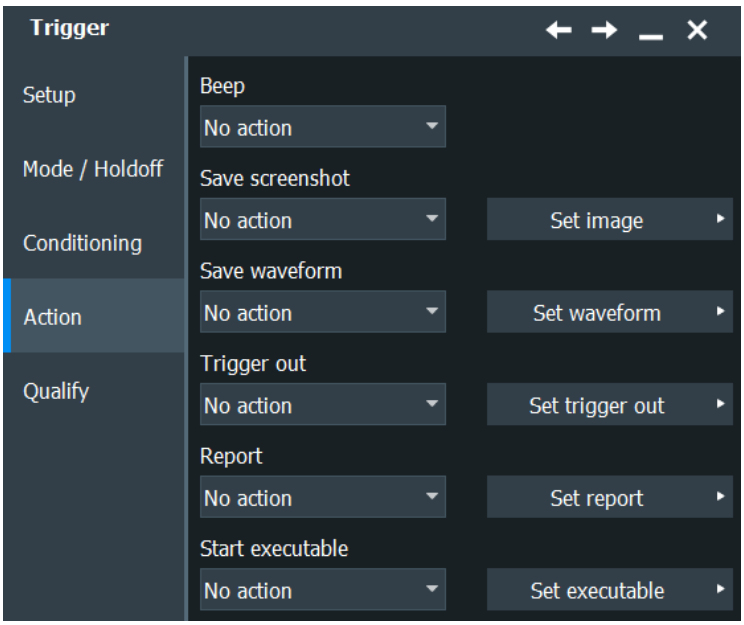
**Hysteresis**
Displays the automatic hysteresis value for "Hysteresis mode" = "Auto".

## 6.6 Actions on trigger

Access: "Menu" > "Trigger" > "Action" tab

The action settings define what happens when a trigger occurs. All available actions can be initiated at the same time.

The R&S RTP can provide an external trigger signal to synchronize the measurements of other instruments. The trigger out signal is also adjusted and enabled in the "Action" tab.



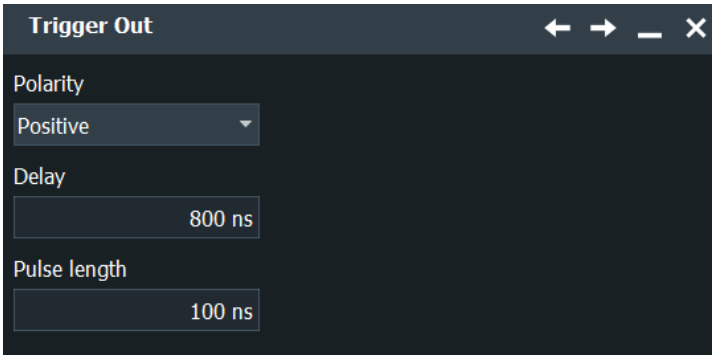

**Actions on trigger**
The trigger can initiate several actions, each time a trigger occurs. To activate an action, set it to "On trigger". The following actions are available:

| | |
|---|---|
| "Beep" | Generates a beep sound. |
| "Save screenshot" | Saves a screenshot according to settings in "Menu" > "Save/Recall" > "Save" tab > "Screenshot". |
| "Save Waveform" | Saves the waveform data to a file according to settings in "Menu" > "Save/Recall" > "Save" tab > "Waveforms". |
| "Trigger out" | Selects, if a pulse is provided to the [Trigger Out] connector on the rear panel.<br>Adjust the pulse with "Set trigger out", see "Trigger out signal setup" on page 230. |
| "Report" | Creates and saves a report using the settings defined in "Menu" key > "Save/Recall" key > "Save" tab > "Report". |
| "Start executable" | Starts an external application. Tap "Set executable" to set the application path and parameters.<br>See: Chapter 4.8.3, "External application", on page 110. |

Remote command:
`TRIGger<m>:EVENt:BEEP` on page 1372
`TRIGger<m>:EVENt:PRINt` on page 1372
`TRIGger<m>:EVENt:WFMSave` on page 1373
`TRIGger<m>:EVENt:RUNexec` on page 1373
`TRIGger<m>:OUT:ACTion` on page 1374

**Trigger out signal setup**
Defines the pulse that is provided to the [Trigger Out] connector on the rear panel.

A trigger out pulse can be provided either when a trigger occurs, or when a mask test violation occurs, or when a limit check violation in a measurement occurs.




"Polarity" Sets the polarity of the trigger out pulse, which is the direction of the first pulse edge.

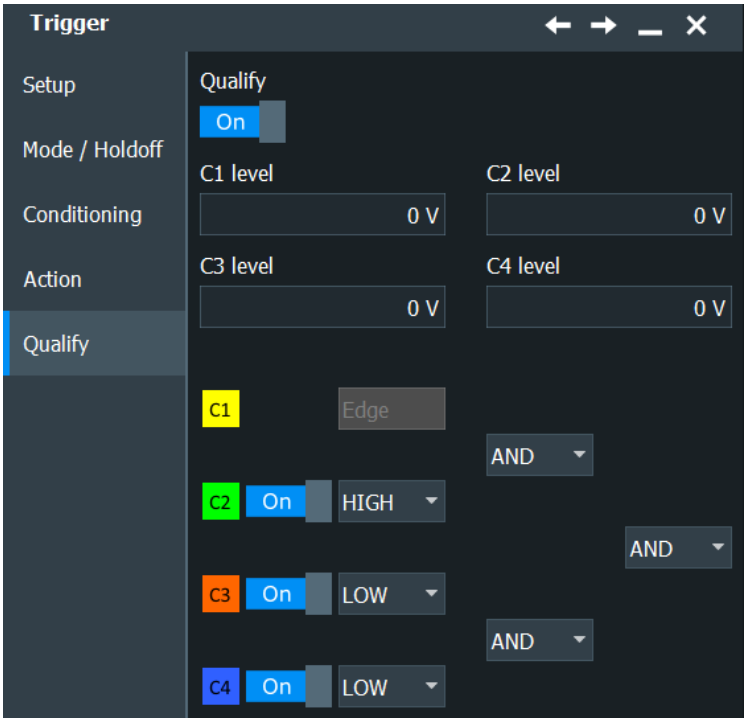
"Delay" Only available if "Trigger out" is set to "On trigger". The setting defines the delay of the first pulse edge to the trigger point. The minimum delay is 800 ns.

"Pulse length" Sets the length of the trigger out pulse.

Remote command:
`TRIGger<m>:OUT:POLarity` on page 1371
`TRIGger<m>:OUT:PLENgth` on page 1371
`TRIGger<m>:OUT:DELay` on page 1372

## 6.7 Qualification

Access: "Menu" > "Trigger" > "Qualify" tab



By qualifying a trigger event, you can logically combine the trigger signal with the state of other analog channel signals.

The instrument triggers if both of the following apply:

- The basic conditions of the trigger event definition are fulfilled.
- The logical conditions of the trigger qualification are true.

Qualification is only available for the A-event.

Qualification is not supported if:

- The trigger source is "Extern".
- One of the following trigger types is selected: slew rate, Setup & Hold, serial pattern, TV, and NFC.

► To enable the qualification settings, select Qualify.

**Example: Trigger on write access of a specific device of a bus system**

In circuits using SPI, several secondary devices use the same lines for reading and writing data, and each secondary device has its own select line. To trigger on write access of specific secondary device, the write line is the trigger source and the select line of the secondary device is set as qualifying condition.

**Qualify**

Enables the settings for trigger qualification that are defined in the "Qualification" tab. Qualification adds additional trigger conditions considering the logic states of other digital channel signals.

The checkmark is only active if at least one qualification channel is selected.

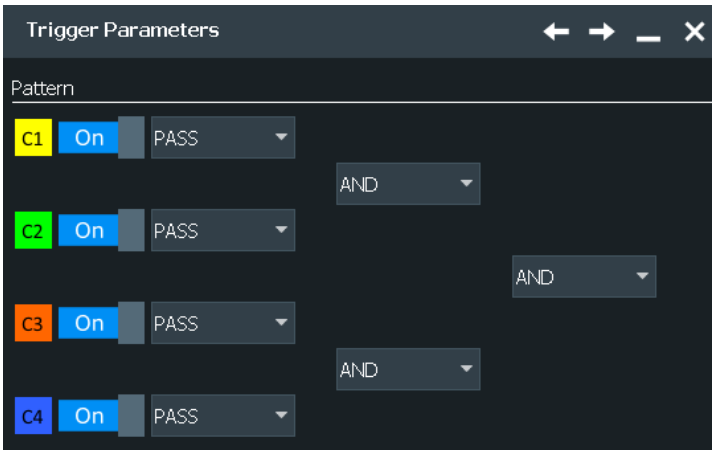
Qualification is available for many trigger types: Edge, glitch, width, runt, window, timeout, and interval.

Qualification is not possible for the R-event.

See also: Chapter 6.7, "Qualification", on page 231.

**Pattern**

The pattern contains the channel selection, and the logical operations structure of hardware-based Boolean logic.



"Channel" Select the channels to be considered. For qualification, you can select all channel signals except for the trigger source. In pattern trigger setup, the trigger source channel is selected by default, and you can select all other channel signals.

"Boolean operator" Defines the logical operation on the digital signal resulting from the comparison with the trigger level.

- "PASS": leaves the input value unchanged
- "NOT": inverts the input value

"Logical operator" Defines the logic combination of two sources. The sources are channel 1/2 and channel 3/4 on the first step, and in the second step the logical combination resulting from the first step.

- "AND": logical AND, conjunctive combination
- "NAND": logical NOT AND
- "OR": logical OR, disjunctive combination
- "NOR": logical NOT OR

Remote command:
`TRIGger<m>:QUALify<n>:A:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:A[:ENABle]` on page 1358
`TRIGger<m>:QUALify<n>:AB:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:ABCD:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:B:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:B[:ENABle]` on page 1358
`TRIGger<m>:QUALify<n>:C:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:C[:ENABle]` on page 1358
`TRIGger<m>:QUALify<n>:CD:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:D:LOGic` on page 1359
`TRIGger<m>:QUALify<n>:D[:ENABle]` on page 1358

**Trigger Levels**
Define the trigger level for each input channel. For state and pattern trigger, the trigger level is a decision threshold: If the signal value is higher than the trigger level, the signal state is high (1 or true for the Boolean logic). Otherwise, the signal state is considered low (0 or false) if the signal value is below the trigger level.

These trigger levels are also used in qualification setup.

You can set the trigger levels for all channels to the same value, see "Use data level for clock level" on page 212.

**Use data level for clock level**
Sets the trigger levels and hysteresis values for all channels to the values of the currently selected trigger source. The function affects only the levels defined for the selected condition. The hysteresis of the external trigger input is an independent value, and it is not affected by level coupling.

In trigger sequences, another coupling of trigger levels is possible: "Couple sequence thresholds" on page 236.

Remote command:
`TRIGger<m>:SCOupling` on page 1323

## 6.8 Sequence

A trigger sequence consists of at least one trigger condition and additional conditions defining when the trigger occurs.

The trigger sequence "A → B → R", for example, consists of two subsequent conditions: A-trigger and B-trigger with optional B-trigger delay and count. In addition, a reset condition R can be configured: timeout or R-trigger condition. A-, B-, and R-triggers are configured in the same way.

The instrument checks all trigger settings for compatibility and disables settings that do not fit the previous settings in the sequence.

After the A-trigger conditions have been met, and an optional delay has passed, the B-trigger with independent conditions is enabled. The instrument waits until one or a specified number of B-trigger conditions occur. If the reset condition is not fulfilled, the latest B-trigger causes the trigger event, and then the sequence starts again. The B-trigger can only cause the trigger event if it occurs after the A-trigger and after the delay time.

If you expect, for example, an irregular B-trigger, you can configure a reset condition to restart the sequence. The reset condition can be a simple timeout, and/or a reset event that is defined in the same way as the A- and B-trigger conditions.

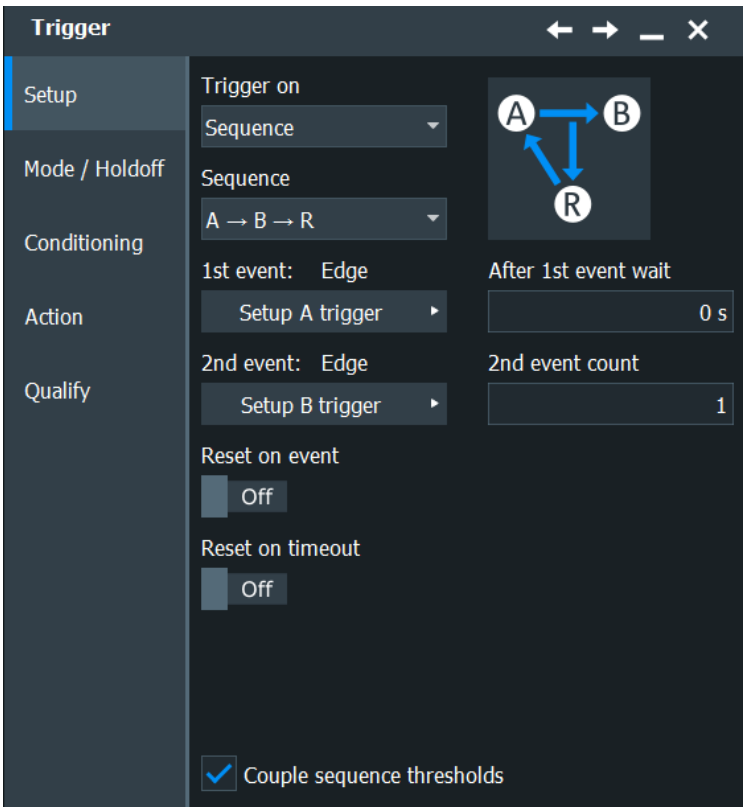
All trigger sequences require that input channels CH1...4 are set as trigger sources for all conditions. All other input sources are disabled. Trigger sequences are not available if one the following trigger types is set as A-trigger:

- Setup & Hold
- TV
- NFC
- CDR

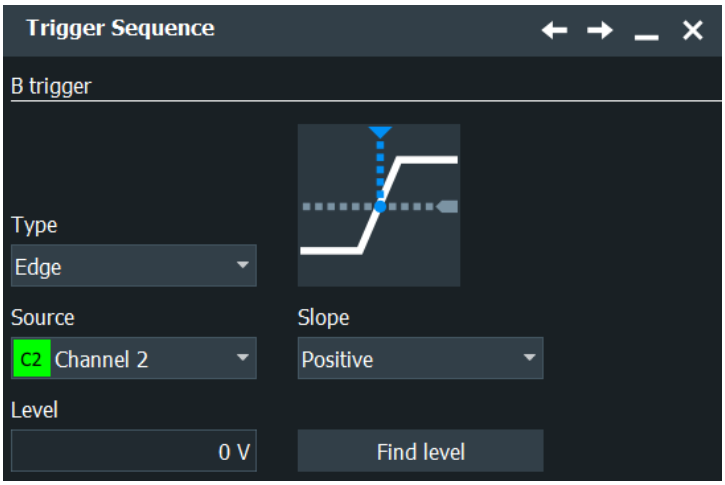
For the zone trigger, more trigger sequences are available, see Chapter 6.9, "Zone trigger", on page 237. Zone trigger requires option R&S RTP-K19.

### 6.8.1 Sequence setup

Access: "Menu">"Trigger" > "Setup" tab > "Trigger on" = "Sequence"




The B-trigger is the second condition of the trigger sequence. You can configure a delay between the A- and B-trigger, and define a number of fulfilled B-trigger conditions to be ignored. The B-trigger condition is configured in the same way as the A-trigger. The instrument disables settings that do not fit the previous settings in the sequence.




The reset condition R can be a timeout or a trigger condition, or a combination of both.

**Sequence**
Selects the type of the sequence.

"A → B → R" Triggers if all conditions of A- and B-events, as well as additional delay and reset timeout or R-event (reset) conditions are fulfilled.

| | |
|---|---|
| "A → Zone" | Triggers if the conditions of the A-event and the zone trigger are fulfilled. A holdoff condition also can be set. |
| "A → B → R → Zone" | Triggers if all conditions of A- and B-events, reset timeout or R-event (reset) conditions are fulfilled for the defined zone. |
| "A OR B" | Triggers if the conditions of A-event or the conditions of the B-event are fulfilled. |
| "A OR B → Zone" | Triggers if the conditions of A-event or the conditions of the B-event are fulfilled for the defined zone. A holdoff condition also can be set. |

Remote command:
`TRIGger<m>:SEQuence:TYPE` on page 1367

**After 1st event wait**
Sets the time that the instrument waits after an A-trigger until it recognizes B-triggers.

Remote command:
`TRIGger<m>:SEQuence:DELay` on page 1368

**2nd event count**
Sets the number of B-trigger conditions to be fulfilled after an A-trigger. The last B-trigger causes the trigger event.

The waiting time for B-triggers can be restricted with a reset condition: timeout or reset event.

Only available in "A → B → R" sequences.

Remote command:
`TRIGger<m>:SEQuence:COUNt` on page 1369

**Reset on event/Reset event**
If enabled, the trigger sequence is restarted by the R-trigger condition if the specified number of B-triggers does not occur before the trigger conditions are fulfilled.

The R-trigger condition is configured in the same way as the A-trigger. The instrument disables settings that do not fit the previous settings in the sequence.

Remote command:
`TRIGger<m>:SEQuence:RESet:EVENt` on page 1369

**Reset on timeout/Reset timeout**
If timeout is enabled, the instrument waits for the "Timeout" time for the specified number of B-triggers. If no trigger occurs during that time, the sequence is restarted with the A-trigger.

Remote command:
`TRIGger<m>:SEQuence:RESet:TIMeout[:ENABle]` on page 1369
`TRIGger<m>:SEQuence:RESet:TIMeout:TIME` on page 1370

**Couple sequence thresholds**
Sets the trigger levels to the values of the latest configured trigger condition. Each channel has its own trigger level.

**Example:**

You have configured the A-trigger at last, and the trigger level for C1 is 70 mV. Enabling the coupling sets the trigger levels for C1 in the B- and R trigger also to 70 mV. If the B-trigger and/or R-trigger use another source as the A-trigger, the level remains unchanged.

Remote command:
`TRIGger<m>:ECOupling` on page 1368

## 6.9 Zone trigger

The zone trigger function requires option R&S RTP-K19.

The zone trigger triggers on the intersection or non-intersection of the signal and one or more zones or masks. The zone can be applied to any active input signal, math waveform including FFT, and XY-waveform.

You can use the zone trigger, for example, to solve the following tasks:

- Trigger if a peak in the spectrum occurs: define a zone in the FFT diagram to filter amplitude peaks. In the same way, you can filter harmonics.
- Separate rising and falling edges: define a zone around the base or top of the data signal.
- Separate read/write cycles: define a zone in the eye diagram.
- Identify a tube violation of signals with an infrequent non-monotonic edge.
- Filter events in the history, after acquisition has finished

To document the trigger events, use the actions on trigger. For example, create a report when the instrument triggers, or save the waveform.

Zone trigger is not available for serial protocol triggers.

### Trigger sequence with zone trigger

You can combine the zone trigger with common trigger conditions and use one of the following trigger sequences:

- "A → Zone"
- "A → B → R → Zone"
- "(A OR B) → ZoneTrigger"

In a sequence, the zone trigger acts like a filter on previous acquisitions. Previous conditions and then the zone condition must be fulfilled. For example, in an "A → Zone" sequence, the trigger marker is set to the A event, and a corresponding trigger-out signal is sent when the zone condition is fulfilled.

### 6.9.1 About trigger zones

A zone is a mask without result box. You can adjust the shape of the zone in the same way as mask segments, graphically by dragging the corner points, or numerically in the "Masks" > "Mask Definition" dialog box. You can also use existing masks in the zone trigger. The zone trigger and usual mask tests run in parallel.

If you switch off a mask test that is used by the zone trigger, the mask is removed from the zone trigger expression automatically.

All zones and masks that are included in the zone trigger expression are indicated with yellow color.

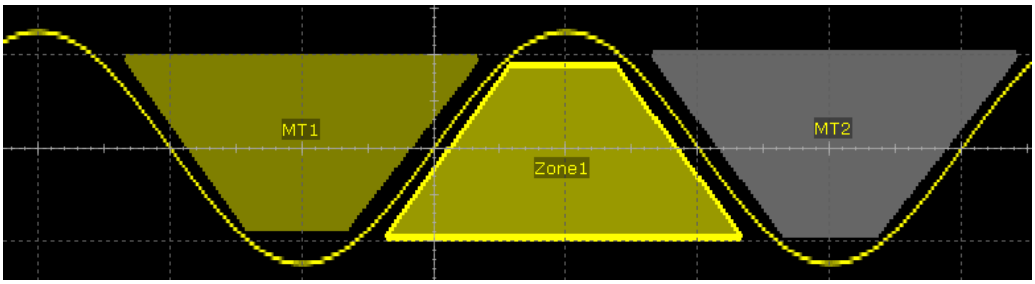



***Figure 6-4: Indication of trigger zones***

MT1 = Mask is included in zone trigger. Trigger if signal hits the mask (must intersect).
MT2 = Mask is not included in zone trigger.
Zone1 = Zone is included in zone trigger. Trigger if signal does not hit the zone (must not intersect).

If average or envelope acquisition is enabled, only triggered waveforms are used for envelope and average calculation on channel and math waveforms.

### 6.9.2 Creating trigger zones

You can define the trigger zones on the display, or use existing mask definitions as trigger zones.

**Define trigger zones**

1. Tap the "Zone trigger" icon on the toolbar.

2. In the signal bar, select if the signal must violate the zone to cause a trigger (must intersect), or vice versa (must not intersect).
3. Tap the corner points of the zone on the screen.
4. If all points are set, tap "Finish zone" in the signal bar.

   "Zone1" is shown in the expression field.

5. If you want to define another zone, select the logic combination of the zones AND (both zones) or OR (either zone).
6. To create the second zone, repeat steps 3 and 4.

7. To define more zones, repeat steps 5 and 6.

**Use existing masks for zone trigger**

Prerequisites: At least one mask is created and visible on the display.

1. Tap the "Zone trigger" icon on the toolbar.

2. In the signal bar, select if the signal must violate the zone to cause a trigger ("Must Intersect"), or vice versa ("Must Not Intersect").
3. Tap the mask on the display.
4. Tap "Use selected mask" in the signal bar.

   "MT1" is shown in the expression field.

5. If you want to add another mask test or zone, select the logic combination of the zones AND (both zones) or OR (either zone).
6. To add another mask, tap the mask on the display.
   Tap "Use selected mask" in the signal bar.
7. To add another zone, tap the corner points of the zone on the screen.
   If all points are set, tap "Finish zone" in the signal bar.

If a zone trigger is already defined, you can also add a new mask to the zone trigger when creating the mask.

**Intersect**
Defines if the signal must intersect the zone to allow the instrument to trigger, or if it must not intersect the zone.

**Combine**
Sets the logic combination of two zones.

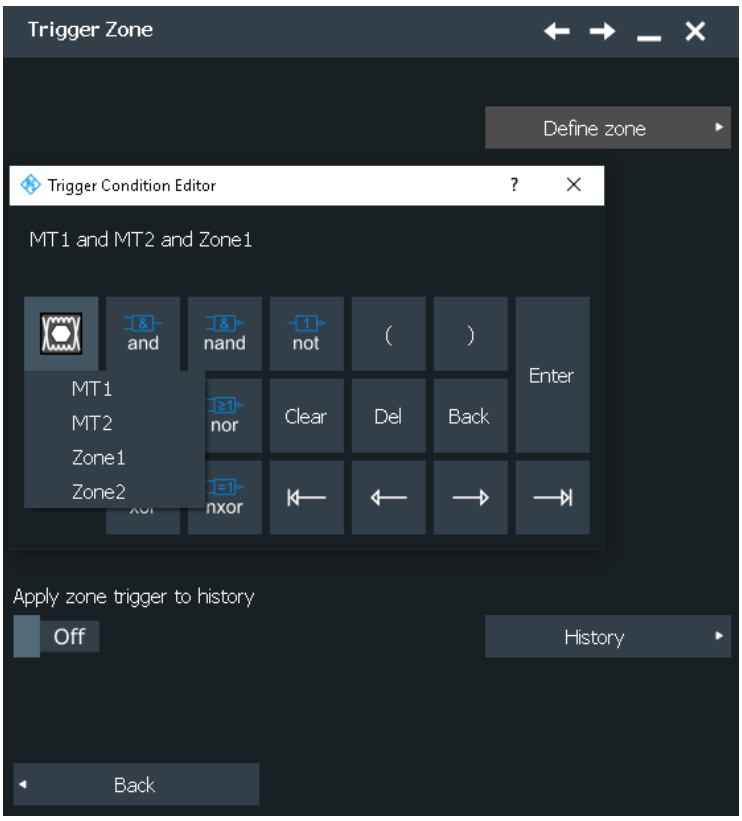
**Use selected mask**
Includes the selected zone or mask in the zone expression.

**Clear Expr**
Deletes the zone trigger expression. The zones are not deleted, they remain as usual masks, and the result boxes of the mask test appear.

**Delete all**
Deletes the zone trigger expression and the zones.

### 6.9.3 Advanced setup and analysis with zone trigger

Access: overlay menu for zone trigger > "Advanced Setup", or "Trigger" menu > "Setup" > select sequence with zone trigger > "Setup Z trigger" dialog




**Logical trigger condition**
If all required mask tests and zones are defined, you can type the logical expression directly, or use the trigger condition editor. All logical combinations are available in the editor. To express the "Must Not Intersect" condition, use the logic NOT.

Remote command:
`TRIGger<m>:ZONE:EXPRession[:DEFine]` on page 1374

**Apply zone trigger to history**
Applies the zone trigger condition to the acquisitions in the history memory.

The history saves all acquisitions that fulfill the trigger condition that is set during acquisition (zone trigger condition, or another trigger condition). If "Apply zone trigger to history" is disabled, the history "Play" reads and displays all saved acquisitions from the memory.

If the setting is enabled, the zone trigger is applied to the history replay. "Play" displays only acquisitions that fulfill the zone trigger condition.

Using "Apply zone trigger to history", you can:
- Acquire waveforms with high speed and filter them afterwards in the history.
- Change the zone trigger condition after acquisition.

Remote command:
`SWTRigger:HISTory` on page 1374

# 6.10 High speed serial pattern trigger (option R&S RTP-K140/K141)

Option R&S RTP-K140/K141 provides additional real-time trigger capabilities on high speed communication protocols that use NRZ (non-return-to-zero) line encoding and bit transmission rates starting at 21 kbps and up to 16 Gbit/s.

The option provides:

- Support transmission rates starting at 21 kbps and up to 8 Gbit/s (R&S RTP-K140) / 16 Gbit/s (R&S RTP-K141)
- Triggering on single/dual bit patterns and complex words.
- Triggering on PRBS errors.
- Trigger on protocols that use the line codes 8b/10b and 128b/132b.
- Trigger on descrambled 8b/10b communications
- Real-time hardware-based clock data recovery (CDR) up to 16 Gbit/s.

## 6.10.1 About the serial pattern trigger

A trigger occurs if the complete set of trigger conditions is fulfilled. In the serial pattern trigger dialog, you can set the basic settings and pattern type specific settings.

**Basic trigger settings**

In the basic settings, you define if the clock data recovery is used for extracting a clock signal from the signal (enabled CDR), or if a dedicated clock signal is provided on another channel (disabled CDR).

Digital high-speed serial data streams use typically an embedded clock. No additional accompanying clock signal such as for parallel buses (e.g. DDR memory interface DDR3/ DDR4) is required.

With enabled CDR, the timing information is embedded in the data. The signal is extracted, resulting in a clock signal that can be further processed by the trigger system. The internally generated CDR clock signal is used to sample the original data signal and decode the sequence of transmitted bits.

The following example displays the trigger position for a single bit pattern using the hardware CDR setup:

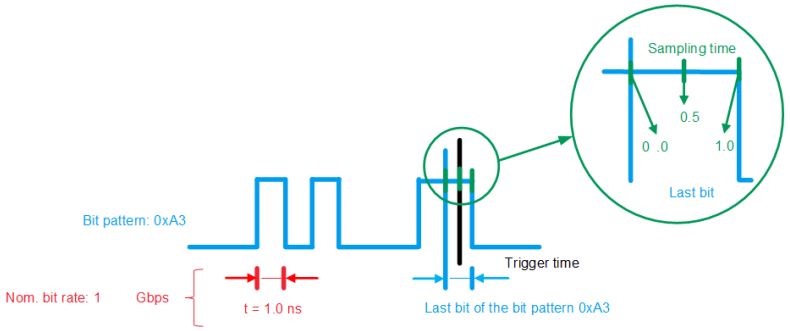



*Figure 6-5: Example for a triggering parameters of a recovered clock signal*

You can define the "Nom. bit rate" of your signal, or use the predefined value of one of the supported "Serial standard" and "Sampling time".

### Maximum nominal bit rate

The available maximum nominal bit rate depends on the configuration settings of your instrument. The following maximum nominal bit rate can be achieved:

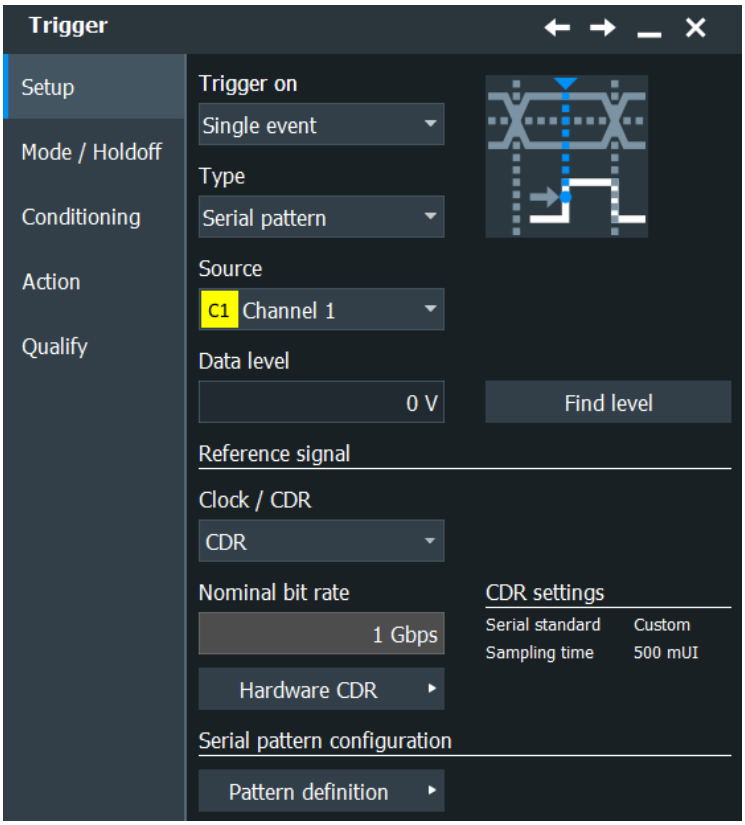
| Instruments bandwidth | Number of active channels | Maximum acquisition mode | Maximum nominal bit rate | |
|---|---|---|---|---|
| | | | R&S RTP-K140 | R&S RTP-K141 |
| up to 8 GHz | 1 channel | 40 GSa/s | 8 Gbit/s | 8 Gbit/s |
| | 2 channels:<br>CH1 and CH3 or<br>CH1 and CH4 or<br>CH2 and CH3 or<br>CH2 and CH4 | If serial number > 103000:<br>40 GSa/s<br>If serial number < 103000:<br>20 GSa/s | | |
| | 2 channels:<br>CH1 and CH2 or<br>CH3 and CH4 | 20 GSa/s | | |
| | 3 or 4 channels | 20 GSa/s | | |
| 13 GHz, 16 GHz | 1 channel | 40 GSa/s | 8 Gbit/s | 16 Gbit/s |
| | 2 channels:<br>CH1 and CH3 or<br>CH1 and CH4 or<br>CH2 and CH3 or<br>CH2 and CH4 | 40 GSa/s | | |
| | 2 channels:<br>CH1 and CH2 or<br>CH3 and CH4 | 20 GSa/s | | 8 Gbit/s |
| | 3 or 4 channels | 20 GSa/s | | |

**Pattern settings**

The pattern settings depend on the selected "Pattern type", see Chapter 6.10.2, "High speed serial pattern settings", on page 243.

## 6.10.2 High speed serial pattern settings

Access: "Menu" > "Trigger" > "Setup" tab > "Type = Serial Pattern"




The available settings depend on the selected "Pattern mode" in the "Pattern definition" dialog.

### 6.10.2.1 General settings for serial pattern trigger

In the basic serial pattern settings, you can define the clock signal in relation to which you want to trigger. You can use the recovered clock signal for the serial pattern trigger ("CDR" ) instead of an explicit clock signal.

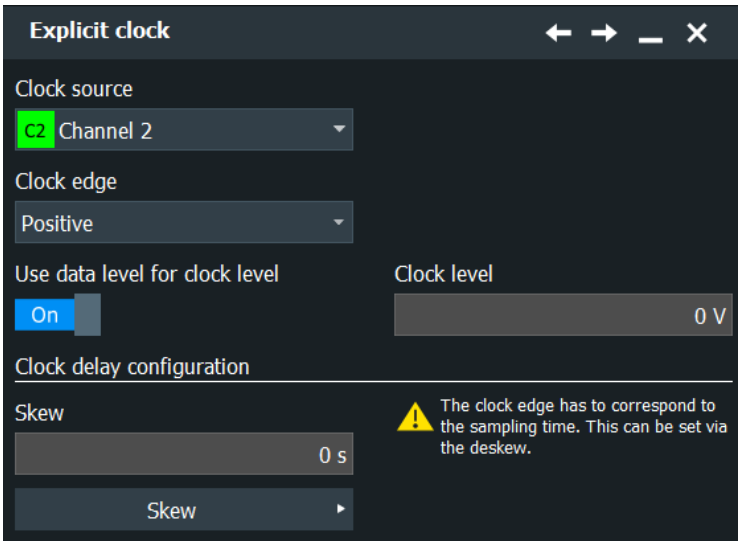
**Clock / CDR**
Selects the type of waveform that is used to obtain the reference signal: .

"CDR" Uses software clock data recovery.

"Clock" Uses an explicit clock signal. The signal must be connected to an additional channel.

Remote command:
TRIGger<m>:SPATtern:REFSource on page 1331

**Explicit clock**
Opens a dialog to define the explicit clock, for "Clock / CDR" = "Clock".




**Clock source ← Explicit clock**
Selects the input channel of the clock signal.

Remote command:
TRIGger<m>:SETHold:CSOurce[:VALue] on page 1322

**Clock edge ← Explicit clock**
Together with the clock level, the clock edge sets the point in time when the state of the data signal is checked:

"Positive" Rising edge, a positive voltage change.

"Negative" Falling edge, a negative voltage change.

"Both" Both the rising and the falling edge.

Remote command:
TRIGger<m>:SPATtern:CSOurce:EDGE on page 1330

**Use data level for clock level ← Explicit clock**
Sets the trigger levels and hysteresis values for all channels to the values of the currently selected trigger source. The function affects only the levels defined for the selected condition. The hysteresis of the external trigger input is an independent value, and it is not affected by level coupling.

In trigger sequences, another coupling of trigger levels is possible: "Couple sequence thresholds" on page 236.

Remote command:
TRIGger<m>:SCOupling on page 1323

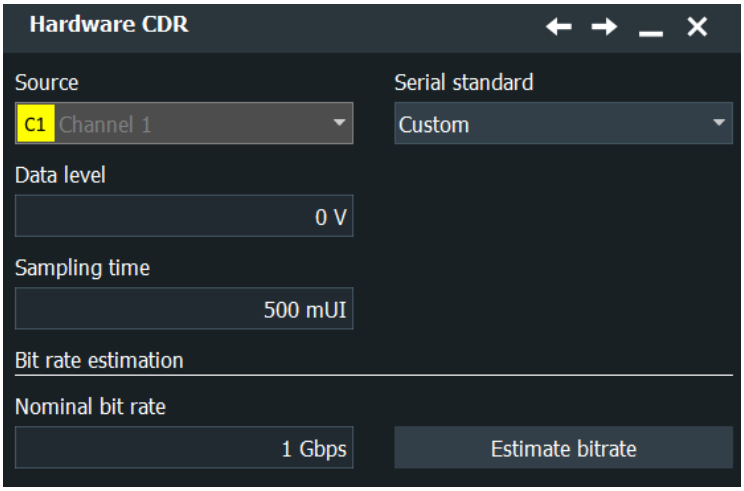
**Clock level ← Explicit clock**
Sets the voltage level for the clock signal.

Remote command:
`TRIGger<m>:SPATtern:CSOurce:LEVel` on page 1330

**Skew**
Sets a delay value, which is known from the circuit specifics but cannot be compensated by the instrument automatically. It affects only the selected input channel.

You can define additional skew parameters in the "Skew" dialog, see Chapter 5.3.1, "Skew", on page 125.

**Hardware CDR**
Opens a dialog to define the explicit clock, for "Clock / CDR" = "CDR".




**Serial standard ← Hardware CDR**
For some serial standards, the bit rate is known to the instrument. Select the standard, and the pattern type is set automatically, no estimation is required. The serial standards are available only if the instrument is able to support its maximum nominal bit rate, see "Maximum nominal bit rate" on page 242.

The following predefined standards are currently available:

| Serial standard | Nom. bit rate | Pattern type |
|---|---|---|
| USB 2.0 | 480 Mbit/s | 8b/10b |
| USB 3.1 Gen 1 | 5 Gbit/s | 8b/10b |
| USB 3.1 Gen 2 | 10 Gbit/s | 128b/132b |
| DisplayPort 1.1 | 1.296 Gbit/s | 8b/10b |
| DisplayPort 1.2 | 4.32 Gbit/s | 8b/10b |
| Fibre Channel 1x | 1 Gbit/s | 8b/10b |
| Fibre Channel 2x | 2 Gbit/s | 8b/10b |
| Fibre Channel 4x | 4 Gbit/s | 8b/10b |

| Serial standard | Nom. bit rate | Pattern type |
|---|---|---|
| Fibre Channel 8x | 8 Gbit/s | 8b/10b |
| HDMI 1.2 | 1.32 Gbit/s | 8b/10b |
| HDMI 1.4 | 3.4 Gbit/s | 8b/10b |
| HDMI 2.0 | 4.8 Gbit/s | 8b/10b |
| PCIe Gen 1 | 2.5 Gbit/s | 8b/10b |
| PCIe Gen 2 | 5 Gbit/s | 8b/10b |
| SATA Gen 1 | 1.5 Gbit/s | 8b/10b |
| SATA Gen 2 | 3 Gbit/s | 8b/10b |
| SATA Gen 3 | 6 Gbit/s | 8b/10b |
| SAS Gen 1 | 3 Gbit/s | 8b/10b |
| SAS Gen 2 | 6 Gbit/s | 8b/10b |
| SAS Gen 3 | 12 Gbit/s | 8b/10b |
| Thunderbolt 10G | 10.3125 Gbit/s | Single bit pattern |
| XAUI | 3.75 Gbit/s | 8b/10b |

Remote command:
`CDR:HARDware:SERStandard` on page 1331

**Data level ← Hardware CDR**
Sets the voltage level for the data signal.

If the signal value is higher than the data level, the state is 1. Below the level, the signal state is 0.

Remote command:
`TRIGger<m>:LEVel<n>[:VALue]` on page 1306

**Sampling time ← Hardware CDR**
Defines the position of the last bit of the bit pattern to be triggered on, relative to the trigger time. The unit interval offset is a number between 0 and 1:

- Value 0: start/"left edge" of the last bit
- Value 0.5: middle of the last bit
- Value 1: end/"right edge" of the last bit

Remote command:
`CDR:HARDware:UIOFfset` on page 1333

**Nominal bit rate ← Hardware CDR**
Sets the frequency of the feed forward CDR. It corresponds to the data rate of the data stream from which the clock is to be recovered.

Remote command:
`CDR:HARDware:BITRate` on page 1332

**Estimate bit rate ← Hardware CDR**
Estimates the bite rate once, for the current acquisition.

Remote command:
`CDR:HARDware:BREStimate` on page 1332

**Pattern definition**
Opens a dialog to define the pattern.

**Pattern mode ← Pattern definition**
Selects the pattern type for the serial pattern trigger.

The following pattern types are available:

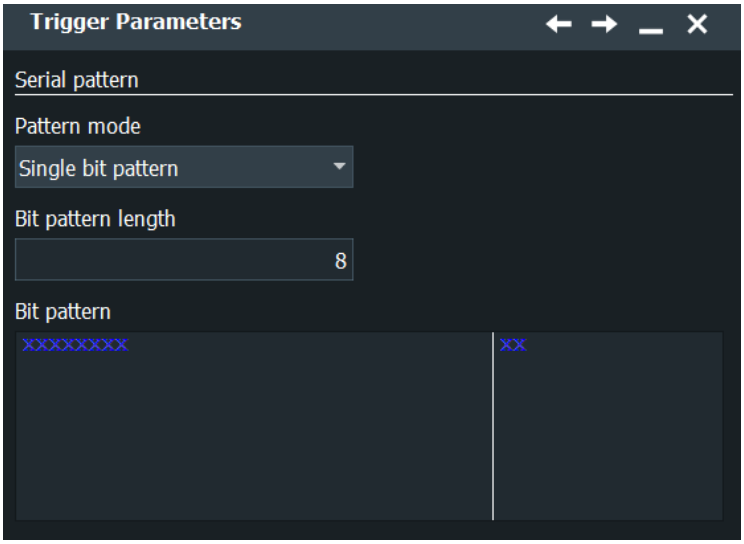
- Single/dual bit pattern
- Complex word pattern
- PRBS error
- 8b/10b
- 128b/132b and 128b/130b

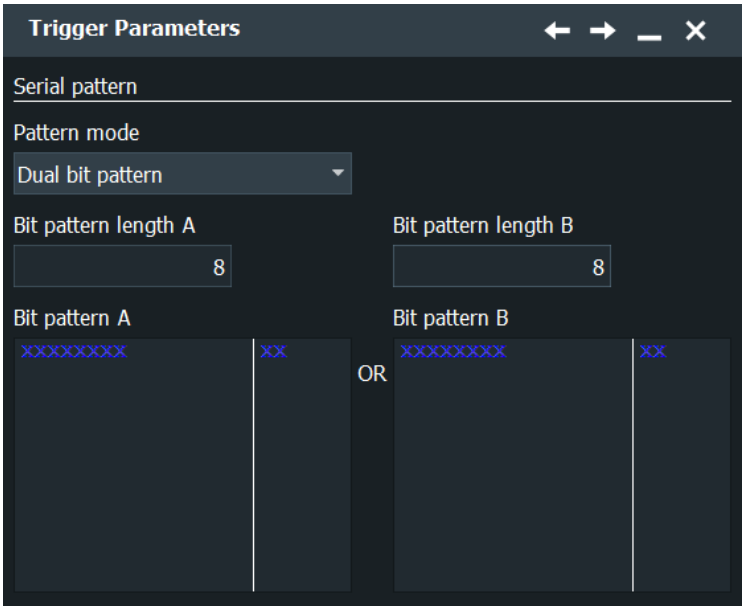
Remote command:
`TRIGger<m>:SERPattern:PTMDe` on page 1328

### 6.10.2.2 Single/dual bit pattern

For the "Single Bit Pattern", you can define a single bit pattern of up to 160 bits.

The "Dual bit pattern" is a logical OR operation of two defined bit patterns that you can trigger on.






**Bit pattern length, Bit pattern length A, Bit pattern length B**
Sets the length of the bit pattern. You can select a value of up to 160 bits.

Remote command:
`TRIGger<m>:SERPattern:BITPattern:PTALength` on page 1334
`TRIGger<m>:SERPattern:BITPattern:PTBLength` on page 1334

**Bit pattern, Bit pattern A, Bit pattern B**
Defines the bit pattern in binary or hex format to.

The bit pattern editor helps you to enter the pattern, see Chapter 13.1.7, "Bit pattern editor", on page 524.

Remote command:
`TRIGger<m>:SERPattern:BITPattern:APATtern` on page 1333
`TRIGger<m>:SERPattern:BITPattern:BPATtern` on page 1333

### 6.10.2.3 Complex word pattern

Within the pattern type "Complex word pattern", a total of four different pattern conditions can be triggered on within a data stream. First, aligning to a frame id in form of bit pattern (or timeout) has to be done.

**Example:**

Figure 6-6 shows a sample configuration of a complex word pattern.

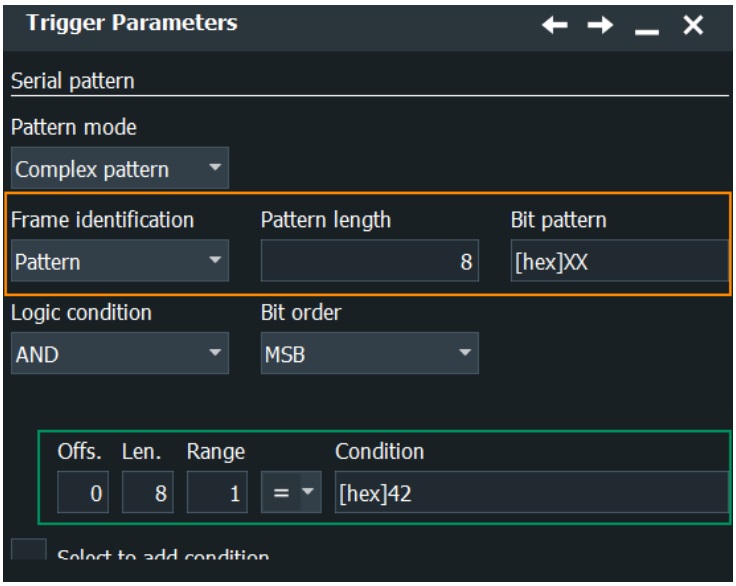



*Figure 6-6: Example for complex word serial trigger pattern settings*

The number of bits to search for a pattern condition is the multiplication of the parameters "Pattern length" and "Search range". The starting point for the search of the pattern condition is the "Bit offset", which is the number of arbitrary bits between the first bit after the frame id and the first bit of the first pattern condition to be triggered on.

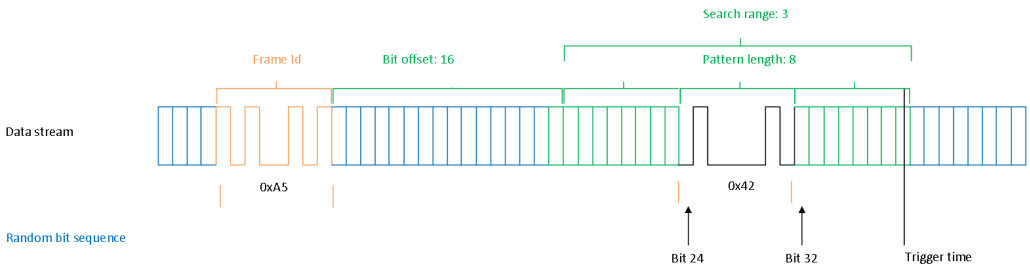



*Figure 6-7: Example data stream of a complex word serial pattern trigger*

The pattern conditions are searched word by word. Hence, there are only three possible start bits – bit 16, bit 24 and bit 32 – in the data stream for which the trigger condition (equal 0x42) is fulfilled. The trigger condition is not satisfied if the bit pattern appears from bit 20 to bit 27.

**Frame Id**
Defines the condition for alignment before triggering on a word pattern.

**Type ← Frame Id**
Selects if the alignment is done on a pattern value or on a defined timeout.

Remote command:
`TRIGger<m>:SERPattern:COMPlex:SOFState` on page 1336

**Length ← Frame Id**
Sets the length of the pattern for the trigger alignment condition.

Remote command:
TRIGger<m>:SERPattern:COMPlex:SOFLength on page 1336

**Bit Pattern ← Frame Id**
Sets the bit pattern value for the trigger alignment condition.

Remote command:
TRIGger<m>:SERPattern:COMPlex:SOFPattern on page 1337

**Timeout ← Frame Id**
Sets the timeout, a duration that defines an idle mark.

Remote command:
TRIGger<m>:SERPattern:COMPlex:SOFTimeout on page 1336

**Idle range**
Selects if the timeout should be within the range or out of range.

Remote command:
TRIGger<m>:SERPattern:COMPlex:IDLRange on page 1335

**Logic**
Selects the logic for the condition, if more than one pattern conditions are defined.

"OR" At least one of the defined pattern conditions have to be met.

"AND" All the defined pattern conditions must be met.

Remote command:
TRIGger<m>:SERPattern:COMPlex:CLOGic on page 1335

**Bit order**
Select most significant bit (MSB) or least significant bit (LSB) to define the data bit order.

Remote command:
TRIGger<m>:SERPattern:COMPlex:BORDer on page 1335

**Bit offset**
Sets the number of arbitrary bits between the first bit after the frame id and the first bit of the first pattern condition to be triggered on.

Remote command:
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:BOFFset on page 1337

**Pattern length**
Sets the word pattern length in bits. The maximum bit length of all selected pattern conditions is 160 bits, with arbitrary bits (e.g., due to bit offsets) not counted.

Remote command:
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:PATLength on page 1338

**Search range**
Sets the search range of the respective pattern condition to be triggered.

Remote command:
`TRIGger<m>:SERPattern:COMPlex:PATTern<n>:SRNGe` on page 1339

**Pattern condition**
Selects the operator for the pattern condition: "Equal", "Not equal", or a range definition.

Remote command:
`TRIGger<m>:SERPattern:COMPlex:PATTern<n>:PCONdition` on page 1338

**Pattern**
Defines the bit pattern in binary or hex format.

The bit pattern editor helps you to enter the pattern, see Chapter 13.1.7, "Bit pattern editor", on page 524.

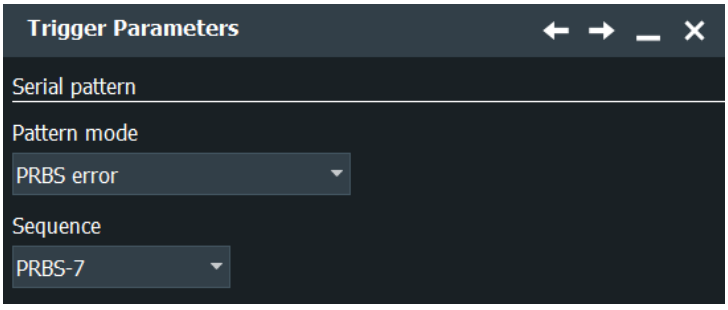
Remote command:
`TRIGger<m>:SERPattern:COMPlex:PATTern<n>:FRPTtern` on page 1337
`TRIGger<m>:SERPattern:COMPlex:PATTern<n>:TOPTtern` on page 1339

**Select to add condition**
If enabled, you can define a new pattern condition. You can define up to 4 conditions.

Remote command:
`TRIGger<m>:SERPattern:COMPlex:PATTern<n>:ENABle` on page 1337

### 6.10.2.4 PRBS error

PRBS sequences are generated by linear feedback shift registers (LFSR) implementing a sequence generating polynomial. LFSR register with K bits generates a PRBS-K sequence of maximum length $2^K$-1 bits. Every group of K consecutive bits appears only once within the sequence and can therefore be used as an alignment mark for synchronization.




**PRBS Sequence**
Selects a predefined pseudo random binary sequence. Available are the following sequences:

- PRBS-7
- PRBS-9
- PRBS-11

- PRBS-15
- PRBS-16
- PRBS-17
- PRBS-20
- PRBS-23
- PRBS-29
- PRBS-31

Remote command:
TRIGger<m>:SERPattern:PRBS:PSEQuence on page 1340

**PRBS Bit error rate**
Sets the bit error rate for the PRBS sequence.

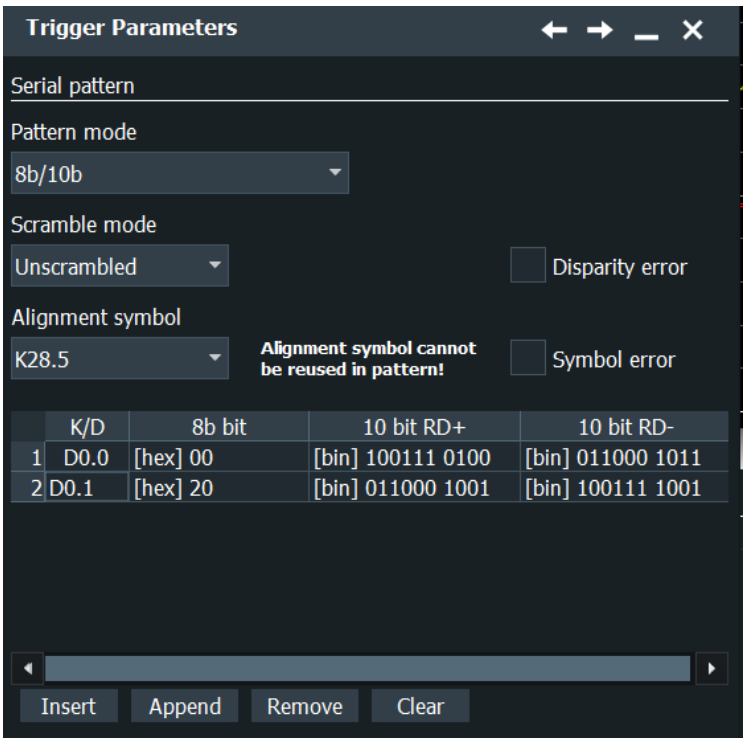
Remote command:
TRIGger<m>:SERPattern:PRBS:ERATe on page 1339

**Nominal bit rate**
Sets the nominal bit rate for the PRBS sequence. It corresponds to the data rate of the data stream from which the clock is to be recovered.

Remote command:
TRIGger<m>:SERPattern:PRBS:NBRTe on page 1340

### 6.10.2.5 8b/10b

You can define 8b/10b pattern that is used in technologies like PCIe Gen1/2 and USB3.1 Gen 1.



**Alignment**
Selects one of the comma control symbols (K28.1, K28.5, K28.7) used for synchronization.

Remote command:
TRIGger<m>:SERPattern:EBTB:ALIGnment on page 1340

**Scramble**
Selects if the data is scrambled or unscrambled.

Remote command:
TRIGger<m>:SERPattern:EBTB:SCRMble on page 1344

**8b/10b table**
In this table, you can define up to 16 symbols.

| | |
|---|---|
| "K/D symbol" | Selects the data characters (Dx.y) or the control characters (Kx.y) |
| "8b bit" | Sets the 8b bit value. |
| "10b bit RD+" | Displays the value for the positive running disparity. |
| "10b bit RD-" | Displays the value for the negative running disparity. |
| "Insert" | Inserts an entry before the currently selected row. |
| "Append" | Inserts an entry at the end of the table. |
| "Remove" | Removes an entry. |
| "Clear" | Clears all entries. |

Remote command:
TRIGger<m>:SERPattern:EBTB:SYMBols<n>:KDSYmbol on page 1343
TRIGger<m>:SERPattern:EBTB:SYMBols<n>:EBIT on page 1343
TRIGger<m>:SERPattern:EBTB:SYMBols<n>:DMINus? on page 1342
TRIGger<m>:SERPattern:EBTB:SYMBols<n>:DPLus? on page 1343
TRIGger<m>:SERPattern:EBTB:APPend on page 1341
TRIGger<m>:SERPattern:EBTB:COUNt? on page 1341
TRIGger<m>:SERPattern:EBTB:CLEar on page 1341

**Disparity error**
Enables triggering on violation of the disparity condition.

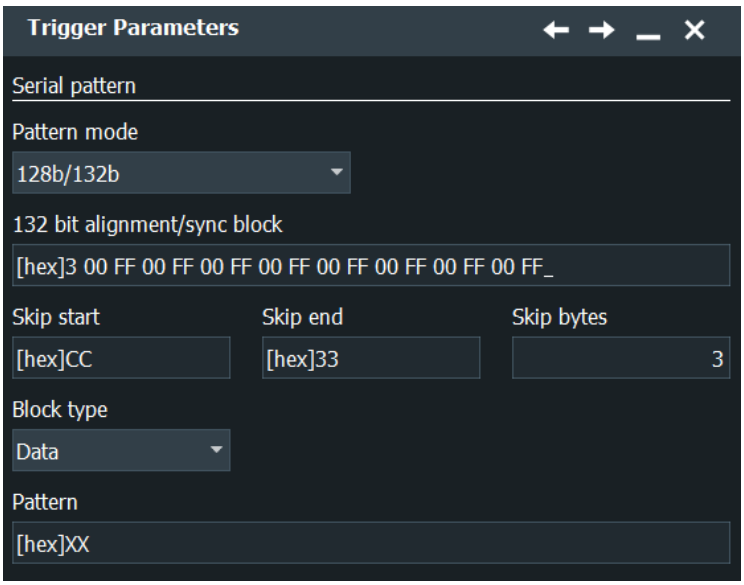
Remote command:
TRIGger<m>:SERPattern:EBTB:ERRor<n>:DISParity on page 1342

**Symbol error**
Enables triggering on symbol errors. A symbol error occurs, when the decoded symbol cannot be found in the table of 8B/10B data symbol codes.

Remote command:
TRIGger<m>:SERPattern:EBTB:ERRor<n>:SYMBol on page 1342

#### 6.10.2.6 128b/132b and 128b/130b

The 128b/132b code is, for example, used for high speed transmission rates of 10 Gbit/s USB 3.1 Gen2.




**Alignment / Sync**
Sets the alignment pattern.

Remote command:
`TRIGger<m>:SERPattern:I32B:APATtern` on page 1344
`TRIGger<m>:SERPattern:I30B:APATtern` on page 1346

**Skip start**
Sets the skip start value. It is used to compensate for different bit rates between two communicating ports.

Remote command:
`TRIGger<m>:SERPattern:I32B:SSTR` on page 1345
`TRIGger<m>:SERPattern:I30B:SSTR` on page 1347

**Skip end**
Sets the skip end symbol. It marks the boundary between skip symbols and the rest of the skip-ordered sets.

Remote command:
`TRIGger<m>:SERPattern:I32B:SSTP` on page 1345
`TRIGger<m>:SERPattern:I30B:SSTP` on page 1347

**Skip bytes**
Sets the skip bytes symbols send after a skip end symbol to describe the LSFR state.

Remote command:
`TRIGger<m>:SERPattern:I32B:SKAFter` on page 1345
`TRIGger<m>:SERPattern:I30B:SKAFter` on page 1346

**Block type**
Selects the type of pattern, control block or the data block.

Remote command:
TRIGger<m>:SERPattern:I32B:SYMType on page 1345
TRIGger<m>:SERPattern:I30B:SYMType on page 1347

**Pattern**
Defines the control block or data block pattern in binary or hex format.

The bit pattern editor helps you to enter the pattern, see Chapter 13.1.7, "Bit pattern editor", on page 524.

Remote command:
TRIGger<m>:SERPattern:I32B:PATTern on page 1344
TRIGger<m>:SERPattern:I30B:PATTern on page 1346

### 6.10.3 Configuring the serial pattern trigger

For details on the high speed serial pattern trigger settings, see Chapter 6.10.2, "High speed serial pattern settings", on page 243.

**Setting up basic serial pattern settings**

1. Select "Menu" > "Trigger" > "Setup".
2. Select the channel for the "Source".
3. Select "Type" > "Serial pattern".
4. Set the clock signal:
   a) Select the "Clock / CDR" respective to your reference clock signal.
   b) For "Clock / CDR" = "Clock" define the hardware clock in the "Hardware CDR" dialog.
   c) For "Clock / CDR" = "CDR" define the explicit clock in the "Explicit clock" dialog.
5. Open the "Pattern definition" dialog.
6. Select the "Pattern type".
7. Set the pattern type specific settings:
   - For dual bit serial pattern trigger, see "Setting up a dual bit serial pattern trigger" on page 255.
   - For 8B/10B serial pattern trigger, see "Setting up a 8B/10B serial pattern trigger" on page 256.

**Setting up a dual bit serial pattern trigger**

1. Set the basic serial pattern settings. See "Setting up basic serial pattern settings" on page 255.
2. Open the "Pattern definition" dialog.
3. Select "Pattern type" > "Dual bit pattern".

4. Set the "Bit pattern length A/B".
5. Set the "Bit pattern A/B".

**Setting up a 8B/10B serial pattern trigger**

1. Select "Menu" > "Trigger" > "Setup".
2. Select "Type" > "Serial pattern".
3. Set the "Clock / CDR" to "CDR".
4. Open the "Pattern definition" dialog.
5. Select "Pattern type" > "8b/10b".
6. Set the "Alignment".
7. Press "Append" to add a new K/D symbol.
8. Set the value of the "K/D symbol".

## 6.11 External trigger input

Except for using analog or digital input channels as trigger source, you can also use external signals as trigger source. The external signal is connected to the external trigger input, which is equipped with the Rohde & Schwarz probe interface. Thus, you can use all Rohde & Schwarz probes to connect the external trigger signal. The only trigger type to trigger on external signals is the edge trigger.

1. Connect the external trigger signal to the external trigger input.
2. Set up the trigger:
   a) Select the "Trigger" > "Setup" tab.
   b) Select the source: "Extern"
   c) Adjust the trigger settings.
   See Chapter 6.11.1, "External trigger setup", on page 257.
3. Set up the probe: In the "Trigger" > "Setup" tab, tap "Setup external probe".
   See: Chapter 6.11.2, "Probe setup for external trigger input", on page 258.

If the trigger source is a channel input, the trigger system uses the digitized signal. The trigger system of the instrument is a separate system, thus the signal processing by enhancement, decimation and arithmetic has no impact on the trigger signal. Most of the R&S RTP trigger types use the digitized trigger signal.

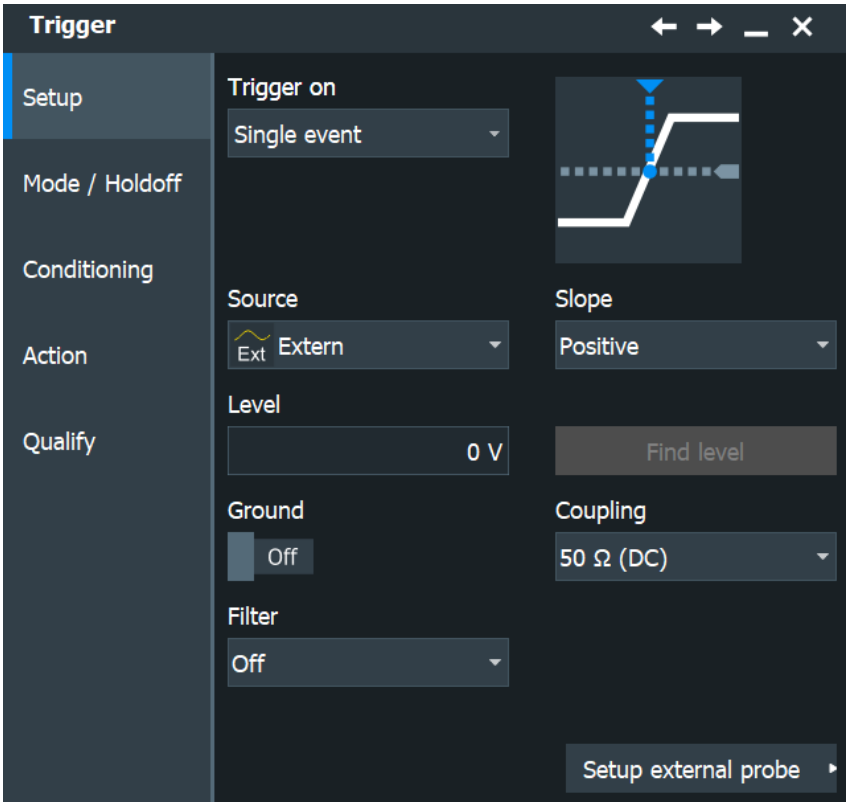
If the trigger source is the external trigger input, the trigger comparator uses the analog input signal. For the external trigger signal, only the edge trigger of the A-trigger is available. Trigger sequence is not supported.

Qualification of the external trigger signal is not available.

### 6.11.1 External trigger setup

Access: "Menu" > "Trigger" > "Setup" tab > "Source = Extern"

External trigger signals, which are connected to the external trigger input, can be triggered with an edge trigger. The "Find level" function is not available for external trigger signals.




**Ground**
If the selected trigger source is the external trigger input, you can connect the trigger input to the ground.

Remote command:
`TRIGger<m>:ANEDge:GND` on page 1376

**Coupling**
The external trigger input is a direct connection with 50 Ω termination, which passes both DC and AC components of the trigger signal.

Remote command:
`TRIGger<m>:ANEDge:COUPling` on page 1375

**Filter**
If the selected trigger source is "Extern" (external trigger input), you can directly select a filter to reject high or low frequencies.

For all other trigger sources, you can add a digital filter using the "Digital Filter" setup.

"Off" The trigger signal is not filtered.

"LF reject" Frequencies below 50 kHz are rejected, higher frequencies pass the filter.

"RF reject" Frequencies higher than 50 kHz are rejected, lower frequencies pass the filter.

Remote command:
`TRIGger<m>:ANEDge:FILTer` on page 1375

**Slope**
Sets the edge type for the trigger condition.

"Positive" Selects the rising edge, which is a positive voltage change.

"Negative" Selects the falling edge, which is a negative voltage change.

"Both" Selects the rising as well as the falling edge. This option is not available if the trigger source is the external trigger input.

Remote command:
`TRIGger<m>:EDGE:SLOPe` on page 1308
`TRIGger<m>:ANEDge:SLOPe` on page 1377
`TRIGger<m>:SLEW:SLOPe` on page 1319

**Trigger level**
Sets the voltage level for the trigger condition. You can also drag the trigger level marker on the display (TA or TB on the right edge of the display). The range of the trigger level is limited in a way so that always a hysteresis for stable trigger conditions is available.

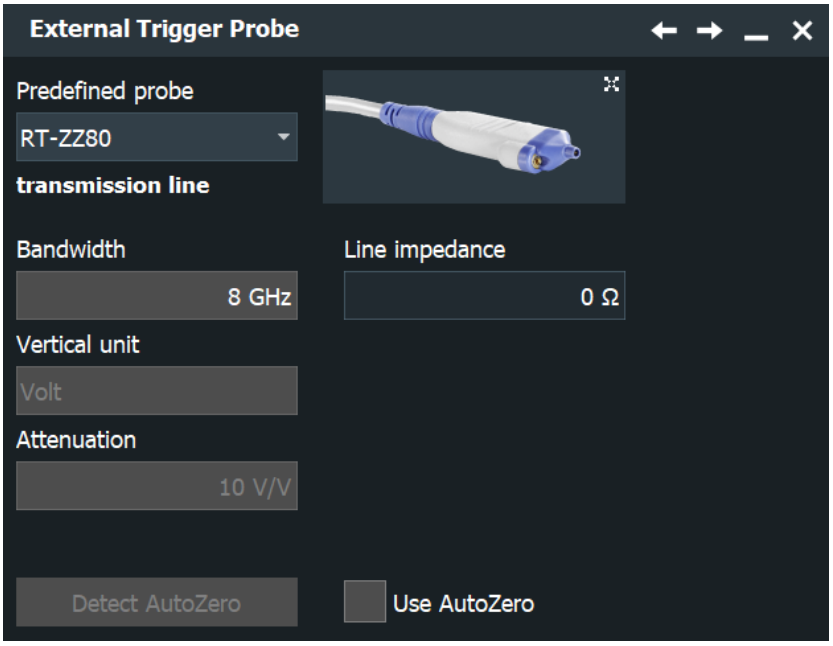
Remote command:
`TRIGger<m>:LEVel<n>[:VALue]` on page 1306

### 6.11.2 Probe setup for external trigger input

Access: "Trigger" > "Setup" > "Source" = "Extern" > "Setup external probe"

The "External Trigger Probe" tab provides all settings that are relevant for the probe that is connected to the external trigger input. The functionality on the tab changes according to the type of the attached probe. They are the same as for probe setup of input channels: see Chapter 5.6, "Probes", on page 146.

If the external trigger input is overloaded, a message informs you.




Remote commands:

- `TRIGger<m>:EXTern:OVERload` on page 1377
- `TRPRobe:ID:PARTnumber?` on page 1277
- `TRPRobe:SETup:STATe?` on page 1255
- `TRPRobe:SETup:TYPE?` on page 1255
- `TRPRobe:SETup:ATTenuation:DEFProbe` on page 1274
- `TRPRobe:SETup:ATTenuation:MANual` on page 1263
- `TRPRobe:SETup:ATTenuation:MODE` on page 1263
- `TRPRobe:SETup:ATTenuation:UNIT` on page 1263
- `TRPRobe:SETup:ATTenuation[:AUTO]?` on page 1256
- `TRPRobe:SETup:NAME?` on page 1256
- `TRPRobe:SETup:BANDwidth?` on page 1256
- `TRPRobe:SETup:CMOFfset` on page 1264
- `TRPRobe:SETup:GAIN:MANual` on page 1264
- `TRPRobe:SETup:ZAXV` on page 1265

R&S ProbeMeter: remote commands:

- `TRPRobe:SETup:DISPlaydiff` on page 1260
- `TRPRobe:PMETer:VISibility` on page 1260
- `TRPRobe:PMETer:RESults:COMMon?` on page 1261
- `TRPRobe:PMETer:RESults:DIFFerential?` on page 1261
- `TRPRobe:PMETer:RESults:NEGative?` on page 1262
- `TRPRobe:PMETer:RESults:POSitive?` on page 1262
- `TRPRobe:PMETer:RESults:SINGle?` on page 1260

Probe attributes: remote commands

- TRPRobe:ID:PRDate? on page 1277
- TRPRobe:ID:SRNumber? on page 1278
- TRPRobe:ID:SWVersion? on page 1277
- TRPRobe:SETup:CAPacitance? on page 1278
- TRPRobe:SETup:IMPedance? on page 1278

# 7 Waveform analysis

This chapter describes general methods to check and analyze waveforms. These are:

- Zoom....261
- Reference waveforms....271
- Mathematics....279
- History....298
- XY-diagram....304

## 7.1 Zoom

The zoom functions allow you to magnify a specific section of the diagram in order to view more details. You can define several zoom areas for the same diagram and even couple them, or you use the hardware zoom.

### 7.1.1 Methods of zooming

The R&S RTP provides various ways of zooming: You define the section of a diagram that you want to magnify, and the zoomed view is shown in a separate zoom diagram. Additionally, you can magnify the diagram directly: The hardware zoom changes the horizontal and vertical scales of the diagram so that you see the selected section.

There are different ways to initiate and configure the zoom function:

- **Fingertip zoom**: magnifies the waveforms around your fingertip. When you drag your finger, the magnifier moves, too. You can convert the fingertip zoom into a standard zoom diagram.
- **Graphical method**: you draw, move and adjust the zoom area on the touchscreen – a very quick and simple method for standard zoom and hardware zoom.
- **Numeric method**: you enter x- and y-values in a dialog box or adjust them using navigation controls. These are precise ways which can be used to optimize a graphically defined zoom.
  With the numeric method there are two ways of defining the zoom area:
  – Specifying **start and stop values** for the x- and y-axes; the acquired data within those values is zoomed.

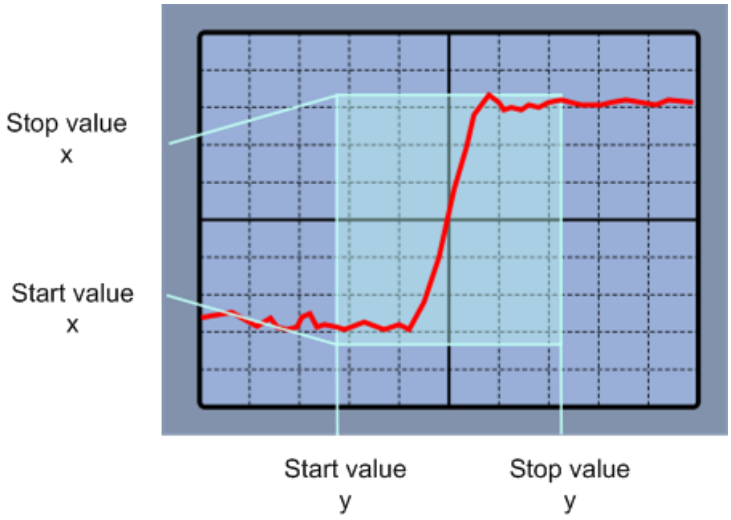



*Figure 7-1: Numeric zoom using start and stop values*

– Specifying the x- and y-**position** of the centerpoint of the area plus a **range** for the x- and y-axes; the area defined by that centerpoint and the ranges is zoomed.

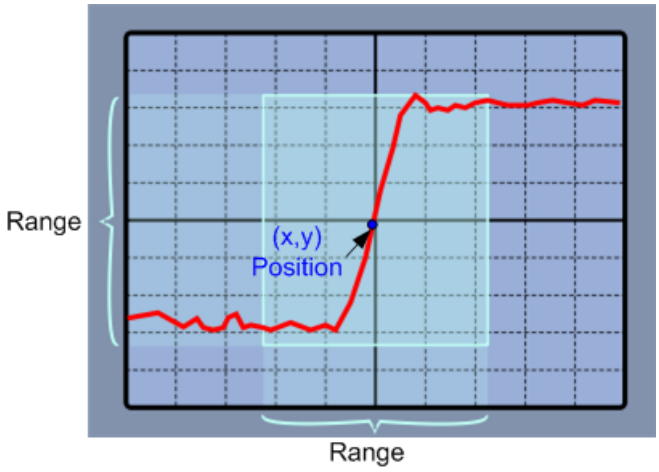



*Figure 7-2: Numeric zoom using position and range*

- **Coupled zoom** creates a copy of the selected zoom area. Coupled zoom areas always have the same size (size coupling). They can be positioned separately or together (position coupling).

Zoom areas can be used for gating, for example, to define a measurement gate. You can set the gate exactly to the limits of the zoom.

Evaluation gates - available histogram areas, masks, and measurement gates - can be displayed in zoom diagrams to simplify the graphical gate adjustment on the touch-screen.

See: Chapter 7.1.3, "Zooming for details", on page 267.

## 7.1.2 Zoom settings

The zoom area, i.e. the section to be enlarged, can be defined using two different methods:

- Using the zoom functions on the toolbar and draw the zoom area on the touch-screen
- Specifying numeric values:
  - start and stop values for the x- and y-axes
  - x and y position of one point in the diagram plus a range for the x- and y-axes

  See also: Chapter 7.1.1, "Methods of zooming", on page 261.

- Zoom functions on the toolbar.....263
- Zoom settings.....263

### 7.1.2.1 Zoom functions on the toolbar

The zoom icon on the toolbar shows the last selected zoom type. A short tap on the icon activates the selected zoom. If you touch the icon and drag your finger down, a menu opens where you can select another zoom type.

**Standard zoom**
Displays a magnified section of the diagram in an additional zoom diagram. It is a display zoom, instrument settings are not changed.

Touch and hold the zoom area to open the "Zoom" dialog box.

Remote command:
`LAYout:ZOOM:ADD` on page 1379

**Hardware zoom**
Changes the instrument settings - horizontal and vertical scales as well as trigger level and offset - to display a part of the diagram in greater detail.

**Coupled zoom**
Creates a coupled zoom area and its related zoom diagram. If you change the size of one zoom area, the size of all coupled zoom areas is changed as well.

Remote command:
`LAYout:ZOOM:ADDCoupled` on page 1379

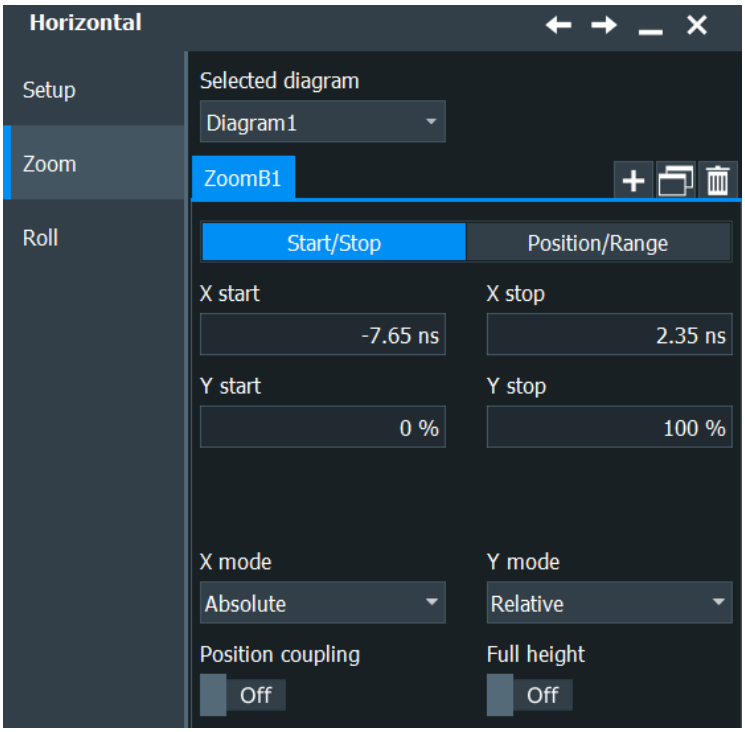
**Fingertip zoom**
Magnifies the waveforms around your fingertip.

Tap the icon and put your finger on the waveform. The touched part of the waveform is displayed in a magnifier. Drag your finger on the screen to move the magnifier. You can change the zoom factor using the [Multiuse] knob.

### 7.1.2.2 Zoom settings

Access: "Menu" > "Horizontal" > "Zoom" tab.

The "Zoom" tab allows you to specify start and stop values for the x- and y-axes. The acquired data within these ranges is zoomed.

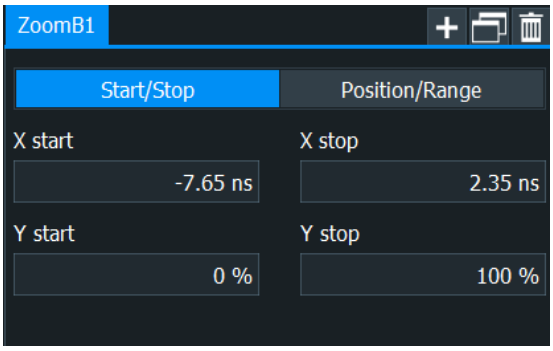



Additionally, you can enable a "Zoom Overlay" in the "Menu" > "Settings" > "Appearance" > "Diagram", see Chapter 4.3.4, "Diagram appearance settings", on page 82.

**Selected diagram**
Indicates which of the waveform diagrams is selected for zooming.

**Mode**
Selects how the window for the zoom diagram is defined. You can select between defining "Start/Stop" values or "Position/Range".

**Start/Stop**
The "Start/Stop" tab allows you to specify start and stop values for the x- and y-axes. The acquired data within these ranges is zoomed.



**X start ← Start/Stop**
Defines the lower limit of the zoom area on the x-axis.

Remote command:
LAYout:ZOOM:HORZ:ABSolute:STARt on page 1381
LAYout:ZOOM:HORZ:RELative:STARt on page 1383

**X stop ← Start/Stop**
Defines the upper limit of the zoom area on the x-axis.

Remote command:
LAYout:ZOOM:HORZ:ABSolute:STOP on page 1382
LAYout:ZOOM:HORZ:RELative:STOP on page 1383

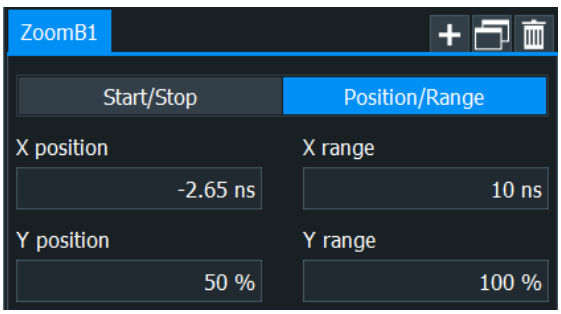
**Y start ← Start/Stop**
Defines the lower limit of the zoom area on the y-axis.

Remote command:
LAYout:ZOOM:VERTical:RELative:STARt on page 1387
LAYout:ZOOM:VERTical:ABSolute:STARt on page 1385

**Y stop ← Start/Stop**
Defines the upper limit of the zoom area on the y-axis.

Remote command:
LAYout:ZOOM:VERTical:RELative:STOP on page 1387
LAYout:ZOOM:VERTical:ABSolute:STOP on page 1385

**Position/Range**
In the "Position/Range" tab, you specify the x and y position of center point of the zoom area plus a range for the x- and y-axes; the area defined by that point and the ranges is zoomed.




**X position ← Position/Range**
Defines the x-value of the centerpoint of the zoom area.

Remote command:
LAYout:ZOOM:HORZ:ABSolute:POSition on page 1381
LAYout:ZOOM:HORZ:RELative:POSition on page 1382
SEARch:RESDiagram:HORZ:ABSolute:POSition on page 1598
SEARch:RESDiagram:HORZ:RELative:POSition on page 1599

**X range ← Position/Range**
Defines the width of the zoom area.

Remote command:
LAYout:ZOOM:HORZ:ABSolute:SPAN on page 1381
LAYout:ZOOM:HORZ:RELative:SPAN on page 1383
SEARch:RESDiagram:HORZ:ABSolute:SPAN on page 1599
SEARch:RESDiagram:HORZ:RELative:SPAN on page 1600

**Y position ← Position/Range**
Defines the y-value of the centerpoint of the zoom area.

Remote command:
LAYout:ZOOM:VERTical:ABSolute:POSition on page 1384
LAYout:ZOOM:VERTical:RELative:POSition on page 1386
SEARch:RESDiagram:VERT:ABSolute:POSition on page 1600
SEARch:RESDiagram:VERT:RELative:POSition on page 1601

**Y range ← Position/Range**
Defines the height of the zoom area.

Remote command:
LAYout:ZOOM:VERTical:RELative:SPAN on page 1386
LAYout:ZOOM:VERTical:ABSolute:SPAN on page 1385
SEARch:RESDiagram:VERT:ABSolute:SPAN on page 1601
SEARch:RESDiagram:VERT:RELative:SPAN on page 1602

**X mode**
Defines whether absolute or relative values are used to specify the x-axis values.

Remote command:
LAYout:ZOOM:HORZ:MODE on page 1380
SEARch:RESDiagram:HORZ:MODE on page 1599

**Y mode**
Defines whether absolute or relative values are used to specify the y-axis values.

Remote command:
LAYout:ZOOM:VERTical:MODE on page 1384
SEARch:RESDiagram:VERT:MODE on page 1601

**Position coupling**
Enables or disables the position coupling of coupled zooms. If position coupling is enabled and you move one zoom area, the other coupled zoom areas are moved, too, and keep their distance.

Remote command:
LAYout:ZOOM:POSCoupling on page 1380

**Full height**
Uses the full diagram height for the zoom area. Only horizontal zoom settings can be changed.

### 7.1.3 Zooming for details

The usage of the various zoom methods is described in the following procedures:

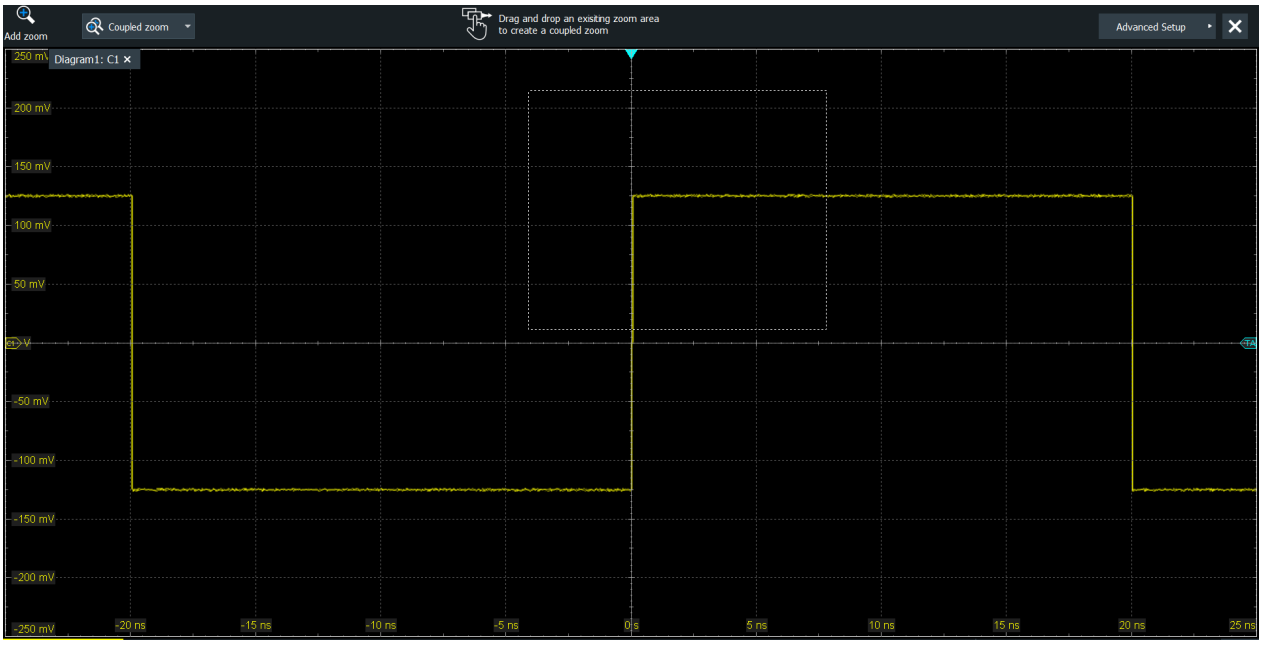
- To define the zoom area graphically on the touchscreen
- To define the zoom area numerically using start/stop values
- To define the zoom area numerically using position and range values
- To define multiple zoom areas
- To define coupled zoom areas

**To define the zoom area graphically on the touchscreen**

For graphical zooming, you use your finger on the screen.

1. On the toolbar, tap the "Standard Zoom" icon.

2. Touch the position that you want to define as one corner of the zoom area. Then drag your finger to the opposite corner of the zoom area.
While you drag your finger on the touchscreen, a dotted rectangle indicates the current zoom area. When the rectangle covers the required zoom area, remove your finger.




The indicated area is magnified in a new zoom diagram. The original diagram is displayed with the zoom area indicated as a rectangle.

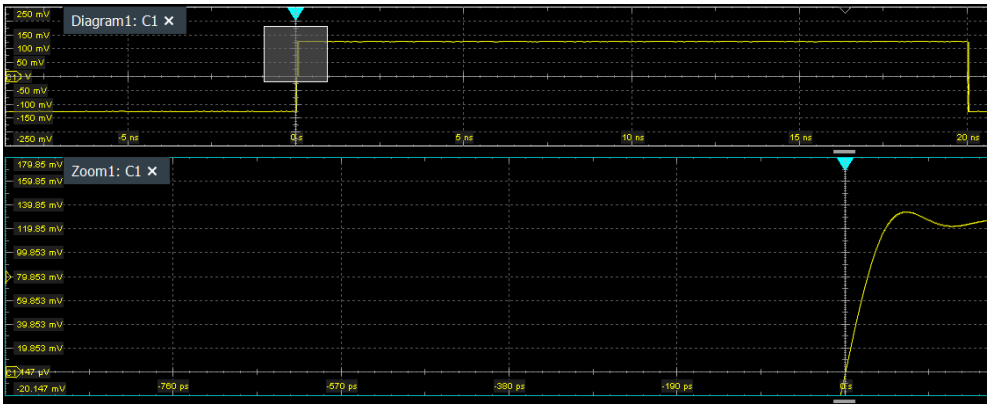



*Figure 7-3: Zoom diagram and overview diagram*

3. If the position of the zoom area is not correct, drag the rectangle in the overview to the correct position.
4. If the size of the zoom area is not yet ideal, tap the rectangle in the overview diagram.
   Now, 4 red lines indicate the edges of the zoom area. A dashed red line indicates the selected edge, which you can adjust.
5. Touch the edge that you want to move, and drag it to the required position.

To optimize the zoom definition of an active zoom diagram, double-tap the zoom diagram. The "Zoom" dialog box for numeric definition is opened.

**To create a new zoom using the Zoom dialog box**

1. There are two ways to create a new zoom:
   - If you want to create a new, unconfigured zoom, tap the "Add" icon.

   - If you want to create zoom based on an existing one, tap the "Copy" icon.

2. Enter a name for the zoom using the on-screen keyboard.

**To define the zoom area numerically using start/stop values**

1. On the "Menu" menu, tap "Horizontal".
2. In the "Zoom" tab, select "Start/Stop".
3. Select "X mode" to define "Absolute" or "Relative" x-axis values. Relative values cause the zoom area to adapt to the input values dynamically.
4. Define the "X start" and "X stop" values that define the lower and upper borders of the zoom area on the x-axis (see Figure 7-1).

5. Select the "Y mode" to define "Absolute" or "Relative" y-axis values.
6. Define the "Y start" and "Y stop" values that define the lower and upper borders (respectively) of the zoom area on the y-axis.

   When you close the dialog box, the specified area is magnified in a new zoom diagram. The original diagram is displayed with the zoom area indicated as a rectangle (see Figure 7-3).

**To define the zoom area numerically using position and range values**

1. On the "Menu" menu, tap "Horizontal".
2. In the "Zoom" tab, select "Position/Range".
3. Select "X mode" to define "Absolute" or "Relative" x-axis values. Relative values cause the zoom area to adapt to the input values dynamically.
4. Under "X position", define the x-value of the center point of the zoom area (see Figure 7-2).
5. Under "X range", define the width of the zoom area.
6. Select the "Y mode" to define "Absolute" or "Relative" y-axis values.
7. Under "Y position", define the y-value of the center point of the zoom area.
8. Under "Y range", define the height of the zoom area.

   When you close the dialog box, the specified area is magnified in a new zoom diagram. The original diagram is displayed with the zoom area indicated as a rectangle.

**To define multiple zoom areas**

You can define more than one zoom area for the same diagram, for example to compare several peaks in a measurement. These zoom areas can be displayed in separate zoom diagrams, or together in one zoom diagram.

To define multiple zoom areas graphically, simply repeat the steps described in To define the zoom area graphically on the touchscreen - for each area. Numerically, proceed as follows:

1. On the "Menu" menu, tap "Horizontal".
2. Select the "Zoom" tab.
3. Select the required tab according to the method you want to use to define the zoom area.
4. To copy the current zoom area definition, tap the "Copy" icon.
   Alternatively, tap the "Add" icon to add a new zoom area.
5. Enter a name for the new zoom diagram using the displayed on-screen keyboard.
6. Define the zoom area as described for the first zoom.

An additional zoom diagram is displayed for the new zoom area, and another rectangle in the original diagram indicates the new zoom area. Each rectangle in the overview has the same color as the corresponding zoom diagram frame.

7. Alternatively, you can overlay the zoom areas: The zooms are shown in the same zoom diagram, as if the zoom areas are overlaid:
In the "Settings" > "Appearance" dialog > "Diagram " tab , enable "Zoom > Overlay".

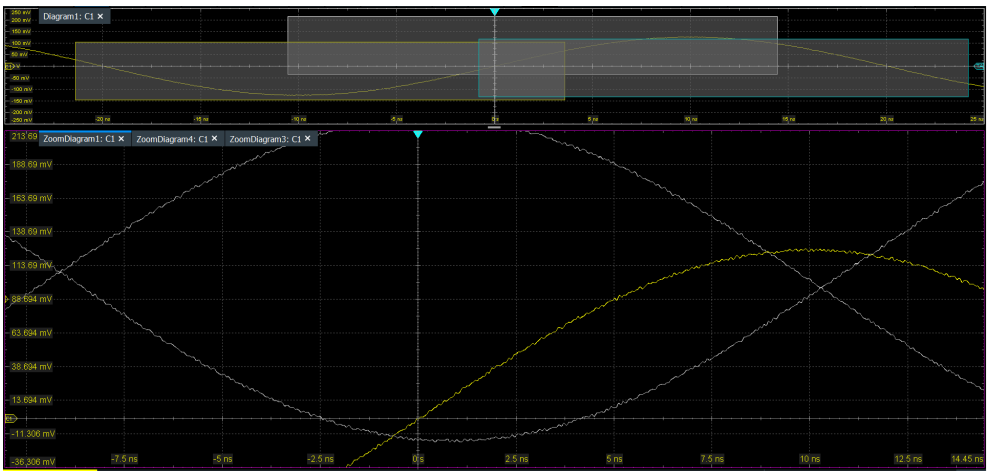



*Figure 7-4: Multiple zoom diagrams. Left: separate zoom diagrams. Right: overlaid zoom*

### To define coupled zoom areas

You can define multiple zoom areas for one diagram that are coupled. If you change the size of one zoom area, the size of all coupled zoom areas is changed as well. Furthermore, you can couple also the position in order to move all coupled zooms at once. Coupling is useful, for example, if you want to compare recurring peaks in a signal.

1. On the toolbar, tap the "Coupled Zoom" icon.

2. In the diagram overview, tap an existing zoom area.

   The selected zoom area is duplicated.

3. Drag the duplicate zoom area to the required position.
4. To create further coupled zooms, repeat the steps above.

   Now, if you change the zoom area size of any of the coupled zoom areas in the "Zoom" dialog box, the settings are changed for all coupled zoom areas.

5. In the "Zoom" dialog box, select the diagram that contains the coupled zooms.
6. Select a zoom tab.
7. Enable "Position coupling".

If you move one of the coupled zoom areas in the diagram, all other coupled zooms are moved as well, and their distance is kept unchanged.

**To use the hardware zoom**

In contrast to the normal zoom, the hardware zoom changes the instrument settings - horizontal and vertical scales, and also the trigger level and offset. Thus, the selected area is displayed in the diagram instead of the original waveform. No additional zoom diagram is opened.

1. On the toolbar, tap the "Hardware Zoom" icon.

2. Drag your finger on the touch screen to mark the zoom area.
   A dotted rectangle indicates the current zoom area. When the rectangle covers the required zoom area, remove your finger. The diagram changes and shows the magnified area.

   **Tip:** To return to the previous display, use the "Undo" icon.

   **Note:** You can combine hardware zoom and normal zoom - first use the hardware zoom, then the zoom into the display. The reverse approach is also possible: Create a zoom diagram, and then apply the hardware zoom to the waveform diagram. Both the waveform and the zoom diagrams are changed.

## 7.2 Reference waveforms

You can configure up to four reference waveforms to display stored waveforms. Any active signal or mathematical waveform can be stored as a reference waveform. It can then be loaded again later to restore the waveform on the screen.

### 7.2.1 Working with reference waveforms

Reference waveforms can be displayed in addition to the signal waveforms, saved to file, and loaded back for further analysis. Reference waveforms can be loaded only from BIN files.

**Note:** Saving and loading reference waveforms, and preset with active reference waveform delete the undo stack. After these actions, undo is not possible.

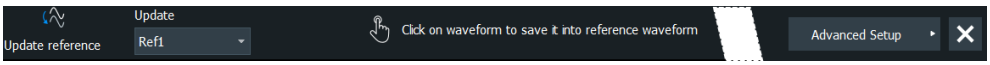
**To update a reference waveform using the toolbar icon**

If you often need to update a reference waveform, you can use the "Save Reference" toolbar icon.

1. Add the "Save Reference" icon to the toolbar, see Chapter 3.3.6.2, "Configuring the toolbar", on page 56 .

2. Tap the icon.

   The overlay menu opens.




3. Set "Update" to the reference waveform to be used.
4. Tap the waveform to be used as reference waveform.

**To display a reference waveform**

1. In the "Menu" > "Apps" > "Analysis" tab, select "Reference Curve".
   Alternatively, press the [Ref] key.
2. Select the tab for the reference waveform you want to display ("R1"-"R4").
3. Load a stored reference waveform as described in "To load a reference waveform" on page 273, or select a source to be displayed as a reference:
   a) In the "Setup" tab, select "Signal".
   b) Select the "Source" from the selection list. The source can be any active signal, math, or other reference waveform.
   c) Tap the "Update" button to update the current reference waveform with the source data.
4. Tap the "Show" button.

   The reference waveform is displayed on the screen.
5. A reference waveform can have its own scaling settings or it can be scaled according to the source settings. By default, the scaling of the reference waveform is coupled to the source settings. Additionally, it can be stretched or compressed in vertical and horizontal direction.
   If necessary, change the settings on the "Vertical" and "Rescale" tabs of the "Reference Waveform" dialog boxes.
   To restore the original settings, tap the "Restore settings" in the "Vertical" tab.
   For a description of the scaling settings, see Chapter 7.2.2.4, "Rescale", on page 277

**To save a reference waveform**

1. In the "Save/Recall" dialog, select "Save > Waveform".

   **Tip:** You can also save a waveform as a reference waveform in the "File" dialog box, see Chapter 12.2.7, "Saving and loading waveform data", on page 505. Here, you can also save multiple waveforms in one file.
2. Select the tab for the reference waveform you want to store ("Ref1"-"Ref4").
3. Display and configure the reference waveform as described in "To display a reference waveform" on page 272.

4. Select the file format.

   **Note:** Reference waveforms can be loaded only from BIN files. XML and CSV formats are meant for further processing in other applications.

5. To save the waveform to the currently selected file, tap "Save". By default, the prefix for reference waveform files is "RefCurve".
   To save the waveform to another file, select "Save As".
   Enter a file name and select the directory. The file type is already defined according to the selection in the previous step. In order to load the reference waveform on the instrument again later, use the file type BIN.

   The source settings of the reference waveform and the current scaling settings are stored to the specified file.

**To load a reference waveform**

**Note:** Reference waveforms can be loaded only from BIN files.

1. Press the [Ref] key.
2. Select the tab for the reference waveform you want to load ("Ref1" - "Ref4").
3. In the "Setup" tab, select "File".
4. To load the waveform from the specified file, tap "Open".
   To load the waveform from a different file, tap "Open". Select the file from the file selection dialog box. Only BIN files are displayed in the file list.

   The selected waveform is loaded as the specified reference waveform.
   If multiple waveforms are saved in the file, you are asked to assign each waveform to a reference waveform. All waveforms are loaded together.

**To view a reference waveform**

► You can view a reference waveform, using the "Saveset with preview" function.
For details, see: Chapter 12.1.4, "Saveset with preview function", on page 479.

## 7.2.2 Settings for reference waveforms

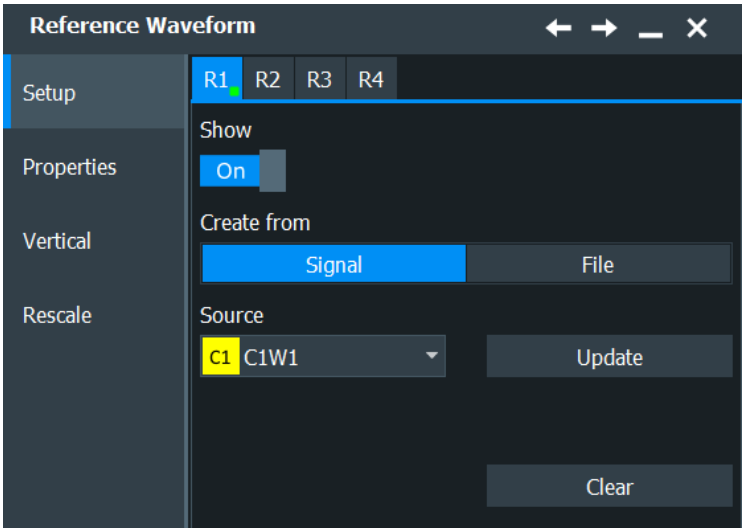
To compare waveforms and analyze differences between waveforms, you can use up to four reference waveforms R1 to R4. Each reference waveform has its own memory on the instrument. You can also save an unlimited number of reference waveforms and load them for further use.

The display of a reference waveform is independent from the display of the source waveform; you can move, stretch and compress the curve vertically and horizontally.

### 7.2.2.1 Reference waveform setup

Access: [Ref] key > "Setup" tab

In the "Setup" tab, you select the reference waveform and its source. The source can be an active waveform - trace of an input channel, math waveform or another reference waveform - or a stored waveform.




R1/2/3/4 ..... 274
Show ..... 274
Signal ..... 274
Source ..... 274
Update ..... 275
File ..... 275
Clear ..... 275

**R1/2/3/4**
Each tab contains the settings for one of the four available reference waveforms.

**Show**
Displays the reference waveform in the diagram.

Remote command:
`REFCurve<m>:STATe` on page 1388

**Signal**
Selects a signal as the reference waveform.

**Source**
Selects the source signal from the active waveforms of input channels, math signals and other reference waveforms.

If the selected source is defined as average or envelope waveform, make sure that all required acquisitions ("Average count") are done before you create the reference waveform. Otherwise, the resulting reference waveform is not correct.

Remote command:
`REFCurve<m>:SOURce` on page 1388

**Update**
Copies the selected source waveform with all its settings to the memory of the reference waveform. If the acquisition is running, the reference waveform is a snapshot.

Remote command:
`REFCurve<m>:UPDate` on page 1389

**File**
Opens a file selection dialog box and loads the selected reference waveform file.

Double-tap the filename to open the file selection dialog box, see also Chapter 12.5, "File selection dialog", on page 513.

**Note:** Note that reference waveforms can be loaded from BIN files only. XML and CSV formats are meant for further processing in other applications.

Remote command:
`REFCurve<m>:OPEN` on page 1389
`REFCurve<m>:SAVE` on page 1389
`REFCurve<m>:DELete` on page 1390

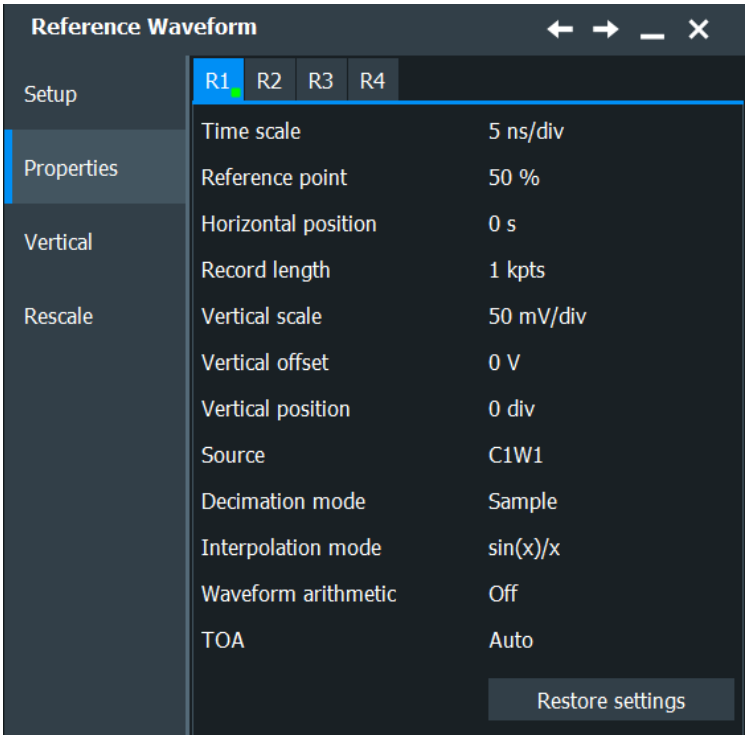
**Clear**
The selected reference waveform disappears, its memory is deleted.

Remote command:
`REFCurve<m>:CLEar` on page 1390

#### 7.2.2.2 Properties

Access: [Ref] key > "Properties" tab

A reference waveform can be scaled, stretched and positioned in the diagram. The "Original Attributes" tab shows the original settings of the reference waveform, which are stored together with the waveform data.

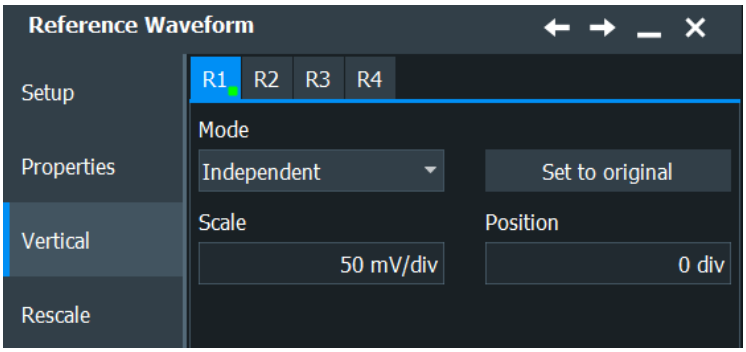



**Restore Settings**
Resets the time scale and the reference point to the original values of the reference waveform.

Remote command:
`REFCurve<m>:RESTore` on page 1391

### 7.2.2.3 Vertical

Access: [Ref] key > "Vertical" tab




**Mode**
Selects the type of vertical settings:

"Coupled to source" Vertical position and scale of the source are used.

"Independent" Scaling and position can be set specific to the reference waveform.

Remote command:
REFCurve<m>:VMODe on page 1391

**Set to original**
Restores the original vertical settings of the reference waveform (vertical scale, position, and offset), if vertical scaling is set to "Independent".

Remote command:
REFCurve<m>:TOORignal on page 1392

**Scale**
Sets the vertical scale for the reference waveform, if vertical scaling is set to "Independent". You can also use the vertical [Scale] knob to adjust this value.

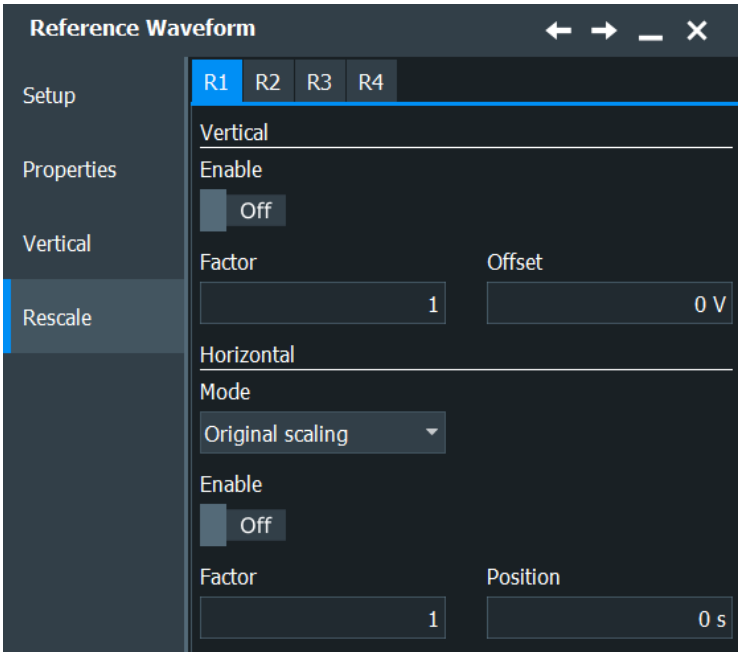
Remote command:
REFCurve<m>:SCALe on page 1391

**Position**
Moves the reference waveform up or down in the diagram, if vertical scaling is set to "Independent". If "Position" is assigned to the vertical [Position] knob, you can also use this knob to adjust the "Vertical position"

Remote command:
REFCurve<m>:POSition on page 1392

### 7.2.2.4 Rescale

Access: [Ref] key > "Rescale" tab



**Vertical Rescaling**
Stretching and offset change the display of the waveform independent of the vertical scale and position.

**Enable ← Vertical Rescaling**
If enabled, the vertical offset and factor are applied to the reference waveform.

Remote command:
`REFCurve<m>:RESCale:VERTical:STATe` on page 1392

**Factor ← Vertical Rescaling**
A factor greater than 1 stretches the waveform vertically, a factor lower than 1 compresses the curve.

Remote command:
`REFCurve<m>:RESCale:VERTical:FACTor` on page 1392

**Offset ← Vertical Rescaling**
Moves the reference waveform vertically. Enter a value with the unit of the waveform. Like vertical offset of a channel waveform, the offset of a reference waveform is subtracted from the measured value. Negative values shift the waveform up, positive values shift it down.

**Note:** As for all waveforms, a vertical offset of a reference waveform can be set using the vertical [Position] knob. This offset is independent from the reference scaling offset, which is described here. If both offsets are set, their values are added up.

Remote command:
`REFCurve<m>:RESCale:VERTical:OFFSet` on page 1393

**Mode ← Vertical Rescaling**
Selects the type of horizontal settings:

| | |
|---|---|
| "Adjust to X Axis" | The current horizontal settings of the diagram are used. |
| "Original Scaling" | Horizontal scaling and reference point of the source waveform are used. |

Remote command:
`REFCurve<m>:HMODe` on page 1393

**Horizontal Rescaling ← Vertical Rescaling**
Stretching and offset change the display of the waveform independent of the horizontal settings of the source waveform and of the horizontal diagram settings.

**Enable ← Horizontal Rescaling ← Vertical Rescaling**
If enabled, the horizontal offset and factor are applied to the reference waveform.

Remote command:
`REFCurve<m>:RESCale:HORizontal:STATe` on page 1394

**Factor ← Horizontal Rescaling ← Vertical Rescaling**
A factor greater than 1 stretches the waveform horizontally, a factor lower than 1 compresses the curve.

Remote command:
REFCurve<m>:RESCale:HORizontal:FACTor on page 1394

**Position ← Horizontal Rescaling ← Vertical Rescaling**
Moves the waveform horizontally. Enter a value with a time unit suitable for the time scale of the diagram. Positive values shift the waveform to the right, negative values shift it to the left.

Remote command:
REFCurve<m>:RESCale:HORizontal:POSition on page 1395

## 7.3 Mathematics

Math waveforms are calculated waveforms. You can define up to eight math waveforms and display them on the screen, and use it as source for further analysis.

Math waveforms are defined by mathematical expressions (formulas). You can enter mathematical expressions using different methods:

- "Operator" tab: you define a simple math function in a graphical editor by selecting the source waveforms and the operator.
- "Equation" tab: you define sophisticated math functions in a formula editor, as required to your needs.

The result of an FFT analysis is a specific math waveform. For information on FFT and spectrograms, see Chapter 9, "Spectrum analysis", on page 399.

The vertical scale of a math waveform is adapted automatically to the measurement results to ensure optimal display. Furthermore, you can scale each math waveform manually in vertical direction like a channel waveform.

As for channel waveforms, you can also change the arithmetic mode for the waveform to display the envelope or an average over several calculations.

You can store a math waveform as a reference waveform and restore it later, see "To save a reference waveform" on page 272.

- Displaying math waveforms....279
- Math setup - general settings....280
- Operator editor....282
- Equation....284
- Filters....292
- Script (option R&S RTP-K39)....295
- Math scale settings....295
- Math arithmetic....296

### 7.3.1 Displaying math waveforms

Math waveforms can be displayed in addition to the channel and other waveforms. They also can be used for analysis, e.g. measurements, even if the math waveform is not active.

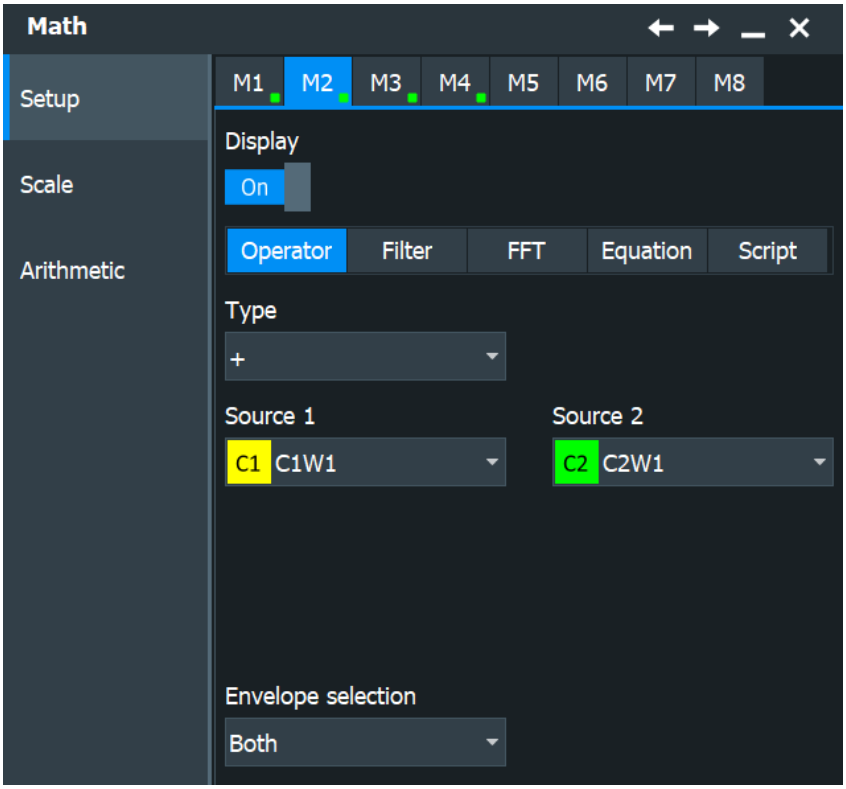
1. In the "Math" menu, select "Math Setup".
   Alternatively, press the [Math] key.
2. Define the math expression for calculation in one of the following ways:
   - Chapter 7.3.3.2, "Defining a formula in the Operator editor", on page 283
   - Chapter 7.3.4, "Equation", on page 284
   - Chapter 9.1.2, "Configuring spectrum waveforms", on page 403
3. In the "Math Setup" dialog box, in the "Setup" tab, tap the "Enable math signal".

   The math waveform is displayed on the screen.
4. To change the vertical scaling of the math waveform, tap the "Manual" icon.
5. Enter the "Vertical scale" factor (per division). If necessary, add a "Vertical offset". By default the instrument performs an automatic scaling.

   **Tip:** You can also use the vertical [Scale] rotary knob for scaling. In this case, the scale mode is set to "Manual" temporarily.
6. If you need the envelope or average of the math waveform over several calculations, change the arithmetic mode for the waveform as for channel waveforms.
   See also: "Arithmetic" on page 133.
7. Close the "Math Setup" dialog box.

### 7.3.2 Math setup - general settings

You can define up to eight different math waveforms. Each waveform is defined in a separate tab in the "Math" dialog box ("M1" to "M8").




The settings for input of mathematical formulas are described in separate chapters:

- Chapter 7.3.3.1, "Settings in the Operator editor", on page 282
- Chapter 7.3.5, "Filters", on page 292
- Chapter 9.1.3, "FFT configuration settings", on page 407
- Chapter 7.3.4, "Equation", on page 284

Display ..... 281
Envelope selection ..... 281

**Display**
If activated, a diagram for the defined math waveform is displayed on the touchscreen.

Remote command:
`CALCulate:MATH<m>:STATe` on page 1401

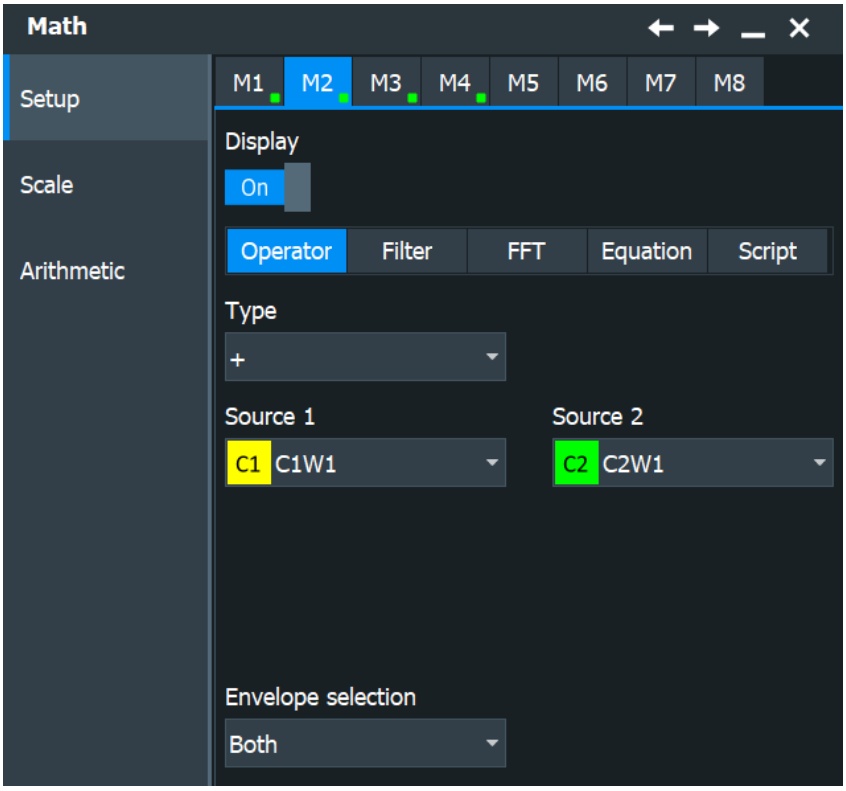
**Envelope selection**
Selects the upper or lower part of the waveform for mathematic calculation, or a combination of both.

The setting is relevant for waveforms with waveform arithmetic mode "Envelope" or with "Peak detect" decimation. All mathematic operations - except for derivation - can be applied to envelope waveforms and waveforms with "Peak detect" decimation.

Remote command:
`CALCulate:MATH<m>:ENVSelection` on page 1401

### 7.3.3 Operator editor

In the Operator editor, you can define the most common mathematical formulas without knowing their correct syntax.




Remote command:

- `CALCulate:MATH<m>[:EXPRession][:DEFine]` on page 1400

#### 7.3.3.1 Settings in the Operator editor

Source 1 / 2 ..... 282
Type ..... 282
Noise reject ..... 283
a / b ..... 283

**Source 1 / 2**
Defines the signal source to be evaluated by the math function. Waveform 1 of channel signals can be selected.

**Note:** If you require other signal sources not listed here, use the formula editor provided in the Equation tab. Any waveform of any input channel can be used as a source. See: Chapter 7.3.4, "Equation", on page 284.

**Type**
Defines the type of operation to be performed on the selected signal sources. The following functions are available:

**Note:** If you require other operators not listed here, use the formula editor provided in the "Equation" tab. See: Chapter 7.3.4, "Equation", on page 284.

| | |
|---|---|
| "+" | Adds up the sources |
| "-" | Subtracts source 2 from source 1. |
| "x" | Multiplies source 1 by source 2. |
| "\|x\|" | Determines the absolute value of the source. |
| "dx/dt" | Differentiates the source value with respect to the time value.<br>Not possible on envelope waveforms and waveforms with "Peak detect" decimation. |
| "log(x)" | Calculates the logarithm of the source value based on 10. |
| "ln(x)" | Calculates the natural logarithm of the source value (based on e). |
| "ld(x)" | Calculates the binary logarithm of the source value (binary logarithm, based on 2). |
| "Rescale" | Rescales the source values by a factor "a" and an offset "b": *ax+b*.<br>See also: "a / b" on page 283. |

**Noise reject**
Only available for "Type" = "dx/dt".

Sets the number of neighboring samples that are skipped for differentiation.

To suppress noise effects during differentiation, it can be useful not to consider two directly neighboring points to calculate *dx* ($x_n$-$x_{n-1}$). Instead, some samples in-between are skipped and a point a few samples further is used (e.g. $x_n$-$x_{n-3}$).

**a / b**
Defines the values for the "Rescale" function (ax+b).

| | |
|---|---|
| "a" | Multiplication factor |
| "b" | Offset of the signal source on the y-axis. |

### 7.3.3.2 Defining a formula in the Operator editor

1. Open the "Menu" > "Math" dialog.
   Alternatively, press the [Math] key.
2. In the "Setup" tab, select the "Operator" tab.
3. Tap the "Source 1" and "Source 2" icons and select the signal sources to which the math function is applied. For details on available signal sources, see "Source 1 / 2" on page 282.
4. Tap the "Type" icon and select the mathematical function.
   For details on available operators, see "Type" on page 282.
5. If the operator requires additional parameters, enter them in the input fields.

## 7.3.4 Equation

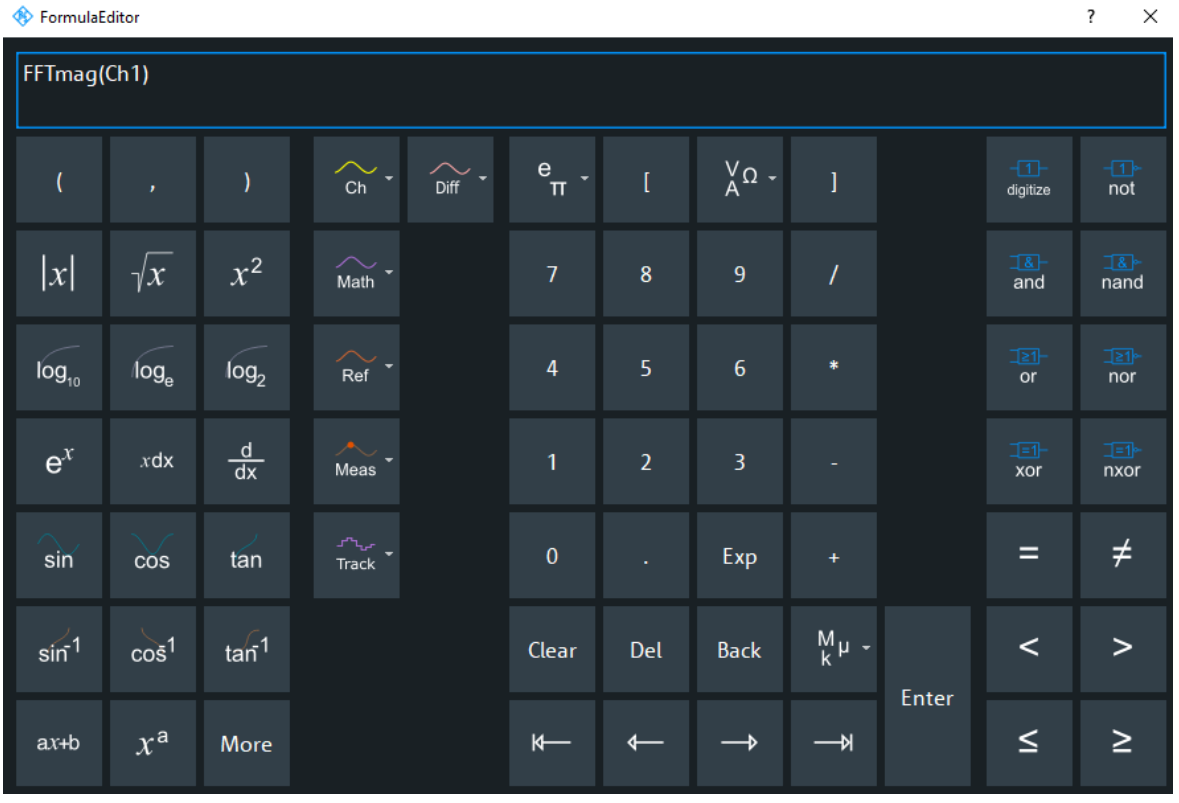
In the Equation tab, you can enter complex formulas to define a math waveform. The formula editor helps to enter formulas easily with correct syntax, using a large selection of operators and signal sources.

All formulas in the formula editor are linear regardless whether a [dB] is set or not.

► Double-tap the Equation tab to display the formula editor.

### 7.3.4.1 Formula editor

Using the formula editor you can define math functions freely, using a large selection of operators and signal sources. For a procedure on using the editor, see Chapter 7.3.4.11, "Defining a formula in the formula editor", on page 291.




Remote command:

- `CALCulate:MATH<m>[:EXPRession][:DEFine]` on page 1400

## Buttons of the formula editor and their usage

***Table 7-1: Basics***

| Icon | Description | Usage/Comment, *FormulaEditor expression* |
|---|---|---|
| ( | Left bracket | Enclose operands |
| , | Comma | Separates operands |
| ) | Right bracket | Enclose operands |
| e / π | Math. constants | e: Euler number: 2.7182...<br>Pi: 3.1415... |
| [ | Left square bracket | Enclose unit |
| V / A / Ω | Units | [<unit>] |
| ] | Right square bracket | Enclose unit |
| $x^a$ | Exponentiation with base x | x: base, a: exponent<br>*x^a* |
| / | Division | |
| * | Multiplication | |
| - | Subtraction | |
| + | Addition | |
| 0...9 | Numeric characters | |
| . | Decimal point | |
| Exp | Exponentiation with base 10 | *e* |
| Enter | Expression complete | Insert expression in "Setup" dialog and close the formula editor |
| Clear | Clear expression in editor | Restart editing |
| Del | Delete | Remove selected part of expression |
| Back | Backspace | Remove last symbol, operator or operand to the left of the cursor |
| M / k / µ | SI-prefix for unit | <SI-prefix>[<unit>] |

***Table 7-2: Signal sources***

| Icon | Description | Usage/Comment, *FormulaEditor expression* |
|---|---|---|
| Ch | Signal waveform | *Ch<1...4>Wfm<1...3>* |
| Math | Math waveform | *Math<1...8>* |
| Ref | Reference waveform | *Ref<1...4>* |
| Meas | Measurement waveform | *Meas<1...8>* |

| Icon | Description | Usage/Comment, *FormulaEditor expression* |
|---|---|---|
| Track | Track waveform | *Track<1...8>* |
| Diff | Differential and common mode waveforms<br>Waveforms of differential deembedding | *Diff<1...2>*<br>*Common<1...2>* |

***Table 7-3: Cursor keys***

| Icon | Description | Usage/Comment, *FormulaEditor expression* |
|---|---|---|
| \|← | Move cursor to beginning | |
| ← | Move cursor 1 step to the left | |
| → | Move cursor 1 step to the right | |
| →\| | Move cursor to end | |

### 7.3.4.2 Math functions: algebra

***Table 7-4: Algebra***

| Icon | Description | Usage/Comment, *FormulaEditor expression* |
|---|---|---|
| \|x\| | Absolute x value | *abs(x)* |
| √x | Square root of x | *sqrt(x)* |
| $x^2$ | x*x | *pow(x)* |
| $\log_{10}$ | Decimal logarithm (base 10) | *log(x)* |
| $\log_e$ | Natural logarithm (base e) | *ln(x)* |
| $\log_2$ | Binary logarithm (base 2) | *ld(x)* |
| $e^x$ | Exponentiation with base e | *exp(x)* |
| $\int x dx$ | Integral of x | *integral(x)* |
| d/dx | Derivation of x | *derivation(x,y)*<br>With x = waveform and y = number of skipped samples (noise reject) |
| ax+b | Scaling of x | *rescale(x,a,b)* |

### 7.3.4.3 Math functions: trigonometry

***Table 7-5: Trigonometry (More keys)***

| Icon | Description | Usage/Comment, *FormulaEditor expression* |
|---|---|---|
| sinh | Hyperbolic sine | *sinh(x)* |
| cosh | Hyperbolic cosine | *cosh(x)* |
| tanh | Hyperbolic tangent | *tanh(x)* |

### 7.3.4.4 Math functions: bit operations

*Table 7-6: Bit operations*

| Icon | Description | Usage/Comment, *FormulaEditor expression* |
|---|---|---|
| digitize | Convert to 0 or 1 | *digitize(x)*<br>The function uses the middle reference level and the hysteresis, which are set in "Menu" > "Measure" > "Advanced" tab > "Reference Level". |
| not | Negation | *not(x)* |
| and | | *and* |
| nand | Negation of and | *nand* |
| or | | *or* |
| nor | Negation of or | *nor* |
| xor | Exclusive or | *xor* |
| nxor | Negation of exclusive or | *nxor* |

### 7.3.4.5 Math functions: comparison

*Table 7-7: Comparison*

| Icon | Description | Usage/Comment, *FormulaEditor expression* |
|---|---|---|
| = | Equal | = |
| ≠ | Not equal | <> |
| < | Smaller | < |
| > | Greater | > |
| ≤ | Smaller or equal | <= |
| ≥ | Greater or equal | >= |
| More | Display additional keys | |

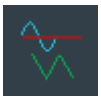
### 7.3.4.6 Math functions: FFT

*Table 7-8: FFT (More keys)*

| Icon | Description | Usage/Comment, *FormulaEditor expression* |
|---|---|---|
| \|FFT\| | Magnitude of FFT value | *fftmag(x)* |
| FFT (φ) | FFT phase value | *fftphi(x)* |
| FFT -dφ*df | FFT group delay | *fftgroupdelay(x)* |
| FFT (re) | Real part of FFT value | *fftre(x)* |
| FFT (im) | imag part of FFT value | *fftim(x)* |

### 7.3.4.7 Math functions: correlation

*Table 7-9: Correlation (More keys)*

| Icon | Description | Usage/Comment, *FormulaEditor expression* |
|---|---|---|
|  | Cross-correlation function of two waveforms<br>Measures the similarity of two waveforms as a function of a time-lag applied to one of them.<br>Function limits the maximum record length to 4 Msample.<br>Two modes of normalization are supported: biased and unbiased.<br>The length of the correlation buffer is $N_0 + N_1 - 1$ samples. The length of the first input signal is $N_0$ samples and the length of the second signal is $N_1$ samples. | *correlation(x1, x2, biased)*<br>*correlation(x1, x2, unbiased)*<br>with x1 = waveform 1 and x2 = waveform 2<br>*correlation(x1, x2)* performs an unbiased correlation |
| | Auto correlation<br>Used to find repeating patterns, for example, a periodic signal obscured by noise.<br>The length of the auto correlation buffer is $2N - 1$ samples, if the length of the input signal is N samples.<br>Two modes of normalization are supported: biased and unbiased. | *autocorrelation(x, biased)*<br>*autocorrelation(x, unbiased)*<br>with x = channel waveform<br>*autocorrelation(x)* performs an unbiased autocorrelation |
| | Biased / unbiased normalization for correlation and auto correlation | see above |

Mathematic background for correlation:

$$Temp1_R_{xy}(m) = \sum_{n=0}^{N1} y_n^* x_{n+m} \quad m \in [0; N_1[$$

$$Temp0_R_{xy}(m) = \sum_{n=1}^{N0} x_n^* y_{n+m} \quad m \in [1; N_0[$$

The R&S RTP uses only the real part of the signal. Two modes of normalization are supported: biased and unbiased.

$$R_{xy}(m) = \begin{cases} \frac{1}{\min(N_0, N_1)} Temp1_R_{xy}(m) & m \in [N_0 - 1; N_1 + N_0 - 1[ \\ \frac{1}{\min(N_0, N_1)} Temp0_R_{xy}^*(-m) & m \in [0; N_0 - 1[ \end{cases}$$

*Equation 7-1: Biased correlation*

$$R_{xy}(m)=\begin{cases}\frac{1}{a(m)}Temp1_R_{xy}(m) & m\in[N_0-1;N_1+N_0-1[\\ \frac{1}{a(m)}Temp0_R^{*}_{xy}(-m) & m\in[0;N_0-1[\end{cases}$$

*Equation 7-2: Unbiased correlation*

Mathematic background for auto correlation:

$$R_{xx}(m)=\begin{cases}\frac{1}{N}\sum_{n=0}^{N-m-1}x_n^{*}x_{n+m} & m\geq 0\\ R_{xx}^{*}(-m) & m<0\end{cases}$$

*Equation 7-3: Biased auto correlation*

$$R_{xx}(m)=\begin{cases}\frac{1}{N-|m|}\sum_{n=0}^{N-m-1}x_n^{*}x_{n+m} & m\geq 0\\ R_{xx}^{*}(-m) & m<0\end{cases}$$

*Equation 7-4: Unbiased auto correlation*

The R&S RTP uses only the real part of the signal.

### 7.3.4.8 Math functions: filter and power

*Table 7-10: Filter and power (More keys)*

| Icon | Description | Usage, comment, *FormulaEditor expression* |
|---|---|---|
| | Electric power | Electric power is calculated from voltage, based on measurement impedance<br>(see Impedance)<br>*elecpower(x)* = $U^2/R$ |
| | Finite impulse response (FIR) filter | *FIR(tpye,source,limit,shape)* or *FIR(userdef,source,path)*<br>See Chapter 7.3.5.2, "FIR filter in the formula editor", on page 294 |
| | Type of FIR filter | *highpass* or *lowpass*, see FIR filter |

| Icon | Description | Usage, comment, *FormulaEditor expression* |
|---|---|---|
| | Characteristics of FIR filter | *gaussian* or *rectangle*, see FIR filter |
| | Moving average | Calculates a mean value of several adjacent sample points. The result is a smoothed waveform. The moving average uses the full data and can be used for non-periodic signals. It works like a lowpass filter and increases the vertical resolution at the expense of bandwidth reduction.<br>*MovingAverage(x,y)*<br>with: x = source (channels only), y = number of samples to be averaged<br>Example:<br>*MovingAverage(Ch1Wfm1,1000)*<br>Averages 1000 subsequent samples of the channel 1 waveform |

### 7.3.4.9 Math functions: CDR

The clock data recovery function requires at least one of the options R&S RTP-K12, or R&S RTP-K133, or R&S RTP-K134.

***Table 7-11: Clock data recovery (More keys)***

| Icon | Description | Usage/Comment, *FormulaEditor expression* |
|---|---|---|
| | CDR: displays the generated clock signal as math waveform | Software 1: *CDR(sw1,<x>)*<br>Software 2: *CDR(sw2,<x>)*<br>where <x> is the signal from which the clock is recovered<br>Reference n: *CDR(R<n>)*<br>where <n> = 1 to 4 is a loaded reference file that contains CDR information<br>Example:<br>*CDR(sw1,CH2Wfm1)*<br>See also: Chapter 18.3.2, "Displaying the recovered clock signal", on page 1070 |

### 7.3.4.10 Transfer functions of a step-like signal

The R&S RTP provides two functions to calculate the transfer function of a system out of its step response. These functions are not supported by the formula editor, but you can enter them directly in the "Advanced" tab in the "Math Setup" dialog box. To enter the formula, you need a keyboard (on-screen keyboard or a connected usual one).

| Math expression | Description |
|---|---|
| *Step2FreqRespNormMag(x,points)*<br>Where:<br>• x is the waveform (channel, math, reference)<br>• points are the number of points in the resulting math waveform (min. 1000 points) | Normalized magnitude of transfer function in frequency domain<br>Example:<br>*Step2FreqRespNormMag(Ch1,2000)* |
| *Step2FreqRespNormPhi(x,points,time offset)*<br>Where:<br>• x is the waveform (channel, math, reference)<br>• points are the number of points in the resulting math waveform (min. 1000 points)<br>• Time offset in s | Normalized phase of transfer function in frequency domain<br>Example:<br>*Step2FreqRespNormPhi(Ch1,2000,2e-9)* |

The calculation of transfer functions has several steps:

- Calculate the frequency domain representation of a step-like signal. A detailed description is given in: A. M. Nicolson, „Forming the fast Fourier Transform of a step response in Time-Domain Metrology,“ in Electronic letters, Bd. 9, Nr. 14, p. 317, 1973.
- Calculate the derivative to convert the result to the transfer function.
- Normalize the result at DC to a magnitude of 1 V and subtract specified time offset from phase.

The step-like signal to be analyzed is a time domain waveform, and the result is a math waveform in the frequency domain.

To get useful results, check and adjust the following settings:

1. Place the analyzed step of the signal in the left half of the time domain diagram. Use the "Reference point" or the horizontal "Position" of the trigger to move the signal.
   See also: Chapter 5.4, "Acquire settings", on page 128.
2. Adjust the record lengths of the source signal and the math signal. The "Record length" of the step-like source signal must be shorter than the number of points in the resulting math function, which is set in the formula. See also: "Record length" on page 129.
3. If the resulting math waveform is noisy, increase the "Average count" for the source signal until the waveform is clear.
   See also: Chapter 5.4.2, "Mode settings", on page 131.

#### 7.3.4.11 Defining a formula in the formula editor

1. Open the "Menu" > "Math" dialog.
2. Select the Equation tab.
3. Double-tap the editing area.

   The "Formula Editor" is displayed.

4. Enter the math formula including all required signal sources and operators by selecting the corresponding keys in the editor. For details on the available keys, see Chapter 7.3.4.1, "Formula editor", on page 284.
5. To insert a physical unit in the formula, proceed as follows:
   a) If necessary, insert a decimal prefix using the "M/k/µ" key.
   b) Insert an opening square bracket using the "[" key.
   c) Insert the physical unit using the "V/A/Ω" key.
   d) Insert a closing square bracket using the "]" key.

   The resulting expression could be, for example: `m[V]`
6. To perform a rescaling function, proceed as follows:
   a) Select the rescaling function using the "ax+b" key.
   b) Behind the left bracket, insert the signal source using one of the following keys:
      - "Ch" for a channel
      - "Math" for a math function
      - "Ref" for a reference waveform
      - "Meas" for a measurement
   c) Insert a comma using the "," key.
   d) Insert the "a" value, i.e. the scaling factor, using the number keys.
   e) Insert a comma using the "," key.
   f) Insert the "b" value, i.e. the scaling offset, using the number keys.
   g) Insert the closing bracket using the ")" key.

   The resulting expression could be, for example: `rescale(Ch1Wfm1,3,4)`

## 7.3.5 Filters

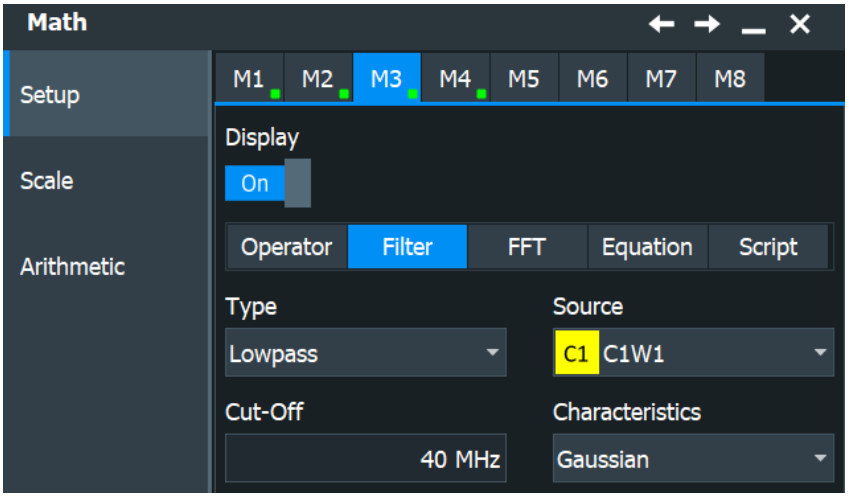
The R&S RTP provides several ways to filter the input signal:

- Lowpass filter by selecting a bandwidth limit in the acquisition path (vertical channel settings)
  Available with probe adapter R&S RT-Z1M, see Chapter 5.6.11.1, "Channel settings with connected adapter", on page 171
- Digital lowpass filter
  See Chapter 5.5.5, "Digital filter setup", on page 145
- FIR filter to create a filtered math waveform (lowpass, highpass, bandpass)
  You can set up a FIR-filtered math waveform in the basic math editor, or in the advanced editor. Both ways are described below in this chapter.

### 7.3.5.1 FIR filter in the math setup

Access: "Math" > "Setup" tab > "Filter" tab.

The finite impulse response filter is a filter to create filtered waveforms with lowpass, highpass, or bandpass. The filter requires additional settings.




- "Type": defines whether the FIR filter is a highpass, lowpass, or bandpass/bandstop (= "User defined") filter.
- "Cut-Off": sets the limit frequency for the highpass or lowpass FIR filter.
- "Characteristics": defines if the filter has a Gaussian or a rectangular shape.
- "Select filter file": relevant for bandpass/bandstop filter. Opens a file dialog to select the file with the filter coefficients.

#### Cutoff frequency for lowpass FIR filter

The cutoff frequency depends on the horizontal resolution and the filter characteristics. The frequency for the lowpass filter can only be set in this range:

*f_g_3dB = (0.001 ... 0.2)* f_a_in* for Gaussian FIR filter

*f_g_3dB = (0,001 ... 0.4)* f_a_in* for rectangular FIR filter

Where: *f_g_3dB* = cutoff frequency to be set for the lowpass filter, and *f_a_in* = reciprocal of the resolution, or sample rate.

#### Cutoff frequency for highpass FIR filter

To check limit frequency for the highpass filter, convert it to an equivalent lowpass frequency:

*f_LP = f_a_in/2 - f_HP*

Where f_HP is the requested highpass limit frequency and f_LP the equivalent lowpass frequency that has to comply with the limits given above.

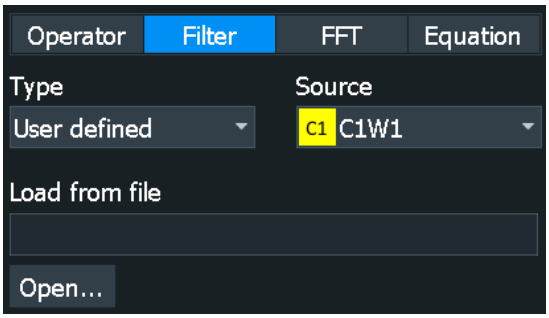
#### Bandpass and bandstop FIR filter: CSV file

To define a bandpass or bandstop, you need a CSV file that contains the comma-separated filter coefficients. The maximum number of filter coefficients is 2 048 000 taps. No other parameters are allowed in the file.

To create the CSV file, we recommend using the Matlab Filter Design & Analysis tool. In the tool, enter the filter type, filter order and filter frequencies. Make sure to set the

sample frequency in MATLAB® and the sample rate at the oscilloscope to the same value. If the values differ, the filter is shifted in frequency.

In R&S RTP, in the "Math" > "Setup" > "Filter", select the filter "Type" = "User defined", and load the CSV file.




### 7.3.5.2 FIR filter in the formula editor

You can type the FIR filter formula directly in the "Equation" tab, or open the formula editor and use the buttons to create the formula.

#### Highpass and lowpass FIR filter

The general syntax for highpass and lowpass filters is:

*FIR(tpye,source,limit,shape)*

- Type is *lowpass* or *highpass*
- Source is the input channel of the signal.
- Limit is the cutoff frequency. Consider the range of possible values, see "Cutoff frequency for lowpass FIR filter" on page 293.
- Shape is *gaussian* or *rectangle* for the lowpass and highpass filter.

► For example, to set a rectangle lowpass filter on channel 1 with 10 MHz cutoff frequency, enter:
*FIR(lowpass,Ch1,1e+07,rectangle)*

► For example, to set a Gaussian highpass filter on channel 2 with 3 GHz cutoff frequency, enter:
*FIR(highpass,Ch2,3e+09,gaussian)*

#### Bandpass and bandstop FIR filter

The syntax for the bandpass filter is:

*FIR(userdef,source,path)*

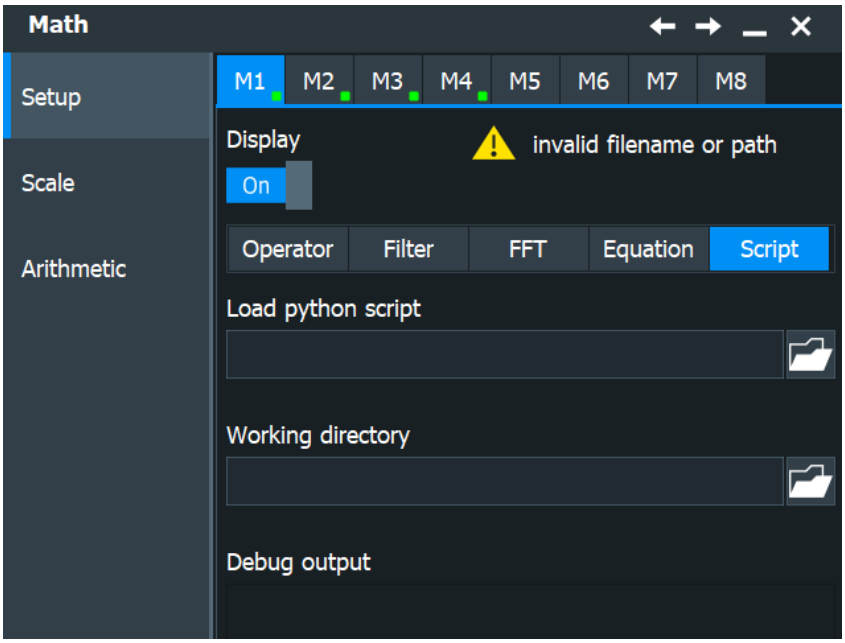
- Source is the input channel of the signal.
- Path is a string containing the path and filename of the filter file. The file contains the comma-separated filter coefficients.

► For example, to set a bandpass on channel 1 with filter coefficients saved in the `bandpass.csv` file, enter:
*FIR(userdef,Ch1,"C:\Users\Public\Documents\Rohde-Schwarz\RTx\bandpass.csv")*

### 7.3.6 Script (option R&S RTP-K39)

The user-defined math option R&S RTP-K39 provides a Python interface to apply Python scripts to the waveform processing. The script replaced the math function and calculates the math waveform, which can be displayed. Scripts can provide solutions that exceed the scope of the math functionality, for example, display the Allen variance of a chirped radar signal to verify the clock stability.

Access: "Math" > "Setup" tab > "Script" tab.




For script-created math waveforms, scaling in the "Scale" tab is not available.

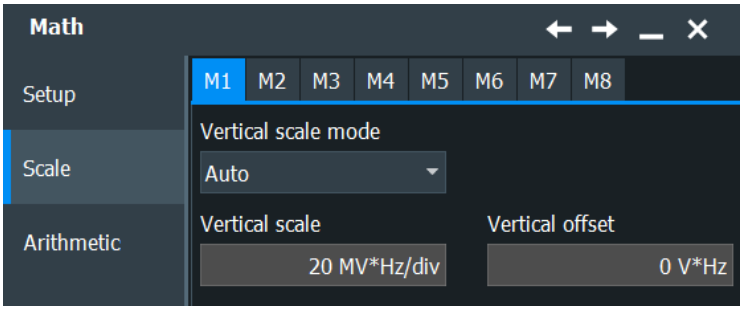
**Load Python script**
Opens a file explorer, where you select the script file to be executed.

**Working directory**
Opens a file explorer, where you select a folder to save result and temporary data.

**Debug output**
Shows information on errors and problems in the script.

### 7.3.7 Math scale settings

Access: [Math] > "Scale" tab.




In this tab, you can find functions to set the vertical parameters of the math waveform.

For FFT scale settings, see Chapter 9.1.4, "FFT scale settings", on page 416.

**Vertical scaling mode (Manual/Auto)**
By default, the vertical scale is adapted to the current measurement results automatically to provide an optimal display. However, if necessary, you can define scaling values manually to suit your requirements.

**Note:** When you change the scaling values manually using the "Scale" rotary knob, the scale mode is set to "Manual" temporarily. When you edit the math function, scaling is automatically set back to "Auto" mode. "Manual" mode is only maintained during math function changes if you select it yourself.

| | |
|---|---|
| "Manual" | Enter the required values for "Vertical scale" and "Vertical offset". For FFT, set "Vertical range" and "Vertical maximum". |
| "Auto" | "Vertical scale" and "Vertical offset" are read-only. For FFT, only the "Vertical maximum" is read-only. |

**Vertical Scale**
Defines the scale of the y-axis in the math function diagram. The value is defined as "<unit> per division", e.g. *50m V/div*. In this case, the horizontal grid lines are displayed in intervals of 50 mV.

If the "Vertical scaling mode (Manual/Auto)" on page 296 is set to "Auto", this setting is read-only.

Remote command:
`CALCulate:MATH<m>:VERTical:SCALe` on page 1403

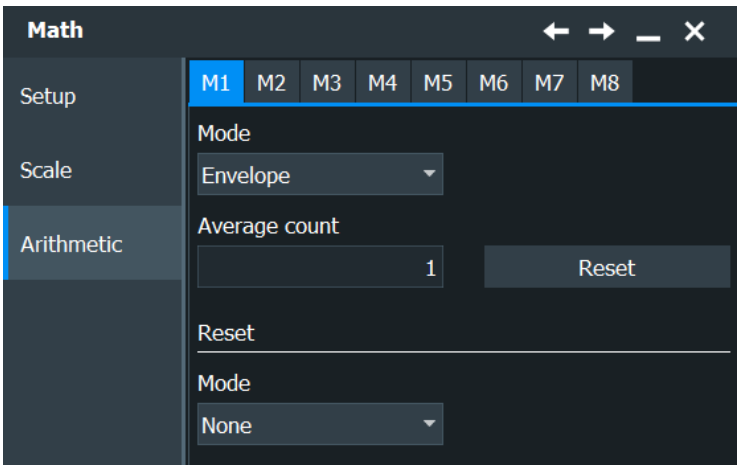
**Vertical Offset**
Sets a voltage offset to adjust the vertical position of the math function on the screen. Negative values move the waveform, positive values move it down.

If the "Vertical scaling mode (Manual/Auto)" on page 296 is set to "Auto", this setting is read-only.

Remote command:
`CALCulate:MATH<m>:VERTical:OFFSet` on page 1402

### 7.3.8 Math arithmetic

Access: [Math] > "Arithmetic" tab.




In this tab you can specify the waveform arithmetic for the math waveforms.

**Mode**
Waveform arithmetic builds the resulting waveform from several consecutive acquisitions and subsequent math calculations of the signal. For details, see "Arithmetic" on page 133.

| | |
|---|---|
| "Original" | The original results are displayed. |
| "Envelope" | The envelope curve of all acquired and calculated results is displayed. |
| "Average" | The average of all acquired and calculated results is displayed. |
| "RMS" | The root mean square of the math data is displayed. The result is the average power spectrum. If you measure the channel power on this RMS spectrum, you get the same result as for the average channel power measurement on waveforms. |
| "MinHold" | Determines the minimum result for each input value from the data of the current acquisition and some acquisitions before. |
| "MaxHold" | Determines the maximum result for each input value from the data of the current acquisition and some acquisitions before. |

Remote command:
`CALCulate:MATH<m>:ARIThmetics` on page 1401

**Average count (N-single count)**
Access:
- "Menu" > "Acquire" > "Setup" tab > "N-single count"
- "Menu" > "Acquire" > "Mode" tab > "Average count"
- "Menu" > "Acquire" > "Segmented" tab > disable "Acquire maximum" > "Required"
- [Math] > "Arithmetic" tab > "Mode" is not "Off" > "Average count"

The acquisition and average count has several effects:
- It sets the number of waveforms acquired with [Single]
- It defines the number of waveforms used to calculate the average waveform. Thus, the instrument acquires sufficient waveforms to calculate the correct average if "Average" is enabled for waveform arithmetic. The higher the value is, the better the noise is reduced.

- It sets the number of acquisitions to be acquired in a fast segmentation acquisition series. Thus, you can acquire exactly one fast segmentation acquisition series with [Single].
  If fast segmentation is enabled and configured to acquire the maximum number of acquisitions, the acquisition count is set to that maximum number and cannot be changed.
  See also "Number of acquisitions" on page 135.
- It is the "Finished" criteria for the state of a mask test.

Remote command:
`ACQuire:COUNt` on page 1245

**Reset**
Forces the immediate restart of the envelope and average calculation for all waveforms.

Remote command:
`ACQuire:ARESet:IMMediate` on page 1245

**Arithmetic Reset > Mode**
Defines when the envelope and average evaluation restarts.

| | |
|---|---|
| "None" | No restart, the number of acquisitions considered by the waveform arithmetics is not limited. |
| "Time" | Restarts the envelope and average calculation after the time defined in "Time". |
| "Waveforms" | Restarts the envelope and average calculation after a number of acquired waveforms defined in "Count". |

Remote command:
`ACQuire:ARESet:MODE` on page 1246
`ACQuire:ARESet:TIME` on page 1246
`ACQuire:ARESet:COUNt` on page 1246

## 7.4 History

The history accesses the data of previous acquisitions and provides them for further analysis.

### 7.4.1 About history

If a continuous acquisition runs, the captured data is stored in the sample memory and the current acquisition is processed and shown on the display. After the acquisition was stopped, the history accesses the captured samples that were stored, displays these samples as history waveforms, and makes them available for further analysis. It considers all channels that were enabled during the running acquisition. When a new acquisition is started with [Run / Stop] or [Single], the memory is cleared and written anew.

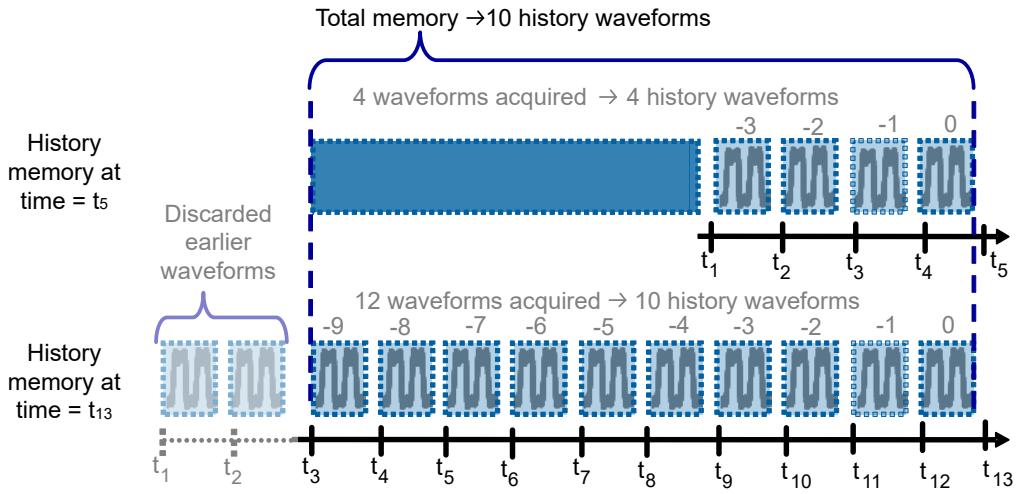



*Figure 7-5: History memory. In this example, the memory can store 10 waveforms.*

You can work with history waveforms in the same way as with the waveform of the latest acquisition: use zoom, cursor measurements, and automatic measurements, create math waveforms, perform mask testing and so on. Saving the history data is also possible, either completely or a part of the data.

The number of stored history waveforms depends on the memory size, the number of enabled channels, and the record length. The shorter the record length, the less the number of channels, and the larger the memory, the more history waveforms are saved.

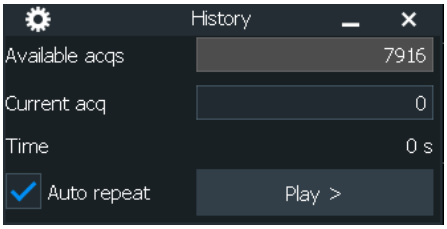
### Memory extension

The memory can be enhanced with options R&S RTP-B101/102/105/110/120. For details, refer to the "Specifications" data sheet.

### Quick-access History dialog box

When you press the [History] key on the front panel or tap "Display" menu > "Show history", the history mode is enabled and the quick-access "History" dialog box is displayed. A running acquistion stops immediately.

The small quick-access "History" dialog box can remain visible on the screen during history replay, so that the history can be replayed at any time by a simple tap on the "Play" button. Closing the quick-access "History" dialog box, or starting a new acquisition disables the history mode.



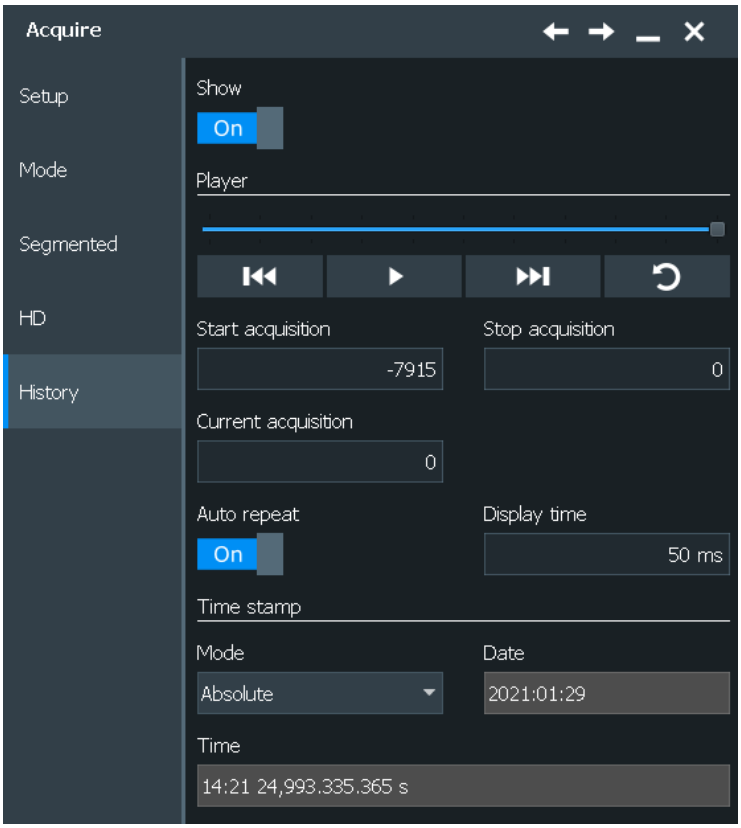
**Export of history waveforms**

## 7.4.2 History setup

The "History" dialog contains the complete functionality on history viewing and information.

The most important information and functions are also provided in the quick-access history dialog box.

Access: "Menu" > "Acquire" > "History" tab.




**Show history / Export history**
Enables the history mode and allows you to save history waveforms to file.

The history display is enabled automatically when you press the [History] button. It is disabled when you close the quick-access "History" dialog box.

For details on data export, see Chapter 12.2.2.2, "Waveform history settings", on page 494.

Remote command:
`CHANnel<m>[:WAVeform<n>]:HISTory[:STATe]` on page 1405

**Player**
The player can be used to control the playback of the history waveforms.

| | |
|---|---|
| ⏮ | Sets the oldest acquisition in the sample memory as "Start acquisition" and "Current acquisition". |
| ▶ | Starts and stops the replay of the history waveforms from "Start acquisition" to "Stop acquisition". |
| ⏭ | Sets the newest acquisition in the sample memory as "Stop acquisition" and "Current acquisition". The newest acquisition always has the index "0". |
| ↺ | Resets the "Stop acquisition" to 0 and the "Start acquisition" to the number of saved acqusitions. Thus, the history viewer shows all acquisitions. |

**Start acquisition**
Sets the index of the first (oldest) acquisition to be displayed or exported. The index is always negative.

Remote command:
`CHANnel<m>[:WAVeform<n>]:HISTory:STARt` on page 1406

**Stop acquisition**
Sets the index of the last (newest) acquisition to be displayed or exported. The newest acquisition of the complete acquisition series always has the index "0".

Remote command:
`CHANnel<m>[:WAVeform<n>]:HISTory:STOP` on page 1407

**Current acquisition**
Accesses a particular acquisition in the memory to display it, or to save it. The newest acquisition always has the index "0". Older acquisition have a negative index.

If a history replay is running, the field shows the number of the currently shown acquisition.

Remote command:
`CHANnel<m>[:WAVeform<n>]:HISTory:CURRent` on page 1406

**Auto repeat**
If selected, the replay of the history waveform sequence repeats automatically. Otherwise, the replay stops at the "Stop acquisition".

Remote command:
`CHANnel<m>[:WAVeform<n>]:HISTory:REPLay` on page 1408

**Display time**
Sets the display time for one acquisition. The shorter the time, the faster the replay is.

The setting takes effect for history replay and the display of a Fast Segmentation series, see Chapter 5.4.3, "Segmented settings", on page 134.

Remote command:
`CHANnel<m>[:WAVeform<n>]:HISTory:TPACq` on page 1407

**Time stamp**
The time stamp shows the time of the currently displayed history acquisition. Thus, the time relation between acquisitions is always available.

The time stamp "Mode" can be absolute or relative:
- In "Absolute" mode, the instrument shows the date and the daytime of the current acquisition.
- In "Relative" mode, the time difference to the newest acquisition (index = 0) is shown.

The time stamp can be included in waveform data export, see"Time stamps" on page 495.

During history replay, the time value is displayed and updated if the replay speed ("Display time") is slow enough, that is 40 ms or slower.

The quick-access history dialog box always shows the relative time. In the "History Viewer" tab, you can select the time mode.

Remote command:
`CHANnel<m>[:WAVeform<n>]:HISTory:TSDate?` on page 1408
`CHANnel<m>[:WAVeform<n>]:HISTory:TSABsolute?` on page 1409
`CHANnel<m>[:WAVeform<n>]:HISTory:TSRelative?` on page 1409
`CHANnel<m>[:WAVeform<n>]:HISTory:TSRReference?` on page 1410

## 7.4.3 Using history

You can access the history waveforms in two ways:
- Display a particular acquisition.
- Replay all or a part of the saved waveforms to track the signal run.

Furthermore, you can export history data to a file.


- "To open the history and get information" on page 302
- "To display a particular acquisition" on page 303
- "To replay history waveforms" on page 303
- "To exit the history" on page 303
- "To save the history data" on page 304


**To open the history and get information**

1. Press the [History] key on the front panel.

   A running acquisition is stopped, the history mode is enabled and the quick-access "History" dialog box is displayed. The [History] key lights up as long as the history mode is active.

2. Open the full configuration dialog box:
   - Tap the icon.
   - Open "Menu" > "Acquire" dialog > "History" tab.

**To display a particular acquisition**

1. In the quick-access "History" dialog box, enter the number of the required acquisition in the "Current acq" field. The newest acquisition always has the index "0", older acquisitions have a negative index.
2. Tap "Play" to start.

Alternatively, you can configure and start the history display from the "History" configuration dialog box:

1. Open the "History" configuration tab.
2. If the history mode is off (the [History] key is not illuminated), select "Show".

   The quick-access dialog box is displayed.
3. Drag the slider to the required acquisition. The current number is shown in the "Current acquisition" field.
   Alternatively, enter the number of the required acquisition in the "Current acquisition" field.
4. Tap "Play" to start.

**To replay history waveforms**

If you want to see the complete acquisition series without any setup, simply tap "Play" in the quick-access "History" dialog box. For specific analysis of history data, use the history "Viewer" setup.

1. Open the "History" configuration dialog tab.
2. If the history mode is off (the [History] key is not illuminated), enable "Show".

   The quick-access dialog box is displayed.
3. Define the part of the history you want to see by doing one of the following:
   - Tap ↺ to see the complete history.
   - Enter the "Start acquisition" of the oldest acquisition to display and the "Stop acquisition" of the newest acquisition to display. All waveforms between the two indexes are displayed.
     To enter the oldest or newest acquisition for either index, tap the appropriate button. The newest acquisition always has the index "0". The "Start acquisition" is always negative.
4. Tap ▸ to start.

**To exit the history**

► Choose one of the following ways:
- Close the quick-access "History" dialog box.
- In the "History" configuration tab, disable "Show".
- Start the acquisition.

**To save the history data**

You can save the complete history, or some subsequent waveforms from the history, or a single history waveform. You can also decide to save the complete waveforms, or a part of each waveform.

1. Open the "Menu" > "Save/Recall" dialog.
2. Select the "Save/Recall" tab.
3. In the "Save " tab, press "Waveform".
4. In the "Setup" tab, tap the source icon to select the waveform you want to save.
5. If you want to save only a part of each waveform, set the "Scope".
   For settings, see "Scope" on page 492.
6. Select the "History" tab.
7. Enable "Export history".
8. If you want to write the timestamps into the data file, enable "Time stamps".
9. To save one waveform out of the history memory:
   a) Make sure that "Multiple acquisitions in one file" is disabled.
   b) Enter the number of the required acquisition in "Acquistion index".
   The newest acquisition in the memory always has the index "0". Older acquisitions have a negative index.
   c) Tap "Save" or "Save As" to save the waveform data to the specified file.
10. To save several subsequent history waveforms:
    a) Enable "Multiple acquisitions in one file".
    b) Define the range of the waveforms to be saved with "Start acquisition" and "Stop acquisition".
    c) Tap "Start export" to play the history and to save the history data to the specified file.

Se also Chapter 12.2.2.2, "Waveform history settings", on page 494.

## 7.5 XY-diagram

XY-diagrams combine the voltage levels of two waveforms in one diagram. They use the voltage level of a second waveform as the x-axis, rather than a time base. This allows you to perform phase shift measurements, for example. You can display up to four different XY-diagrams.

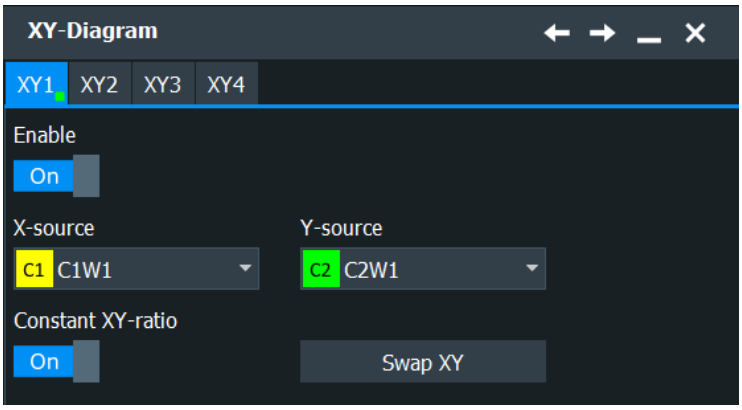
XY-diagrams can be used to display the IQ representation of a signal.

### 7.5.1 Settings for XY-diagrams

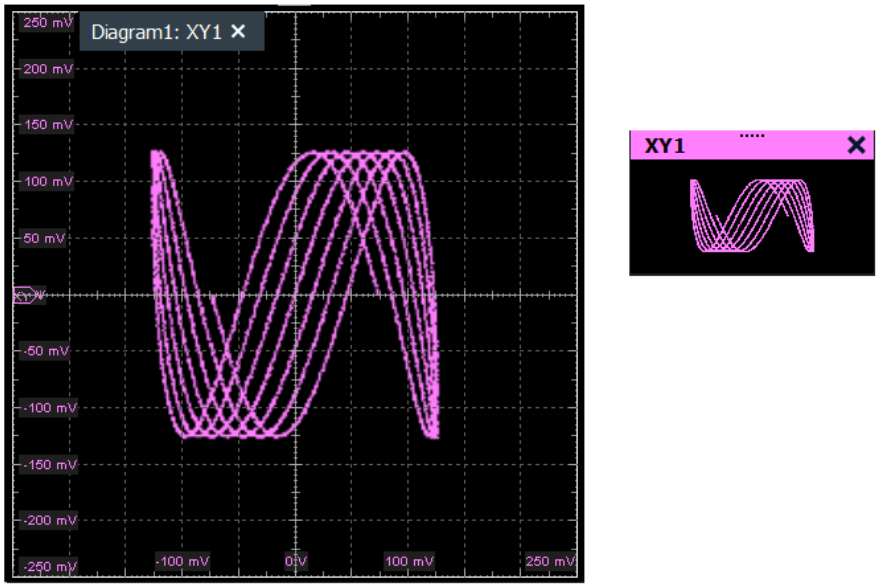
Access: [Apps]>"Analysis" tab > "XY-Diagram"

You can display up to four different XY-diagrams that use the voltage level of a waveform as the x-axis, rather than a time base.

Make sure to select the tab of the required XY-diagram.




**Enable**
If activated, the XY-waveform is active and shown in a diagram, or it is minimized in a signal icon.




Remote command:
`WAVeform<m>:XYCurve:STATe` on page 1411

**X-source**
Defines the signal source that supplies the x-values of the XY-diagram. Select one of the following:

- One of the waveforms of any channel
- A reference waveform

- The results of a mathematical function

Remote command:
WAVeform<m>:XYCurve:XSOurce on page 1412

**Y-source**
Defines the source to be used as the y-axis of the XY-diagram. Select one of the following:

- One of the waveforms of any channel
- A reference waveform
- The results of a mathematical function

Remote command:
WAVeform<m>:XYCurve:YSOurce on page 1412

**Constant XY-ratio**
If enabled, the x- and y-axes maintain a constant ratio in the diagram.

Remote command:
WAVeform<m>:XYCurve:RATio on page 1410

**Swap XY**
Replaces the source of the x-axis with the source of the y-axis and vice versa.

Remote command:
WAVeform<m>:XYCurve:SWAP on page 1411

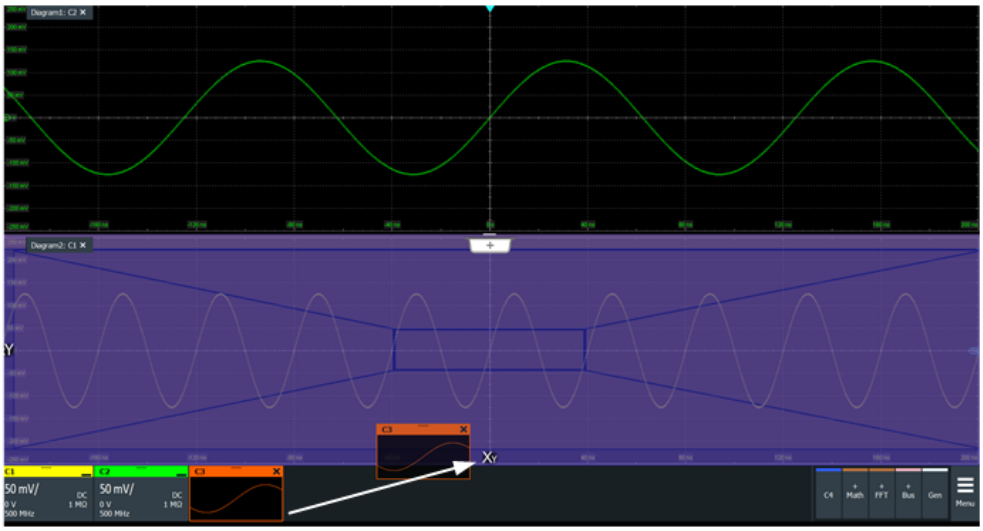
### 7.5.2 Displaying an XY-diagram

You can create the diagram from active waveforms with drag&drop, or use the dialog box for setup.

**To display an XY-diagram with drag&drop**

Prerequisites: The source waveform for the y-axis is active in a diagram, the source waveform for the x-axis is either active or minimized.

1. Drag the x-axis waveform to the lower middle of the diagram with the y-axis waveform.
2. Drop the icon when it overlaps with the Xy shown in the middle of the diagram.



The diagram is converted into an XY-diagram.

**To set up an XY-diagram**

1. Press the [Apps] key.
2. In the >"Analysis" tab, tap the "XY-Diagram".
3. Enable the "XY-diagram".
4. In the "X-source" field, define the signal source that supplies the x-values of the XY-diagram. Select one of the following:
   - One of the waveforms of any channel
   - A reference waveform
   - The results of a mathematical function
5. In the "Y-source" field, define the signal source that supplies the y values of the XY-diagram.
6. To switch the x- and y-values quickly, tap the "Swap XY" button.
7. To maintain a constant ratio while the x- and y-axes are adapted to the acquired data dynamically, activate the "Constant XY-ratio" option.

If the XY-diagram is active or minimized, touch and hold the signal icon to open the "XY-diagram" tab.

# 8 Measurements

Using the R&S RTP you can perform and display different measurements simultaneously, based on the active signal or math waveforms. The color of the results in the result table corresponds with the source waveform color.

The following measurement methods are available:

- Cursor measurements: measurements can be configured for up to 4 cursor sets to determine specific results at the manually defined cursor positions of an active waveform; the results are displayed in a result box.
- Automatic measurements: up to eight measurements can be configured and performed simultaneously on different sources; the results of each measurement are displayed in a result box.
- Quick measurements: performs a set of automatic measurements on the selected waveform at the push of a button. You can configure the set of measurements.

## 8.1 Cursor measurements

- Cursors and results of cursor measurements ..... 308
- Using cursors ..... 309
- Settings for cursor measurements ..... 312

### 8.1.1 Cursors and results of cursor measurements

Cursor measurements determine the results at the current cursor positions. The cursors can be positioned manually, or can be configured to follow the waveform. You can measure on one waveform, or on two different waveforms (sources).

Up to 4 cursor sets can be configured and displayed. Each cursor set consists of a pair of horizontal or vertical cursors, or both. Cursor lines can be coupled so that the initially defined distance is always maintained.

The cursors are displayed in the diagrams of the source waveform only, or in all diagrams. For each measurement, labels can be defined for the cursors. By default, the cursors are labeled as C1.1, C1.2, C2.1, C2.2, C3.1, C3.2, C4.1, C4.2.

How to set up cursor measurements is described in Chapter 8.1.2, "Using cursors", on page 309. The Chapter 8.1.3, "Settings for cursor measurements", on page 312 provides a detailed description of all settings.

Cursors can also define a gate to limit the measurement to the section of the waveform between the cursor lines. See Chapter 8.2.3.2, "Gate settings for measurements", on page 329.

The result display of cursor measurements is configurable. Results can be shown in a docked table, or in floating result boxes for each measurement. Similar to waveform diagrams, you can also minimize the result box to a result icon on the signal bar.

For details on using the result box, see Chapter 3.3.7, "Displaying results", on page 61 and "Result position" on page 88.

#### 8.1.1.1 Cursor measurements on time-based waveforms

The cursor for measurement on time-based waveforms returns the following results. The results are displayed automatically when a cursor measurement is enabled.

***Figure 8-1: Measurement results of a cursor measurement in time domain***

| Label | Description |
|---|---|
| X1, X2 | Time at the position of the vertical cursors. |
| Y1, Y2 | Vertical values of the waveform at the position of the horizontal cursors in V or A. |
| ΔX | Difference between the vertical cursor (time) values |
| ΔY | Difference between the horizontal cursor values |
| 1/ΔX | Inverse time difference |
| ΔY/ΔX | Slope of the waveform between the cursors (if measured on one source) |

#### 8.1.1.2 Cursor measurements on spectrum waveforms

If the measurement source is a spectrum waveform, the results have a different meaning. Measurement on 2 spectrum waveforms is not possible.

The result box for measurement on spectrum waveforms shows the following information.

| Label | Description |
|---|---|
| X1, X2 | Frequency at the position of the vertical cursors |
| Y1, Y2 | Vertical values of the waveform at the position of the horizontal cursors in dB. |
| Bw | Difference between the vertical cursor (frequency) values |
| ΔY | Difference between the horizontal cursor values |

To set the cursor lines to the peaks, various functions are available in the "Peak Search" tab, see Chapter 8.1.3.3, "Peak Search tab", on page 317.

### 8.1.2 Using cursors

You can start cursor measurements by using the "Cursor" icon on the toolbar, or using the [Cursor] key. For detailed configuration, use the "Cursor" dialog box.

- Starting a simple cursor measurement..... 310
- Configuring a cursor measurement.....310
- Configuring the cursor display.....312
- Saving measurement results to file.....312


#### 8.1.2.1 Starting a simple cursor measurement

**To display cursors using the toolbar**

1. Tap the "Cursor" icon on the toolbar.

2. Tap the waveform that you want to measure. Alternatively, you can draw a rectangle in the diagram to position the cursor lines.

   The cursor lines appear and the cursor results are displayed in a table or result box.

**To display cursors using the [Cursor] key**

1. Select the waveform that you want to measure.
2. Press the [Cursor] key.

   The cursor lines and the measurement results are displayed.

**To disable one cursor measurement**

1. Tap the "Delete" icon on the toolbar.
2. Tap a cursor line.

   The cursor set and its measurement results are deleted.

**To disable all cursor measurements**

1. Tap the "Delete" icon on the toolbar.
2. Tap the results table.

   All cursor sets and measurement results are deleted.

#### 8.1.2.2 Configuring a cursor measurement

To modify the position of the cursor lines, you can drag the lines on the screen. In addition, various settings are possible to refine the measurement.

The complete configuration of cursor measurements is provided in the "Cursors" dialog.

1. To open the "Cursors" dialog, use one of these ways:
   - Open the "Menu", and select "Cursor".

- Press the [Cursor] key.
- Use the overlay menu:
  - Double-tap a measurement result in the results table.
  - Adjust source, type, and positions of the cursor lines in the overlay menu.
  - Tap "Advanced Setup" to open the "Cursors" dialog.
- If the results are shown in a floating result box, tap the ⚙ icon in the result box.

2. Select the "Setup" tab.
3. Select the subtab for the cursor set that you want to use.
4. Select the source - the measured waveform. You can select any input channel, or various other active waveforms. Available sources are shown in the source list.
5. If necessary, enable and select a 2nd waveform.
6. Select the cursor type: horizontal, vertical, or both.
7. Define the position of the cursors:
   a) To define exact positions of the cursor lines, enter the X-position for each vertical cursor and the Y-position for each horizontal cursor. If it is not possible to set horizontal cursors, disable "Track waveform".
   b) To position the horizontal cursors automatically, select "Track waveform".
   In this case, cursor 1 indicates the current maximum, cursor 2 indicates the current minimum. If both horizontal and vertical cursors are displayed, the horizontal cursors are placed at the crossing points of the vertical cursors with the waveform. Adjust the vertical cursors manually, and the horizontal cursors follow.
   c) If the waveform arithmetics are set to "Envelope", and "Track waveform" is active, select which horizontal cursor is positioned to the maximum and which to the minimum envelope values.
   d) To keep the distance between the vertical cursors when one cursor is moved, select "Coupling (2 follows 1)".
8. Optionally, select "Show in all diagrams" in the "Advanced" tab. This setting enables the cursor display in all diagrams that are in the same domain as the selected source (time or spectrum).
9. To set the cursors for a spectrum measurement to peak values, select the "Peak Search" tab.
   Tap one of the buttons to place the cursors on the selected peak value. For details, see Chapter 8.1.3.3, "Peak Search tab", on page 317.
10. Tap the "Cursor On/Off" button in the "Setup" tab to activate the cursor measurement.

    The cursors lines and the results are displayed. For details on cursor measurement results, see Chapter 8.1.1, "Cursors and results of cursor measurements", on page 308.

#### 8.1.2.3 Configuring the cursor display

By default, the cursors are displayed as lines in the diagrams and labeled according to the syntax: Cu<cursor set number>.<1|2>

For example, the cursors for the cursor set 2 are labeled 2.1 and 2.2. The horizontal and the vertical cursors lines have the same labels.

You can change the default cursor display and labels.

**To set the cursor style**

1. Open the "Menu" and select "Settings".
2. Select "Appearance".
3. Select the "Cursor" tab.
4. Select the cursor style. See also: "Cursor style" on page 88.

**To add labels to cursor lines**

1. Open the "Cursors" dialog box, for example, with "Menu" > "Cursor".
2. Select the subtab for the cursor set you want to configure.
3. Select the "Advanced" tab.
4. For each vertical and horizontal cursor, enter a label.
5. Select "Show labels".

#### 8.1.2.4 Saving measurement results to file

1. Open the "Menu" and select "Save/Recall".
2. Select the "Save" tab.
3. Select "Results".
4. Select the results that you want to export.
5. Save the file.

### 8.1.3 Settings for cursor measurements

Cursor measurements are configured in the "Cursors" dialog box.

#### 8.1.3.1 Setup tab

The "Setup" tab contains the settings for cursor measurements.

The cursor style and position of the measurement results is defined in the "Settings" > "Appeareance" > "Cursor" dialog box.

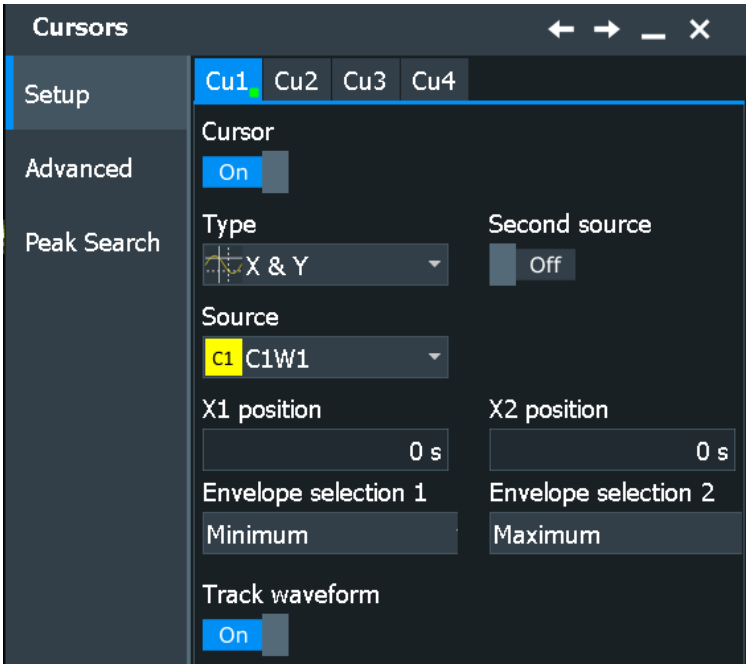



*Figure 8-2: Setup for cursor on 1 source, with waveform tracking*

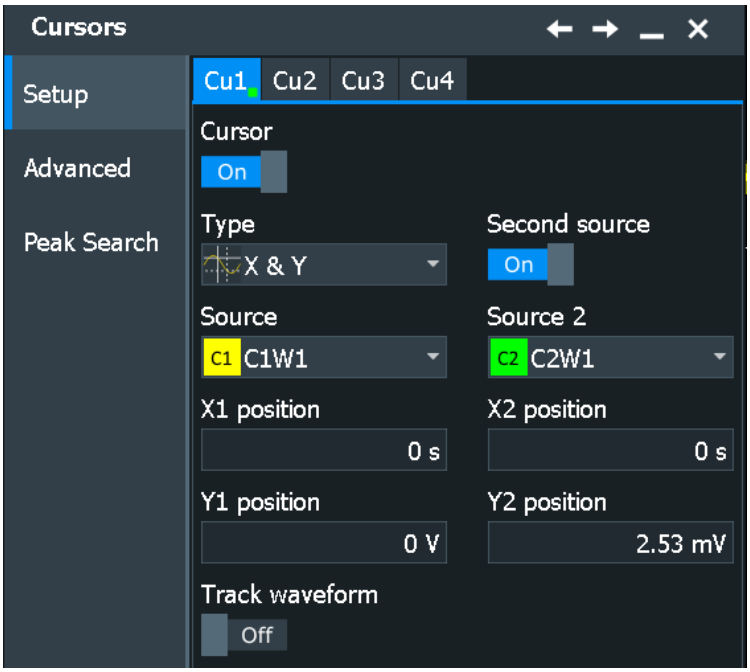



*Figure 8-3: Setup for cursor on 2 sources*

**Cu1, Cu2, Cu3, Cu4**
The settings for each cursor measurement (or cursor set) are configured on separate tabs. For each cursor set, a horizontal pair of cursors, a vertical pair of cursors, or both can be displayed.

**Cursor On/Off**
Enables the selected cursor measurement.

Remote command:
CURSor<m>:STATe on page 1414

**Type**
Defines the cursor type to be used for the measurement.

- Horizontal cursors (Y-cursors)
  Horizontal cursors are positioned automatically along the waveform and can be adjusted manually
- Vertical cursors (X-cursors)
  Both vertical cursor lines are set automatically to the trigger position, and you can reposition them manually.
- Both vertical and horizontal cursors (X&Y-cursors)
  Horizontal cursors are positioned automatically along the waveform and vertical cursors are set to the trigger position. You can reposition all cursor lines manually.

Remote command:
CURSor<m>:FUNCtion on page 1414

**Source**
Defines the source of the cursor measurement. Any of the input signal, math, reference or XY waveforms can be selected.

Remote command:
CURSor<m>:SOURce on page 1415

**Second source, Source 2**
Enables and selects a second source for the cursor measurements. If enabled, the second cursor lines Cx.2 measure on the second source. Using a second source, you can measure differences between two channels with cursors.

Remote command:
CURSor<m>:USSource on page 1417
CURSor<m>:SSOurce on page 1416

**X1 position, X2 position**
Defines the position of the vertical cursors.

Remote command:
CURSor<m>:X1Position on page 1418
CURSor<m>:X2Position on page 1418

**Y1 position, Y2 position**
Defines the position of the horizontal cursor lines. The setting corresponds to the V1 and V2 values in the "Cursor Results" box.

If "Track waveform" is enabled, the user setting is disabled and the measurement results are displayed in the "Cursor Results" box.

Remote command:
CURSor<m>:Y1Position on page 1419
CURSor<m>:Y2Position on page 1420

**Envelope selection 1|2**
Envelope selection is effective under the following conditions:

- The waveform arithmetic of the cursor source waveform is set to envelope waveform (see "Arithmetic" on page 133)
- "Track waveform" is enabled.
- Both horizontal and vertical cursors are enabled ("Type" = *Both*).

The setting defines which horizontal cursor is positioned to the maximum and which to the minimum envelope values.

"Minimum" The horizontal cursor is set to the crossing point of the vertical cursor with the minimum waveform envelope.

"Maximum" The horizontal cursor is set to the crossing point of the vertical cursor with the maximum waveform envelope.

Remote command:
`CURSor<m>:X1ENvelope` on page 1420
`CURSor<m>:X2ENvelope` on page 1421

**Track waveform**
The horizontal cursors track the waveform. The first cursor line indicates the current vertical minimum, and the second cursor line indicates the maximum. If the waveform changes, e.g. during a running measurement, the cursors move along with it. If both horizontal and vertical cursors are displayed, the horizontal cursors are positioned to the crossing points of the vertical cursors with the waveform. The measurement results are displayed in the "Cursor Results" box.

Tracking disables the Y-coupling (coupling of horizontal cursor lines) and the Y user position settings.

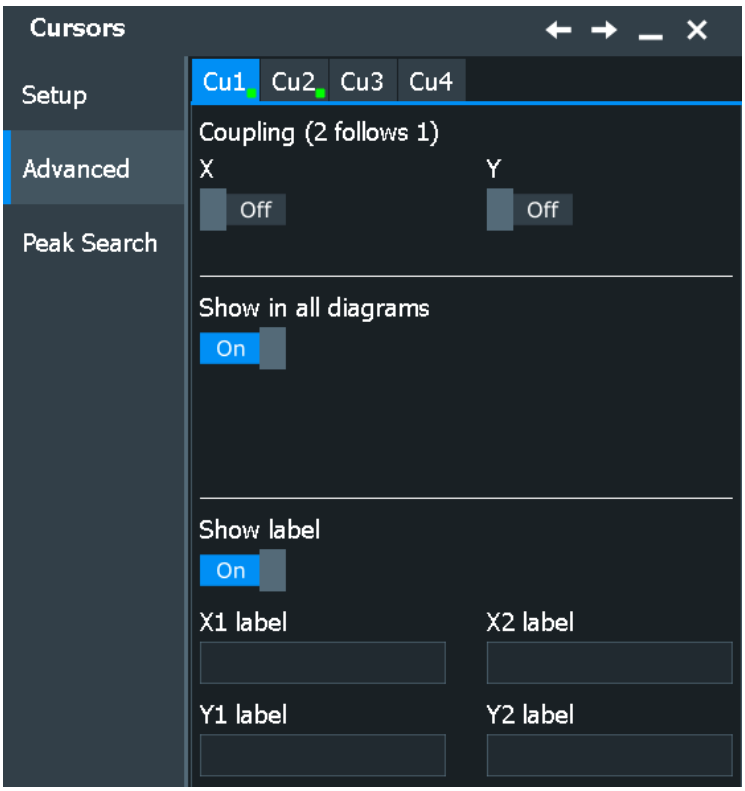
Remote command:
`CURSor<m>:TRACking[:STATe]` on page 1418

### 8.1.3.2 Advanced settings

Access: [Cursor] key > "Advanced" tab

The settings in the "Advanced" cursor tab configure the behavior and display of cursor lines, and labels for the lines.

The cursor style and position of the measurement results is defined in the "Settings" > "Appeareance" > "Cursor" dialog box. See Chapter 4.3.7, "Cursor appearance settings", on page 88.




**Cu1, Cu2, Cu3, Cu4**
The settings for each cursor measurement (or cursor set) are configured on separate tabs. For each cursor set, a horizontal pair of cursors, a vertical pair of cursors, or both can be displayed.

**Coupling (2 follows 1)**
Couple the horizontal and vertical cursor lines so that the distance between the two lines remains the same if one cursor is moved.

Remote command:
`CURSor<m>:YCOupling` on page 1420
`CURSor<m>:XCOupling` on page 1419

**Show in all diagrams**
Shows the enabled cursor measurements in all active diagrams of the same (time/ spectrum) domain.

Remote command:
`CURSor<m>:SIAD` on page 1422

**Show label**
Shows the cursor labels in the diagram.

Remote command:
`CURSor<m>:LABel[:LABel]` on page 1421

**X1 label, X2 label**
Defines a label to be displayed with the vertical cursors.

By default, the cursors are labeled as Cu1.X1, Cu1.X2, Cu2.X1, Cu2.X2, Cu3.X1, Cu3.X2, Cu4.X1, Cu4.X2.

Remote command:
`CURSor<m>:LABel:X<1..2>` on page 1422

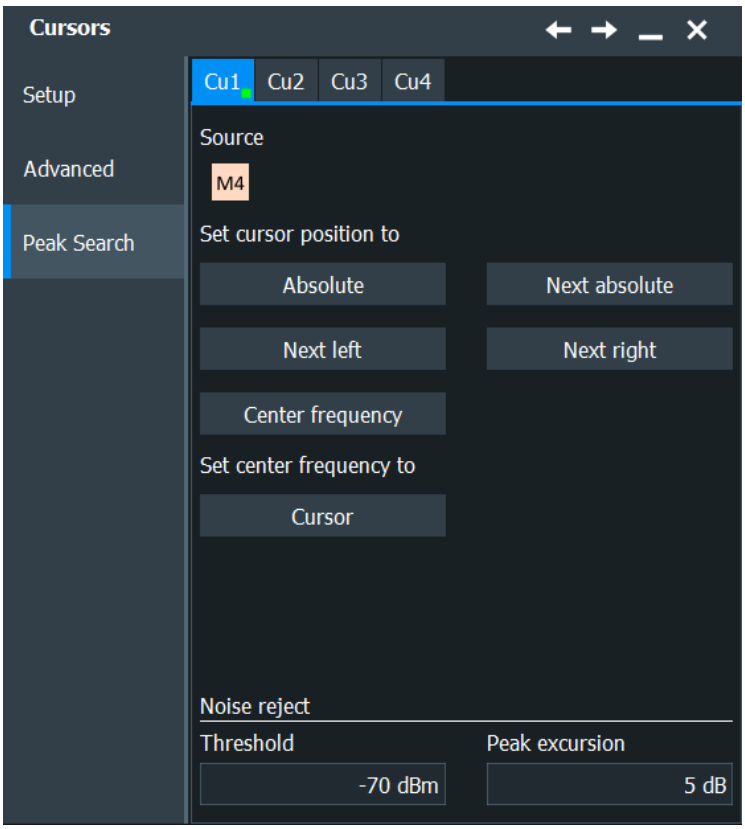
**Y1 label, Y2 label**
Defines a label to be displayed with the horizontal cursors. By default, these cursors are labeled as Cu1.Y1, Cu1.Y2, Cu2.Y1, Cu2.Y2, Cu3.Y1, Cu3.Y2, Cu4.Y1, Cu4.Y2.

Remote command:
`CURSor<m>:LABel:Y<1..2>` on page 1422

### 8.1.3.3 Peak Search tab

Access: [Cursor] key > "Peak Search" tab

The settings on this tab are only available in spectrum mode, i.e. the source of the cursor measurement is an FFT math waveform. In this case, the cursors can indicate the results of a peak search on the waveform. You can define which peaks the instrument determines by defining the noise reject settings.



**Cu1, Cu2, Cu3, Cu4**
The settings for each cursor measurement (or cursor set) are configured on separate tabs. For each cursor set, a horizontal pair of cursors, a vertical pair of cursors, or both can be displayed.

**Source**
Defines the source of the cursor measurement. Any of the input signal, math, reference or XY waveforms can be selected.

Remote command:
`CURSor<m>:SOURce` on page 1415

**Absolute**
Both cursors are set to the absolute peak value.

Remote command:
`CURSor<m>:MAXimum[:PEAK]` on page 1425

**Next absolute**
Cursor 2 is set to the next smaller absolute peak from the current position.

Remote command:
`CURSor<m>:MAXimum:NEXT` on page 1425

**Next left**
Cursor 2 is set to the next peak to the left of the current position.

Remote command:
`CURSor<m>:MAXimum:LEFT` on page 1425

**Next right**
Cursor 2 is set to the next peak to the right of the current position.

Remote command:
`CURSor<m>:MAXimum:RIGHt` on page 1425

**Center frequency**
Sets the vertical cursor line Cu1 to the center frequency.

Remote command:
`CURSor<m>:FFT:TOCenter` on page 1424

**Set center frequency to Cursor**
Sets the center frequency to the frequency value that is measured at cursor line Cu1.

Remote command:
`CURSor<m>:FFT:SETCenter` on page 1424

**Threshold**
Defines an absolute threshold as an additional condition for the peak search. Only peaks that exceed the threshold are detected.

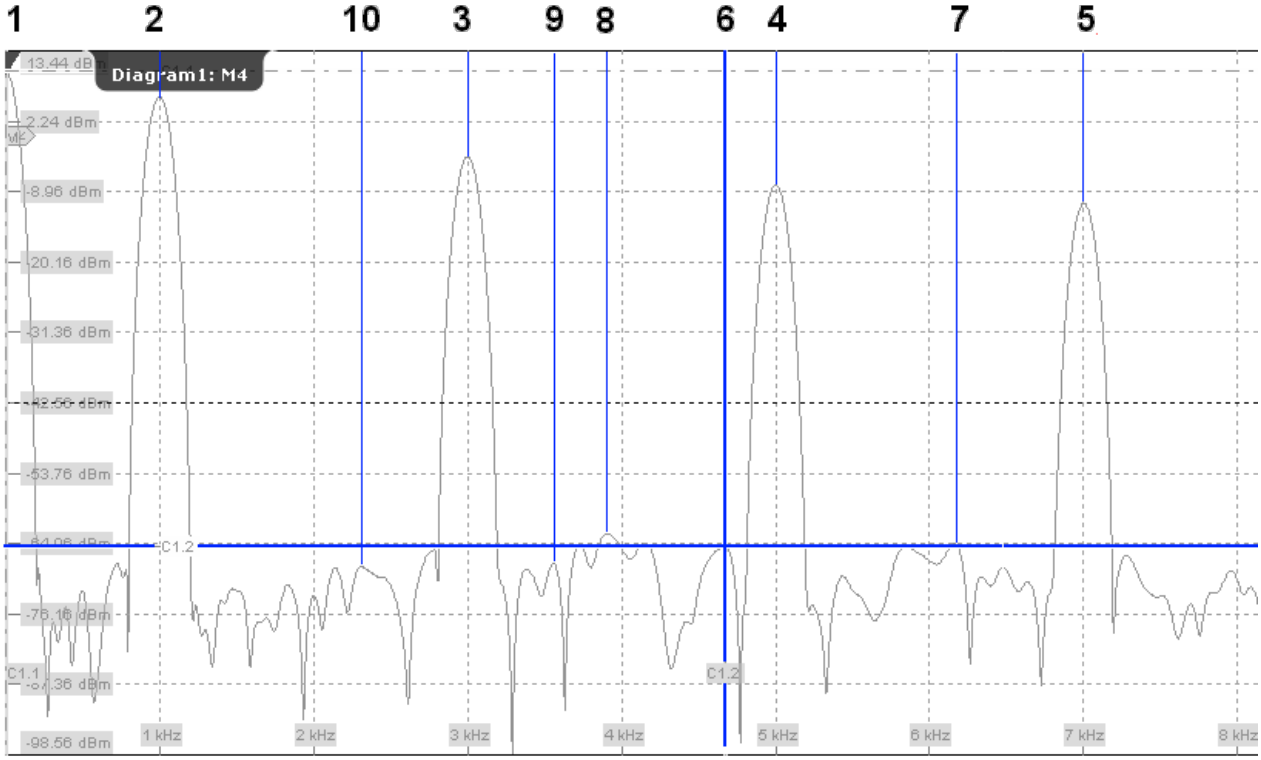
This setting is only available for spectrum waveforms. It is valid for cursor measurements, spectrum measurements and peak search.

Remote command:
CURSor<m>:THReshold on page 1594
MEASurement<m>:SPECtrum:ATHReshold on page 1469

**Peak excursion**
Defines a relative threshold, the minimum level value by which the waveform must rise or fall to be considered as a peak. To avoid identifying noise peaks, enter a peak excursion value that is higher than the noise levels.

This setting is only available for spectrum waveforms. It is valid for cursor measurements, spectrum measurements and peak search.

The following figure shows a cursor measurement on a spectrum waveform:




If "Peak excursion" is 30 dB, the peaks 1 to 5 are found. If "Peak excursion" is 20 dB, also the peaks 6 to 10 are found. The cursor position is on peak 6.

Remote command:
CURSor<m>:PEXCursion on page 1594
MEASurement<m>:SPECtrum:PEXCursion on page 1468

## 8.2 Automatic measurements

The R&S RTP can perform many measurements in parallel. The measurements are combined in up to 8 measurement groups. In addition, you can run the quick measurement.

The basic measurement settings are source, category, and the selection of the measurement. You can refine the setup to get more specific results:

- **Gating**
  A gate limits the measurement to a user-defined part of the waveform. See Chapter 8.2.3, "Measurement gates", on page 329.
- **Statistics and long term measurements**
  To evaluate time-dependent behavior of measurement results, you can use statistics, long term measurements, and tracks. You can also decide, how many measurement results per acquisition contribute to the calculation. See: Chapter 8.2.12, "Result analysis", on page 384.
- **Limit checks and actions on test result**
  Limit and margin checks evaluate if the measurement result exceeds a specified value. You can define actions that are performed on limit or margin violation. See: Chapter 8.2.13, "Limit and margin checks", on page 392.

### Measurement categories

The various measurement are grouped in several categories. The category defines which sources can be analyzed.

Time domain:

- Amp/Time: amplitude and time measurements
- Eye: eye diagram measurements
- Histogram: measurements on histograms
- Jitter measurements (available if one of the jitter options is installed)
- Protocol: advanced analysis (available if option R&S RTP-K35 is installed)

Frequency domain

- Spectrum: measurements on spectrum waveforms
- Histogram: measurements on histograms

Details on automatic measurements are described in the following chapters:

- Measurement setup in general....321
- Measurement results....325
- Measurement gates....329
- Reference levels....332
- Amplitude/time measurements....341
- Eye diagram measurements....355
- Spectrum measurements....363
- Histograms and histogram measurements....368
- Jitter measurements....376
- Optical measurements....376
- Protocol measurements (option R&S RTP-K35)....378
- Result analysis....384
- Limit and margin checks....392

## 8.2.1 Measurement setup in general

Up to 8 measurement groups can be defined. Each measurement group is configured in its own subtab. For each measurement group, the category and the source are defined in the "Meas Group" tab, and you can also enable statistic evaluation. Available sources and measurements depend on the selected category, for example, histogram measurements need a defined histogram as source. For each measurement group, you select all required measurements. If further settings are available for a measurement, a settings icon is shown beside the measurement's name.

### 8.2.1.1 Starting an automatic measurement

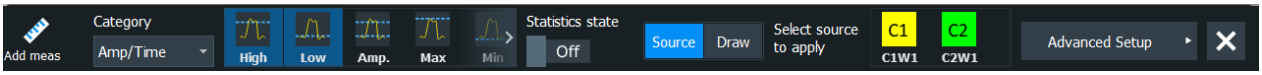
There are three methods to start an automatic measurement, each with slightly different effects:

- Using the "Measure" icon on the toolbar:
  See: "To start a measurement using the toolbar icon" on page 321.
- Pressing the [Measure] key on the front panel.
  See: "To start a measurement with the [Measure] key" on page 321.
- Using the "Measure" menu.
  See: Chapter 8.2.1.2, "Configuring measurements", on page 322.

**To start a measurement using the toolbar icon**

1. Tap the "Measurement" icon on the toolbar.

2. On the toolbar assist, select the category, the measurements and the source to be measured.




3. Define the measurement range in one of these ways:
   - To measure the complete waveform, tap the diagram with the waveform.
   - To define a gate that limits the measurement, draw a rectangle on the screen.

   The "Meas Results" are displayed.

**To start a measurement with the [Measure] key**

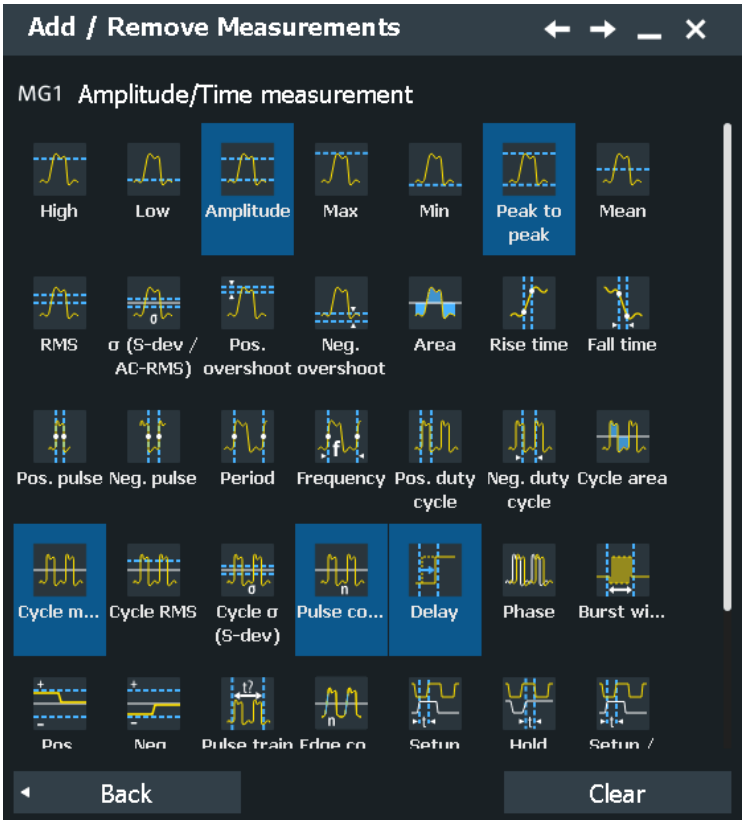
1. Select the waveform on the screen.
2. Press the [Measure] key.

   The measurement for the selected waveform is enabled using the next available measurement configuration. The "Meas Results" are displayed.

### 8.2.1.2 Configuring measurements

1. To open the "Measurements" dialog box, choose one of these ways:
   - If a measurement is already running, tap the result table. Then press the [Measure] key.
   - If no measurement is running, open the menu and select "Measure".

   If you double-tap the title of a result table, the toolbar assist opens. Here you can change the measurement types and enable statistics. If you tap "Advanced Setup" in the toolbar assist, the "Measurements" dialog box opens.
2. Select the subtab for the measurement group that you want to configure.
3. Select the measurement "Category", for example, "Amp/Time".
4. Tap "Source" and select the waveform to be measured.
   Spectrum measurements require an FFT math waveform as measurement source.
   Histogram measurements require a histogram as source.
5. Tap "Add / Remove" and select all measurements that you want to include in the measurement group. Tap "Clear" to deselect all settings.




6. Tap "Back" to close the selection box.
   All selected measurements are displayed in the list of active measurements.

7. A "Settings" icon indicates whether further settings are required. Tap the measurement. A dialog box opens, and you can adjust the measurement.

   The settings are explained in the following chapters:
   - Chapter 8.2.5.2, "Settings for amplitude/time measurements", on page 346
   - Chapter 8.2.6.2, "Settings for eye diagram measurements", on page 358
   - Chapter 8.2.7.2, "Settings for spectrum measurements", on page 365
   - Chapter 8.2.8.5, "Settings for histogram measurement", on page 375

8. Optionally, define a gate to restrict the measurement to a part of the waveform. See Chapter 8.2.3.1, "Using measurement gates", on page 329.

   If you enabled the measurement with the toolbar icon and drew a rectangle on the diagram, the gate is already defined and enabled.

9. To compile and display statistics for the measurement, enable "Statistics". See also Chapter 8.2.12.1, "Statistics", on page 384.

10. Optionally, perform a limit or margin check as described in Chapter 8.2.13.1, "Performing limit checks", on page 393.

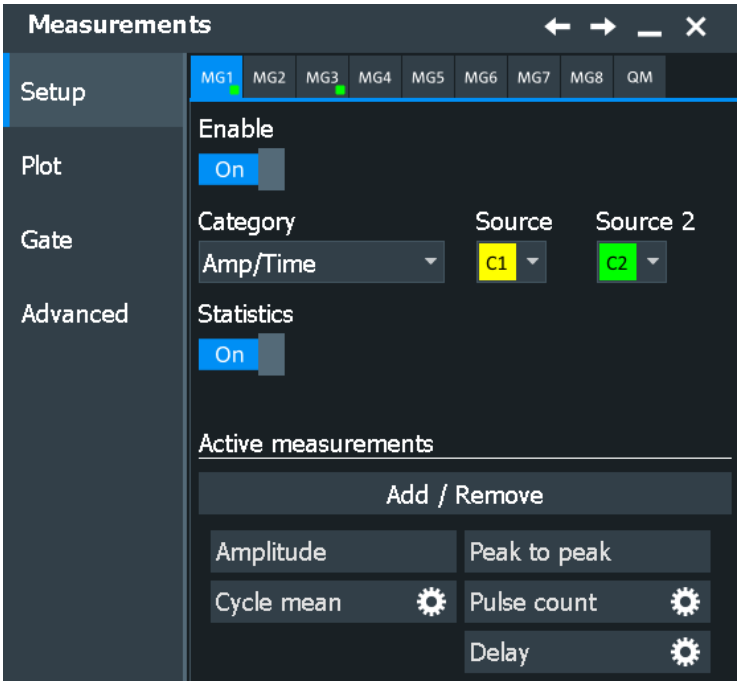
11. "Enable" the measurement group.

    The measurement results are displayed by default below the waveform diagram.

#### 8.2.1.3 General measurement settings

Automatic measurements are configured in the "Measurements" dialog box.

Access: [Measure] > "Meas Group" tab



This section describes the settings that relate to all measurements. Specific settings are described in the corresponding category chapters:

- Chapter 8.2.5.2, "Settings for amplitude/time measurements", on page 346
- Chapter 18.1, "Jitter measurements (option R&S RTP-K12)", on page 1024
- Chapter 8.2.6.2, "Settings for eye diagram measurements", on page 358
- Chapter 8.2.7.2, "Settings for spectrum measurements", on page 365
- Chapter 8.2.8.5, "Settings for histogram measurement", on page 375

**MG 1/2/3/4/5/6/7/8/Quick Meas**
Subtabs for each measurement group, and for the quick measurement. The subtabs contain the configuration settings. A green dot on the tab indicates that the measurement group is active.

**Enable**
Starts the measurements of the selected measurement group.

Remote command:
`MEASurement<m>[:ENABle]` on page 1427

**Category**
Measurement category. The following categories are available:

Time domain:
- Amp/Time: amplitude and time measurements
- Eye: eye diagram measurements
- Histogram: measurements on histograms
- Jitter measurements (available if one of the jitter options is installed)
- Protocol: advanced analysis (available if option R&S RTP-K35 is installed)

Frequency domain
- Spectrum: measurements on spectrum waveforms
- Histogram: measurements on histograms

Remote command:
`MEASurement<m>:CATegory` on page 1429

**Source, Source 2**
Define the source of the measurement group. The 2nd source is required for amplitude/time measurements that are performed on two waveforms (e.g. delay, phase). Availability of sources depends on the selected category and installed options.

Remote command:
`MEASurement<m>:SOURce` on page 1427

**Active measurements, Add / Remove**
The "Active Measurements" list shows the measurements that are selected for the current category. At least, one measurement must be selected. If further settings are available for a measurement, a settings icon is shown beside the measurement's name.

To change the selection, tap "Add / Remove Measurements".

In the "Add / Remove Measurements" dialog box, select the measurements that you need. Tap "Clear" to disable all selected measurements. Tap "OK" to confirm the selection.

Remote command:
`MEASurement<m>:MAIN` on page 1430
`MEASurement<m>:ADDitional` on page 1431

**Envelope**
This setting is only available for measurements on envelope waveforms, see "Arithmetic" on page 133.

"Both" The upper and the lower envelope are used in measurements. For time measurements, the averages of min and max values are used, that is, the measurement is performed on the average waveform built from the upper and lower envelope.

"Maximum" Measurements are performed on the upper envelope.

"Minimum" Measurements are performed on the lower envelope.

Remote command:
`MEASurement<m>:ENVSelect` on page 1439

**Statistics**
Enables the calculation and display of statistical results.

Remote command:
`MEASurement<m>:STATistics[:ENABle]` on page 1489

## 8.2.2 Measurement results

By default, the results of automatic measurements are displayed below the waveform diagram when an automatic measurement is enabled.

| Meas Group 1 C1 | | Meas Group 2 C2 | |
|---|---|---|---|
| High | 122.53 mV | Rise time | 11.847 ns |
| Low | -124.51 mV | Fall time | --- |
| Amplitude | 247.04 mV | Pos. pulse | 20.05 ns |
| Frequency | 25 MHz | Neg. pulse | 20 ns |
| Cycle RMS | 87.999 mV | | |

If you want to save space in the display, drag the results to the signal bar. The most important results are displayed and updated in a results icon.

The function "Clear all" in the "Display" menu resets all results including long-term measurement and statistic results, and also deletes all waveforms and the history.

Which results are displayed depends on the selected measurements and is described in detail in the following chapters.

The following additional results are available:

- **Statistics**

You can enable statistical evaluation of the measurement results, and select the statistical results that you want to see. Statistic information is provided in the result box. Stopping and restarting the acquisition does not reset statistics but only stops and continues them.
See Chapter 8.2.12, "Result analysis", on page 384

- **Measurement histograms**
  The results of measurements can be displayed in a histogram which shows the density distribution of the measurement results in a graphic and thus illustrates the statistics of the measurements.
  See Chapter 8.2.8, "Histograms and histogram measurements", on page 368
- **Long-term measurements**
  Long-term measurements show the behavior of measurement results over a longer time or for many samples. You can define the number of long-term points and export the long-term data, including statistical results. The measurement histogram is a vertical histogram shown in the long-term diagram.
  Long-term measurements are performed on the measurement that is selected for analysis and math on the "Result Analysis" tab.
  See: Chapter 8.2.12, "Result analysis", on page 384
- **Intermediate results**
  You can display auxiliary result lines and reference levels in the source diagram, see Chapter 8.2.2.2, "Configuring the results display", on page 327.

Remote commands:

- `MEASurement<m>:ARES?` on page 1433
- `MEASurement<m>:ARNames` on page 1434
- `MEASurement<m>:RESult[:ACTual]?` on page 1434
- `MEASurement<m>:RESult:COUNt?` on page 1436

### 8.2.2.1 Measurement status

The overall status of measurement results is indicated by various icons. In general, a question mark before the result value indicates that the measurement result might not be correct due to insufficient amplitude level. Check your amplitude and reference level settings. The icon colors indicate the state of the limit and margin checks.

| Icon | Description |
|---|---|
| No icon, no result value ("---") | The instrument cannot measure the required value, for example, if the acquisition does not contain at least one complete period for frequency and cycle measurements. Check and adjust the waveform settings to get results. |
| | The measurement result might not be correct due to insufficient amplitude level. Check your amplitude and reference level settings.<br>Limit and margin checks are disabled. |
| | Limit and margin checks passed, measurement results are reliable. |

| Icon | Description |
|---|---|
| | The measurement result might not be correct due to insufficient amplitude level. Check your amplitude and reference level settings.<br>Limit and margin checks passed. |
| | The measurement result might not be correct due to insufficient amplitude level. Check your amplitude and reference level settings.<br>Margin checks failed. |
| | Margin checks failed. |
| | The measurement result might not be correct due to insufficient amplitude level. Check your amplitude and reference level settings.<br>Limit checks failed. |
| | Limit checks failed. |

### 8.2.2.2 Configuring the results display

The measurement results can be displayed in a table below the waveform diagrams, in a floating result box, or in a minimized result icon on the signal bar.

The display settings for measurements are provided on the "Gate/Display" tab, see Chapter 8.2.2.3, "Display settings for results", on page 327.

**To clear the measurement results**

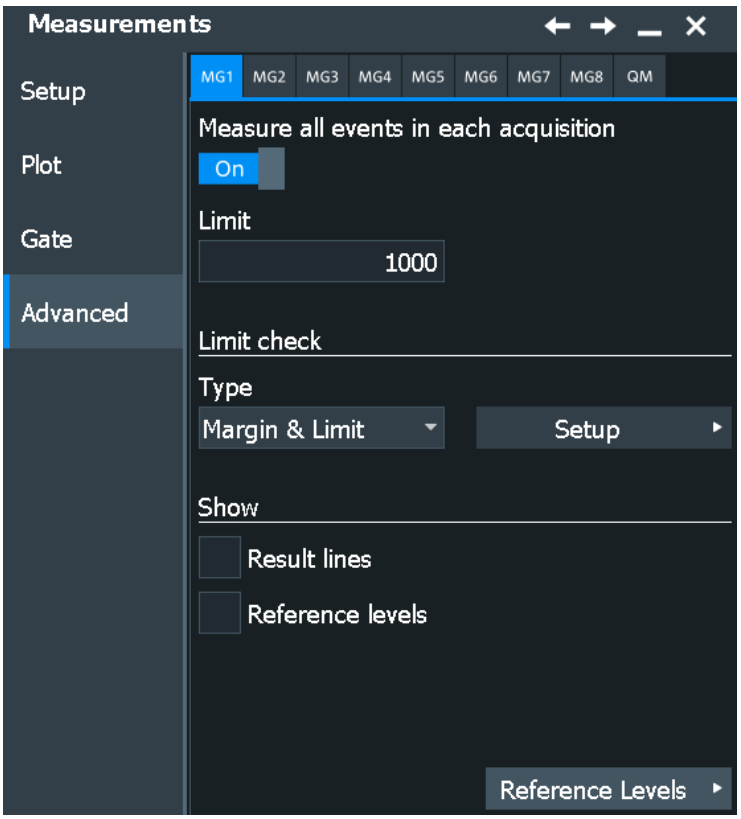
1. On the "Display" menu, tap "Clear all".
2. To restart measurement statistics, without deleting other results, select "Reset" on the "Result Analysis" tab.

   The results in the selected measurement result box are cleared and written anew.

### 8.2.2.3 Display settings for results

Display settings are specific for each measurement group.

Access: [Measure] > "Advanced" tab




General display setting for measurements are provided in "Settings" > "Appearance" > "Measurements", see Chapter 4.3.8, "Measurement appearance settings", on page 89.

**Result lines**
Result lines mark the samples in the waveform that are used to obtain the measurement result. For example, maximum, minimum, high and low values, and mean are merked by horizontal lines; start and end are marked by vertical lines.

Remote command:
`MEASurement<m>:DISPlay:RESults` on page 1487

**Reference levels**
Displays the reference levels used for the measurement in the diagram.

Remote command:
`MEASurement<m>:DISPlay:LEVels` on page 1487

**Result control panel**
Extends the result box of the selected measurement group with the source settings and the statistics enabling. Thus you can check and change the measurement sources directly in the results box, and also enable statistics there. The setting is only available if the result boxes are not grouped ("Group result dialogs" = Off).

### 8.2.3 Measurement gates

Gate areas limit the measurement to a user-defined range of the waveform. The gate settings are defined on the "Gate " tab.

Each measurement group can use its own gate. Make sure to select the correct measurement group.

#### 8.2.3.1 Using measurement gates

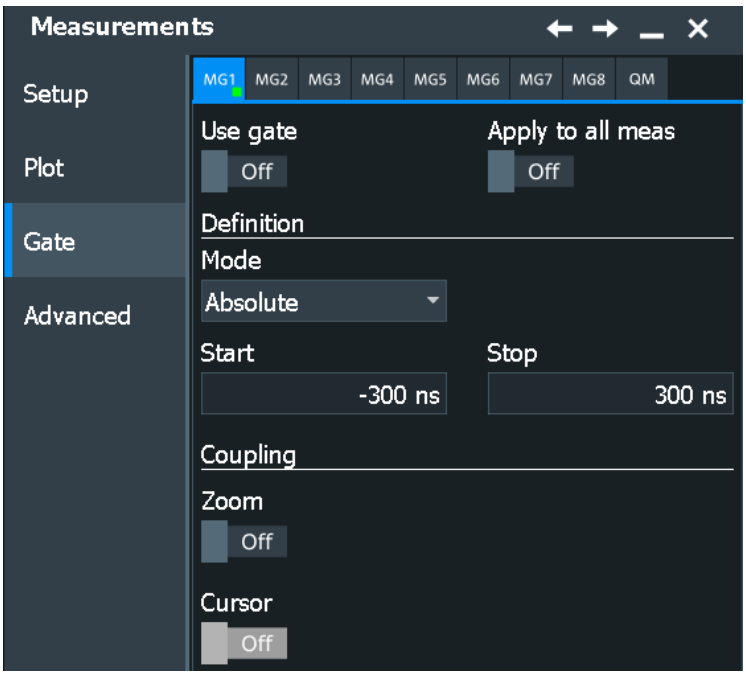
If you have enabled the measurement with the toolbar icon and drew a rectangle on the diagram, the gate is automatically defined and enabled. If you want to create a gate for a running measurement, or if you want to modify the gate area, configuration is done in the "Measurements" > "Gate" dialog box.

1. In the "Measurements" dialog, select the "Gate" tab.
2. Select the subtab for the measurement group you want to configure.
3. To define the gate, use one of the following methods:
   - Define the start value and the stop value of the gate area by entering either absolute or relative values.
   - If a zoom area has already been defined for the waveform, couple the gate area to the zoom area by selecting the "Zoom coupling" option.
   - If a cursor measurement has already been defined for the waveform, couple the gate area to the cursor lines by selecting the "Cursor coupling" option.
4. If you want to use the same gate for all measurement group, enable "Apply to all meas".
5. Tap the "Use gate" icon to enable the gate usage.

   The measurement is performed on the selected part of the waveform. The gate is shown in the diagram.

#### 8.2.3.2 Gate settings for measurements

Access: [Measure] > "Gate" tab




Result display settings are described in Chapter 8.2.2.3, "Display settings for results", on page 327.

**Use gate**
Considers the gating settings for the selected measurement and displays the gate.

Remote command:
`MEASurement<m>:GATE[:STATe]` on page 1503

**Apply to all meas**
If you enable the gate coupling, the gate settings of the selected measurement are copied to all other measurements. Thus, all measurements use the same gate. If zoom or cursor coupling is active in a measurement, the zoom size and cursor positions are adjusted.

Remote command:
`MEASurement<m>:GATE:GCOupling` on page 1506

**Gate description**
Defines whether the gate settings are configured using absolute or relative values.

"Absolute" The gate is defined by absolute start and stop values.

"Relative" The gate's start and stop values are defined by a percentage of the value range.

Remote command:
`CALCulate:MATH<m>:FFT:GATE:MODE` on page 1533
`MEASurement<m>:GATE:MODE` on page 1503
`SEARch:GATE:MODE` on page 1595

**(Relative) Start**
Defines the starting value for the gate.

Remote command:
CALCulate:MATH<m>:FFT:GATE:ABSolute:STARt on page 1532
CALCulate:MATH<m>:FFT:GATE:RELative:STARt on page 1533
MEASurement<m>:GATE:ABSolute:STARt on page 1504
MEASurement<m>:GATE:RELative:STARt on page 1504
SEARch:GATE:ABSolute:STARt on page 1596
SEARch:GATE:RELative:STARt on page 1596

**(Relative) Stop**
Defines the end value for the gate.

Remote command:
CALCulate:MATH<m>:FFT:GATE:ABSolute:STOP on page 1532
CALCulate:MATH<m>:FFT:GATE:RELative:STOP on page 1533
MEASurement<m>:GATE:ABSolute:STOP on page 1504
MEASurement<m>:GATE:RELative:STOP on page 1504
SEARch:GATE:ABSolute:STOP on page 1596
SEARch:GATE:RELative:STOP on page 1597

**Zoom**
Zoom coupling is available if a zoom is defined. As long as "Zoom coupling" is enabled, the gate area is defined identically to the zoom area - if you change the zoom, the gate changes as well.

If several zoom diagrams are defined, select the zoom diagram to be used for gating. The "Start" and "Stop" values of the gate are adjusted accordingly.

Zoom coupling can be set for measurement gates, FFT gates, and search gates. The zoom must be defined on the diagram that contains the signal source of the measurement, FFT, or search.

Remote command:
MEASurement<m>:GATE:ZCOupling on page 1505
MEASurement<m>:GATE:ZDIagram on page 1506
CALCulate:MATH<m>:FFT:GATE:ZCOupling on page 1534
SEARch:GATE:ZCOupling on page 1597
SEARch:GATE:ZDIagram on page 1597

**Cursor**
If enabled, the gate area is defined by the cursor lines of an active cursor measurement.

If several cursor measurements are enabled, select the cursor set to be used for gating. The "Start" and "Stop" values of the gate are adjusted to the values of the cursor line positions. The measurement is limited to the part of the waveform between the cursor lines.

Remote command:
MEASurement<m>:GATE:CCOupling on page 1505
MEASurement<m>:GATE:CURSor on page 1504

### 8.2.4 Reference levels

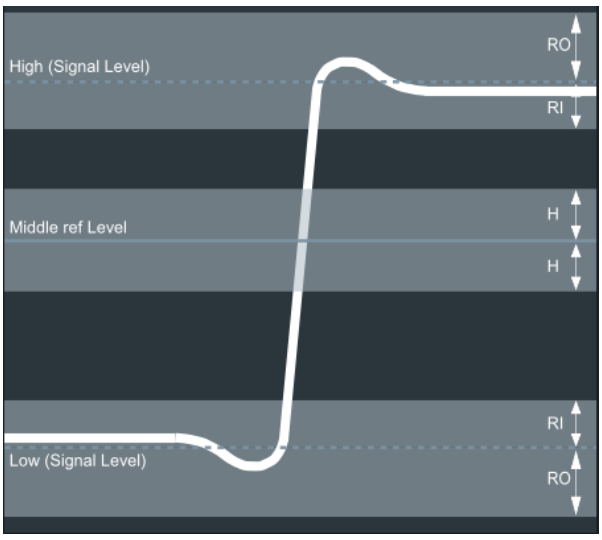
Some measurement require reference levels to obtain the measurement points, e.g. time measurements or pulse count. Reference levels are referred to the signals, for each waveform you can define specific reference levels. Thus, for all measurements on a waveform the same reference levels are used.

Usually, reference levels are determined automatically. The instrument determines the high and low signal levels based on amplitude and histogram measurements of the acquisition. The reference levels are set relatively to the determined signal levels.

However, for irregular data and in special measurement setups it may be useful to configure the levels manually:

- Data signals can contain intervals where no data is transmitted, so that a high and low state cannot be determined for each acquisition. In this case, you can define the high and low signal levels manually to evaluate other measurement results.
- If the signal levels vary strongly or have large overshoots, the rise and fall levels may be difficult to determine.
- If fixed levels are defined for the DUT, you can configure the reference levels in the R&S RTP correspondingly and analyze the resulting measurement data.

In manual configuration, the reference levels can be set relatively to defined signal levels or as absolute values. You can also set the reference levels directly.




In addition to reference and signal levels, you can define a hysteresis for the middle reference level and tubes for signal levels. Hysteresis is useful for measurements that determine zero-crossings. Period, frequency, and pulse measurements are based on hysteresis - the instrument returns results if the amplitude of the signal exceeds the hysteresis. Thus, measurement during the transient oscillation is also possible. Tubes define evaluation ranges for measurements that require detection of the high level or low level. If the signal value remains within the defined tubes, it is considered to be high or low.

Reference levels and result lines can be displayed in the diagram, see "To display reference levels and result lines" on page 334.

#### 8.2.4.1 Configuring reference levels

**To determine reference and signal levels automatically**

By default, the histogram of the measurement data is evaluated to determine the required levels automatically. However, you can define several parameters to adapt the evaluation to your data.

1. Open "Menu" > "Measure" > "Advanced" tab > "Reference Level" > "Levels" tab.
2. Define the "Source", the waveform for which the reference is defined. The source can be any signal input, math or reference waveform.
3. Select "Mode" = "Auto" .
4. Define the "Signal level", the method which is used to determine the signal levels. For details, see "Signal level" on page 337.
5. By default, the lower reference level is defined at 10% of the signal amplitude, the middle reference level at 50% and the upper reference level at 90%. You can select other "Relative levels" to be used for evaluation.
   If default percentages do not fit, select "User defined" and enter the percentages for the upper, middle, and lower reference levels.
   The signal levels are determined by the instrument.
6. To determine the reference levels using average values from several histograms, enable the "Histogram averaging" option and define an "Average Count" to define how many histograms are averaged.
   Averaging is not available if "Absolute peak" are selected as the "Signal level".

**To determine reference levels manually**

You can configure the reference levels manually as fixed absolute or relative values.

1. Open "Menu" > "Measure" > "Advanced" tab > "Reference Level" > "Levels" tab.
2. Define the "Source", the waveform for which the reference levels are defined. The source can be any signal input, math or reference waveform.
3. Select "Mode" = "Manual" .
4. Under "Level definition", select whether you want to define the levels using absolute or relative values.
5. Under "User level", select whether you want to configure the high and low signal levels ("Signal") or the lower, middle and upper reference levels ("Reference").
6. To define high and low signal levels if "Signal" is selected:
   a) Enter the absolute high and low signal levels.

b) If "Level definition" is relative, select one of the predefined "Relative levels". If default percentages do not fit, select "User defined" and enter the percentages for the upper, middle, and lower reference levels.
The upper and lower reference levels are computed from the signal level values and the percentage values.
c) If "Level definition" is absolute, set the absolute "Top distance" and "Bottom distance" values, the differences between signal and reference levels.
The upper and lower reference levels are computed from the signal level values and the distances.

7. To define lower, middle and upper reference levels if "User reference level" is selected:
   a) Enter the absolute upper and lower reference levels.
   b) If "Level definition" is relative, select one of the predefined "Relative levels". If default percentages do not fit, select "User defined" and enter the percentages for the upper, middle, and lower reference levels.
   The high and low signal levels are computed from the reference level values and the percentage values.
   c) If "Level definition" is absolute, set the absolute "Top distance" and "Bottom distance" values, the differences between signal and reference levels.
   The high and low signal levels are computed from the reference level values and the distances.

**To define hysteresis and tubes**

1. To define a hysteresis for the middle reference level:
   a) Select the "Advanced" tab.
   b) Enter a percentage of the selected signal level.

   A rise or fall from the middle reference value that does not exceed the hysteresis is rejected and not considered a zero-crossing.

2. To define a tube for the high and low signal levels:
   a) In the "Relative outer (RO)" field, define a percentage of the signal level by which the absolute signal level may be larger than high signal level or lower than the low signal level.
   b) In the "Relative inner (RI)" field, define a percentage of the signal level by which the absolute signal level may be higher than the low signal level or lower than the high signal level.

**To display reference levels and result lines**

1. Open "Menu" > "Measure" > "Advanced" tab.
2. Select the tab for the measurement you want to configure.

3. Enable "Result lines" or "Reference levels" option, or both.

   The reference levels and intermediate results are displayed in the waveform diagram.

### 8.2.4.2 Level settings

Access: "Menu" > "Measure" > "Advanced" tab > "Reference Level" > "Levels" tab.

On the "Levels" tab, you define how the reference levels are calculated, or you set them directly.

The following setups are available, according to the selected "Mode":

#### Automatic reference level mode

In automatic reference level mode, the reference levels are always relative values. You can select one of the predefined sets, or define individual percentage values.

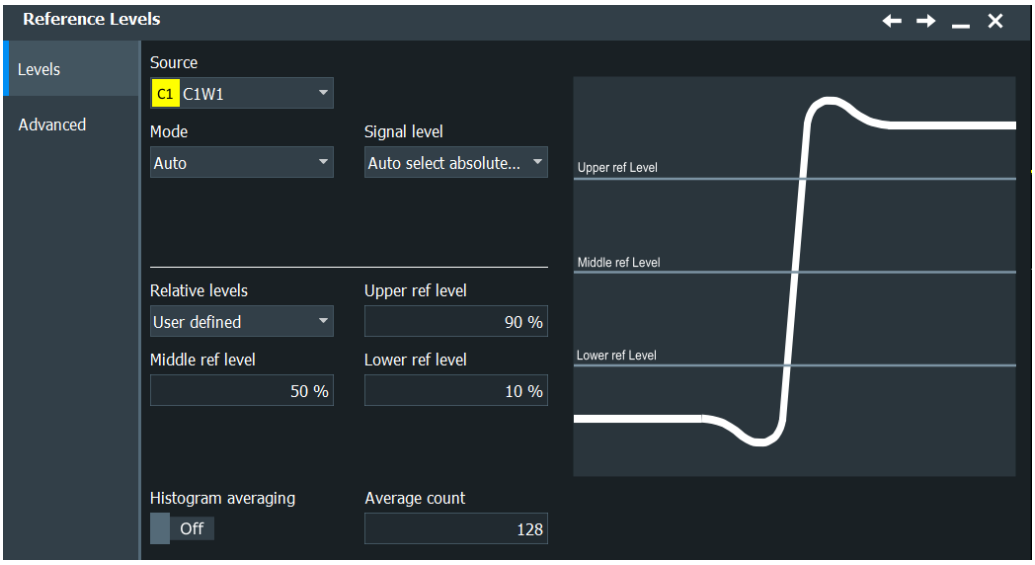



*Figure 8-4: Automatic reference level definition*

#### Manual reference level mode

In manual reference level mode, relative and absolute level definitions are possible.

In manual reference level mode with relative level definition, you define the absolute values of high and low signal levels or reference levels, and the reference levels as percentages of the signal amplitude.

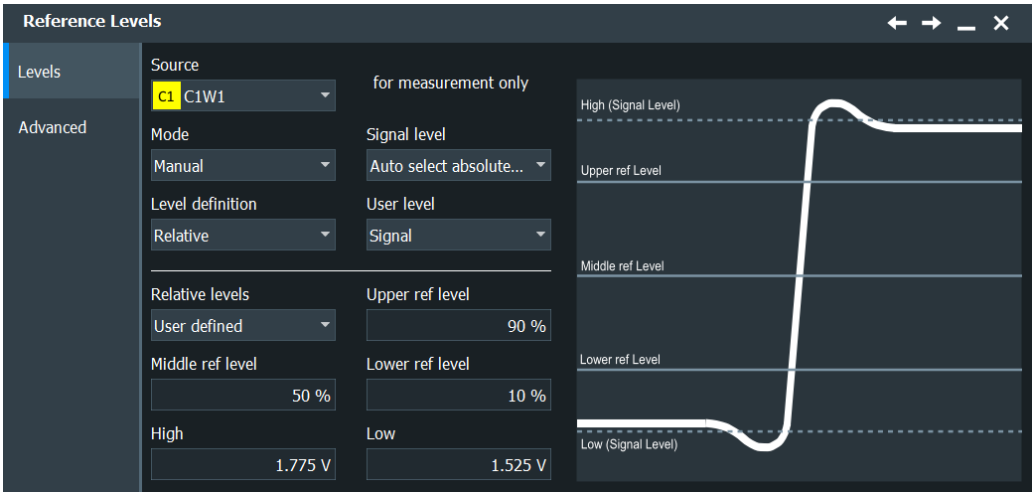



*Figure 8-5: Manual reference level mode, relative level definition*

In manual reference level mode with absolute level definition, you define the absolute values of high and low signal levels or reference levels, and the distances between reference and signal levels.

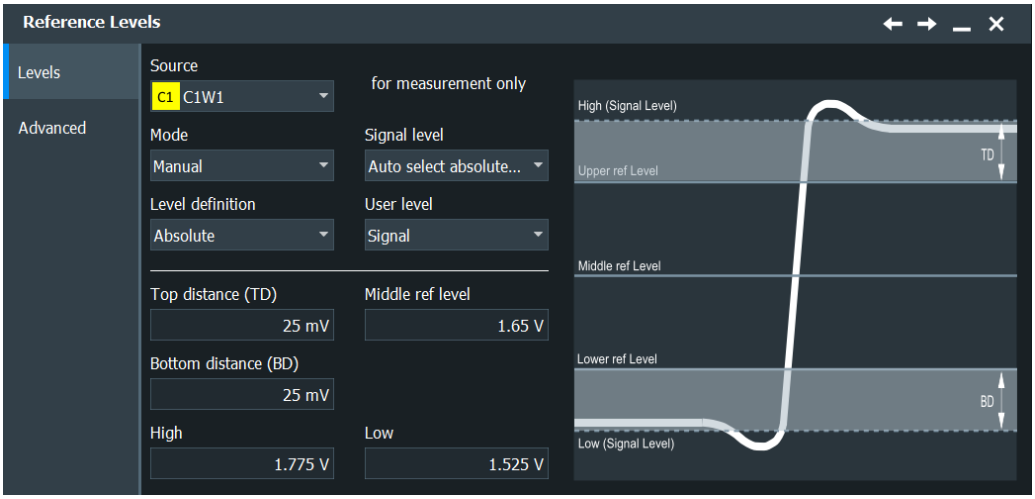



*Figure 8-6: Manual reference level mode, absolute level definition*

**Source**
Defines the source for which the reference levels are defined. The source can be any signal input, math or reference waveform.

Remote command:
Source is defined by suffix <m> in "REFLevel" subsystem, see Chapter 24.12.15, "Reference levels", on page 1509

**Mode**
Defines whether the reference level is configured manually or automatically.

Remote command:
`REFLevel<m>:LDETection` on page 1509

**Signal level**
Defines the computation method for high and low signal levels. The instrument analyzes the signal, performs amplitude and histogram measurements, and defines the signal levels using the selected method.

The selected method is used to compute the signal levels for determination of reference levels in automatic reference level mode. It is also used for high, low, and amplitude measurements in automatic and manual reference level modes.

See also: Chapter 8.2.8, "Histograms and histogram measurements", on page 368

| | |
|---|---|
| "Auto select absolute probability" | The most suitable signal levels for the selected measurement are used. |
| "Peak probability" | The signal levels with the highest probability values are used. These are the upper peak value and the lower peak value of the histogram measurement. |
| "Mean probability" | The signal levels with mean probabilities are used. |
| "Absolute peak" | The absolute peak signal levels are used. These are the maximum and minimum signal values of the amplitude measurement. |
| "Upper absolute peak - Lower mean probability" | The high signal level is the upper absolute peak (the maximum signal level), and the low signal level is the level with the mean probability in the lower half of the histogram. |
| "Upper mean probability - Lower absolute peak" | The high signal level is the level with mean probability in the upper half of the histogram, and the low signal level is the lower absolute peak (the minimum signal level). |
| "Upper absolute peak - Lower manual" | The high signal level is the maximum result value of the amplitude measurement. The low signal level is manually set using "Low". |
| "Upper manual - Lower absolute peak" | The high signal level is set manually using "High". The low signal level is the minimum result value of the amplitude measurement. |

Remote command:
`REFLevel<m>:AUTO:MODE` on page 1512

**Level definition**
In manual reference level mode, the setting defines whether the reference is configured using absolute or relative values.

Remote command:
`REFLevel<m>:LMODe` on page 1510

**User level**
In manual reference level mode, the setting defines whether the user-defined signal levels or user-defined reference levels are used for the measurements.

"Signal" You can define the high and low signal levels.

"Reference" You can define the reference levels.

Remote command:
`REFLevel<m>:USRLevel` on page 1511

**Relative levels**
Sets the lower, middle and upper reference levels, defined as percentages of the signal amplitude.

Available relative levels:

- 5/50/95
- 10/50/90
- 20/50/80
- User defined: Enter "Upper ref level", "Middle ref level", and "Lower ref level".

For example, for "5/50/95" the levels are set to the following values:

- Lower reference level = 5% of the signal amplitude
- Middle reference level = 50% of the signal amplitude
- Upper reference level = 95% of the signal amplitude

Remote command:
`REFLevel<m>:RELative:MODE` on page 1510

**Upper ref level, Middle ref level, Lower ref level**
Define the reference levels in percent, if "Relative levels" is set to "User-defined".

Remote command:
`REFLevel<m>:RELative:UPPer` on page 1517
`REFLevel<m>:RELative:MIDDle` on page 1518
`REFLevel<m>:RELative:LOWer` on page 1518

**High**
Sets the high signal level.

The high signal level is set in manual reference level mode, for absolute level definition and user signal level selection.

Remote command:
`REFLevel<m>:ABSolute:HIGH` on page 1514

**Low**
Sets the low signal level.

The low signal level is set in manual reference level mode, for absolute level definition and user signal level selection.

Remote command:
`REFLevel<m>:ABSolute:LOW` on page 1514

**Middle**
For user signal level selection, the level is the middle level between high and low signal level. The value is adjusted automatically if you change the high or low signal levels. Vice versa, if you change the middle level, the high and low signal levels are adjusted.

For user reference level selection, the level is the middle level between upper and lower reference level. The value is adjusted automatically if you change the upper or lower reference levels. Vice versa, if you change the middle level, the upper and lower reference levels are adjusted.

Remote command:
`REFLevel<m>:ABSolute:MLEVel` on page 1516

**Top distance**
The distance between the high signal level and the upper reference level - for manual reference level mode and absolute level definition.

Remote command:
`REFLevel<m>:ABSolute:TDIStance` on page 1515

**Bottom distance**
The distance between the lower reference level and the low signal value - for manual reference level mode and absolute level definition.

Remote command:
`REFLevel<m>:ABSolute:BDIStance` on page 1515

**Upper level**
The upper reference level required, for example, to determine a rise - for manual reference level mode, absolute level definition and user reference level.

Remote command:
`REFLevel<m>:ABSolute:ULEVel` on page 1516

**Lower level**
The lower reference level required, for example, to determine a fall - for manual reference level mode, absolute level definition and user reference level.

Remote command:
`REFLevel<m>:ABSolute:LLEVel` on page 1517

**Histogram averaging**
Enables averaging over several histograms to determine the reference levels.

This function is only available in automatic reference level mode.

Remote command:
`REFLevel<m>:AUTO[:STATe]` on page 1513

**Average Count**
Defines the number of histograms to calculate the average from.

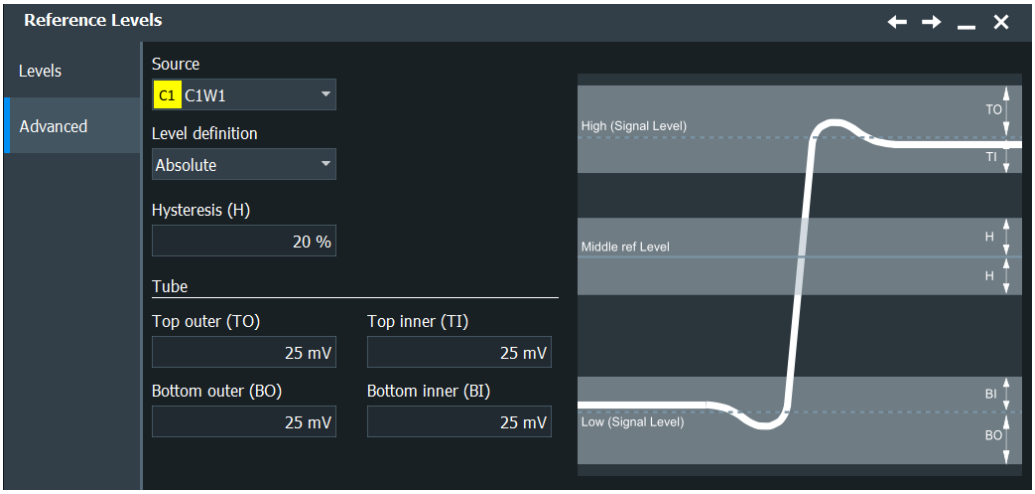
This function is only available in automatic reference level mode.

Remote command:
`REFLevel<m>:AUTO:COUNt` on page 1513

#### 8.2.4.3 Advanced tab

Access: "Menu" > "Measure" > "Advanced" tab > "Reference Level" > "Advanced" tab.

This tab allows you to define evaluation tubes for measurements that require high-level or low-level detection. If the signal value remains within the defined tubes, it is considered to be high or low.




See also: Source and Level definition.

**Hysteresis**
Defines a hysteresis for the middle reference level. A rise or fall from the middle reference value that does not exceed the hysteresis is rejected as noise.

Remote command:
`REFLevel<m>:RELative:HYSTeresis` on page 1519

**Top outer**
Defines an area above the high signal level which is still considered to be high level.

Remote command:
`REFLevel<m>:ABSolute:TOTube` on page 1520
`MEASurement<m>:REFLevel:RESult:TOUTer?` on page 1523

**Top inner**
Defines an area beneath the high signal level which is still considered to be high level.

Remote command:
`REFLevel<m>:ABSolute:TITube` on page 1520
`MEASurement<m>:REFLevel:RESult:TINNer?` on page 1523

**Bottom inner**
Defines an area above the low signal level which is still considered to be low level.

Remote command:
`REFLevel<m>:ABSolute:BITube` on page 1521
`MEASurement<m>:REFLevel:RESult:BINNer?` on page 1522

**Bottom outer**
Defines an area beneath the low signal level which is still considered to be low level.

Remote command:
`REFLevel<m>:ABSolute:BOTube` on page 1521
`MEASurement<m>:REFLevel:RESult:BOUTer?` on page 1522

**Relative outer**
Defines a percentage of the signal level by which the absolute signal level may be larger than the high signal level or lower than the low signal level to be considered high or low, respectively.

Remote command:
`REFLevel<m>:RELative:OTUBe` on page 1519

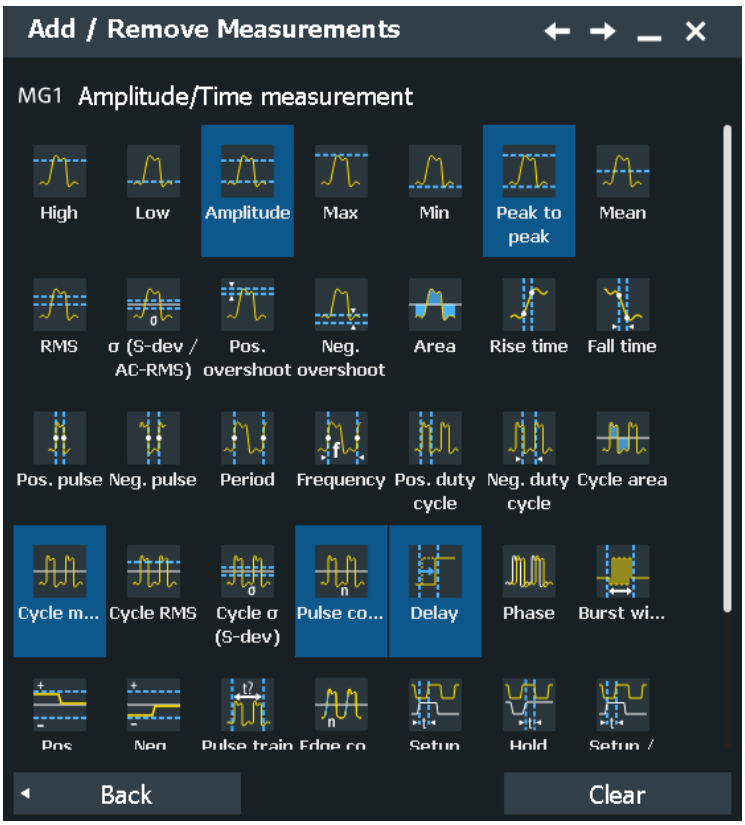
**Relative inner**
Defines a percentage of the signal level by which the absolute signal level may be higher than the low signal level or lower than the high signal level to be considered low or high, respectively.

Remote command:
`REFLevel<m>:RELative:ITUBe` on page 1520

## 8.2.5 Amplitude/time measurements

### 8.2.5.1 Overview of amplitude/time measurements

Access: [Measure] > "Meas Group" tab > "Amp/Time" category



The R&S RTP provides various voltage, time, area and counting measurements in the category "Amp/Time". Some measurements require reference levels to be set according to the measurement purpose.

Reference levels are explained in Chapter 8.2.4, "Reference levels", on page 332.

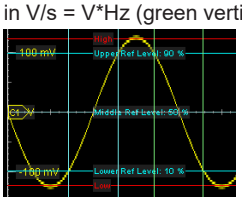

- Amplitude measurements ..... 342
- Time measurements ..... 343
- Area measurements ..... 345
- Counting ..... 346
- Arithmetic ..... 346


## Amplitude measurements

Access: [Measure] > "Meas Group" tab > "Amp/Time" category

***Table 8-1: Amplitude measurements***

| | Meas. type | Symbol | Description/Result |
|---|---|---|---|
| | High | $X_{High}$ | High signal level |
| | Low | $X_{Low}$ | Low signal level |
| | Amplitude | $X_{Ampl}$ | Amplitude of the signal: the difference of high and low signal levels<br>$X_{Ampl} = X_{High} - X_{Low}$ |
| | Max | $X_{Max}$ | Absolute maximum value of the waveform |
| | Min | $X_{Min}$ | Absolute minimum value of the waveform |
| | Peak to peak | $X_{PkPk}$ | Peak-to-peak value of the waveform: the difference of maximum and minimum values<br>$X_{Ampl} = X_{Max} - X_{Min}$ |
| | Mean | $X_{Mean}$ | Arithmetic average of the waveform voltage values<br>$X_{Mean} = \frac{1}{N_{Eval}} \sum_{i=1}^{N_{Eval}} x(i)$ |
| | RMS | $X_{RMS}$ | RMS (root mean square, quadratic mean) of the waveform voltage values<br>$X_{RMS} = \sqrt{\frac{1}{N_{Eval}} \sum_{i=1}^{N_{Eval}} x^2(i)}$ |
| | σ (S-dev) | $\sigma_X$ | Standard deviation of the waveform samples<br>$\sigma_X = \sqrt{\frac{1}{N_{Eval}-1} \sum_{i=1}^{N_{Eval}} (x(i) - X_{Mean})^2}$ |

| | Meas. type | Symbol | Description/Result |
|---|---|---|---|
| | Pos. overshoot | $R_{Pos}$ | Positive overshoot of a square wave, calculated from measurement values High, Max, and Amplitude<br>$+Ovr = \frac{V_{top} - V_{P+}}{V_{Amp}} \cdot 100\%$ |
| | Neg. overshoot | $R_{Neg}$ | Negative overshoot of a square wave, calculated from measurement values Min, Low, and Amplitude<br>$-Ovr = \frac{V_{base} - V_{P-}}{V_{Amp}} \cdot 100\%$ |
| | Cycle mean | | The mean value of one cycle |
| | Cycle RMS | | The RMS (root mean square) value of one cycle |
| | Cycle σ (S-dev) | | The standard deviation of one cycle |
| | Trig. ProbeMeter | | The DC voltage from the connected probe. Only available if an active Rohde & Schwarz probe with ProbeMeter is connected. |
| | Slew rate rising | | Steepness of the rising edge: voltage difference between the lower and higher reference level, divided by the rise time. Result in V/s = V*Hz (blue vertical lines in the picture below). |
| | Slew rate falling | | Steepness of the falling edge: voltage difference between the higher and lower reference level, divided by the fall time. Result in V/s = V*Hz (green vertical lines in the picture below).<br>  |

## Time measurements

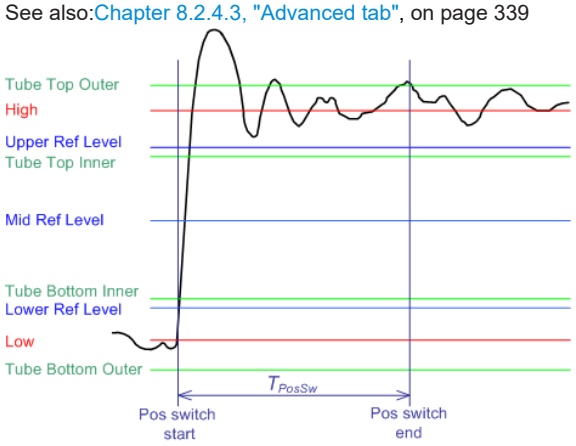
Access: [Measure] > "Meas Group" tab > "Amp/Time" category

Reference levels are explained in Chapter 8.2.4, "Reference levels", on page 332.

***Table 8-2: Time measurements***

| | Meas. type | Symbol | Description/Result |
|---|---|---|---|
| | Rise time | $T_{Rise}$ | Rise time of the left-most rising edge of the waveform. Rise time is the time it takes the signal to rise from the low reference level to the high reference level. Measurement all events in the acquisition is possible. |
| | Fall time | $T_{Fall}$ | Falling time of the left-most falling edge of the waveform. Fall time is the time it takes the signal to fall from the high reference to the low reference. Measurement all events in the acquisition is possible. |

| | Meas. type | Symbol | Description/Result |
|---|---|---|---|
| | Pos. pulse | $T_{PosPulse}$ | Width of a positive pulse: time between a rising edge and the following falling edge measured on the middle reference level. The measurement requires at least one complete period of a triggered signal. Measurement all events in the acquisition is possible. |
| | Neg. pulse | $T_{NegPulse}$ | Width of a negative pulse: time between a falling edge and the following rising edge measured on the middle reference level. The measurement requires at least one complete period of a triggered signal. Measurement all events in the acquisition is possible. |
| | Period | $T_{Period}$ | Time between two consecutive waveform edges of the same direction, measured on the middle reference level. The measurement requires at least one complete period of a triggered signal. Measurement all events in the acquisition is possible. |
| | Frequency | $f_{Period}$ | Frequency of the signal, reciprocal value of the period.<br>$f_{Period} = 1 / T_{Period}$ |
| | Pos. duty cycle | $R_{PosCyc}$ | Positive duty cycle: Width of a positive pulse in relation to the period in %. The measurement requires at least one complete period of a triggered signal. Multiple measurements are possible.<br>$R_{PosCyc} = \frac{T_{PosPulse}}{T_{Period}} \cdot 100\%$ |
| | Neg. duty cycle | $R_{NegCyc}$ | Negative duty cycle: Width of a negative pulse in relation to the period in %. The measurement requires at least one complete period of a triggered signal. Multiple measurements are possible.<br>$R_{NegCyc} = \frac{T_{NegPulse}}{T_{Period}} \cdot 100\%$ |
| | Delay | | Time difference between any two edges of two measurement sources at any reference level.<br>The measurement result is negative if the edge of the second source comes before the edge of the first source.<br>See: "Settings for delay and phase measurements (analog sources)" on page 348 |
| | Phase | | Phase difference between two waveforms.<br>*Phase = Delay / Period * 360* |
| | Burst width | | Duration of one burst, measured from the first edge to the last |

| | Meas. type | Symbol | Description/Result |
|---|---|---|---|
| | Pos. switching | $T_{PosSw}$ | Settling time at rising edges: Time between crossing the lower reference level and the last return of the signal into the top tolerance tube.<br>See also:Chapter 8.2.4.3, "Advanced tab", on page 339<br>  |
| | Neg. switching | $T_{NegSw}$ | Settling time at falling edges: Time between crossing the upper reference level and the last return of the signal into the bottom tolerance tube. See also "Pos. switching" above. |
| | Pulse train | | Duration of N positive pulses, measured from the rising edge of the first pulse to the falling edge of the N-th pulse. Define N for the measurement. |
| | Setup<br>Hold<br>Setup/Hold time | $T_{Setup}$ and $T_{Hold}$ | Setup and Hold time measurement with positive and/or negative clock edge.<br>See: "Setup/Hold measurement settings" on page 349 |
| | Setup/Hold ratio | $T_{Setup} / (T_{Setup} + T_{Hold})$ | Setup/Hold ratio measurement with positive and/or negative clock edge.<br>See: "Setup/Hold measurement settings" on page 349 |
| | Delay to trigger | | Time between the trigger event and a following signal slope. High accuracy even if the trigger event is outside the acquisition data.<br>See:<br>• "Delay to trigger measurement settings" on page 351 |

### Area measurements

Access: [Measure] > "Meas Group" tab > "Amp/Time" category

Area measurements are voltage over time measurements.

*Table 8-3: Area measurements*

| | Meas. type | Symbol | Description/Result |
|---|---|---|---|
| | Area | $A_{Ref}$ | Area between the waveform and a reference level ("Area level", $X_{Ref}$).<br>$A_{Ref} = \frac{T_{Eval}}{N_{Eval}} \cdot \sum_{i=1}^{N_{Eval}} (x(i) - X_{Ref})$<br>$T_{Eval}$: Evaluation time, time of a full waveform or limited by a gate |
| | Cycle area | $A_{RefCyc}$ | Area between the waveform and a reference level ("Area level") measured for one period, see also "Area" measurement. The measurement requires at least one complete period of a triggered signal. Multiple measurements are possible. |

### Counting

Access: [Measure] > "Meas Group" tab > "Amp/Time" category

*Table 8-4: Counting measurements*

| | Meas. type | Symbol | Description/Result |
|---|---|---|---|
| | Pulse count | | The number of positive or negative pulses of the waveform, or of both positive and negative pulses.<br>The mean value of the signal is determined. If the signal passes the mean value, an edge is counted. A positive pulse is counted if a rising edge and a following falling edge are detected. A negative pulse is counted if a falling edge and a following rising edge are detected. |
| | Edge count | | The number of positive or negative edges, or of both positive and negative edges.<br>The instrument determines the mean value of the signal and counts an edge every time the signal passes the mean value. |

### Arithmetic

Access: [Measure] > "Meas Group" tab > "Amp/Time" category

*Table 8-5: Arithmetic measurements*

| | Meas. type | Symbol | Description/Result |
|---|---|---|---|
| | Arithmetic | | Allows you to perform basic arithmetic operations between selected measurement values and/or constant. |

## 8.2.5.2 Settings for amplitude/time measurements

Access: [Measure] > "Meas Group" tab > "Amp/Time" category

Amplitude and time measurements are available for sources in the time domain. For some amplitude/time measurements, such as delay, setup/hold and delay to trigger, further setting are required to get a measurement result.

If further settings are available for a measurement, a settings icon is shown beside the measurement's name in the "Active measurements" list. Tap the icon to configure the measurement.

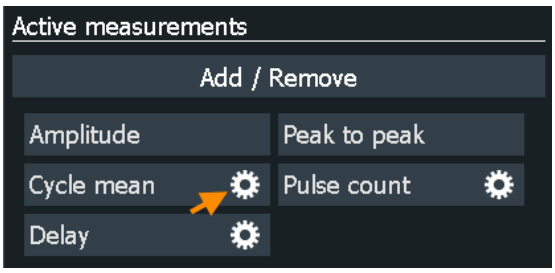



*Figure 8-7: Selected active measurements, some have further settings*

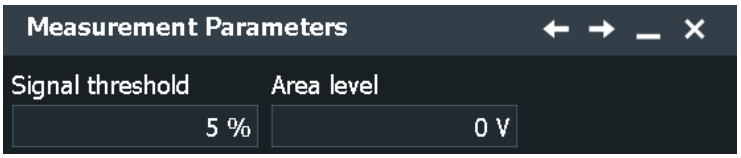



*Figure 8-8: Settings for cycle area measurement*

This chapter explains all settings for amplitude/time measurements. For a description of available measurement, see Chapter 8.2.5.1, "Overview of amplitude/time measurements", on page 341.

Signal threshold ..... 347
Area level ..... 347
Pulses slope ..... 348
Measured slope ..... 348
Settings for delay and phase measurements (analog sources) ..... 348
Settings for delay and phase measurements (digital sources) ..... 349
Setup/Hold measurement settings ..... 349
Clock ref level ..... 350
Data ref level / Reference level ..... 351
Pulse count ..... 351
Edges slope ..... 351
Delay to trigger measurement settings ..... 351

**Signal threshold**
The measured signal value must reach a certain voltage or bit to be included in the measurement algorithm. This minimum value is defined in percent of the signal amplitude (difference of high and low signal levels). Values below this value are considered to be noise and they are ignored. The setting is relevant for area, time, and counting measurements.

Remote command:
`MEASurement<m>:DETThreshold` on page 1440

**Area level**
The reference level used to integrate the waveform. The setting is only relevant for area measurements.

Remote command:
MEASurement<m>:AMPTime:ALEVel on page 1440

**Pulses slope**
Sets the first slope of the pulses to be counted.

The setting is available only for the "Pulse count" measurement.

"Positive" Positive pulses are counted.

"Negative" Negative pulses are counted.

"Either" Both positive and negative pulses are counted.

Remote command:
MEASurement<m>:AMPTime:PSLope on page 1441

**Measured slope**
Selects the slope direction for frequency and period measurements.

"Positive / Negative" Measures the time between rising or falling edges, respectively.

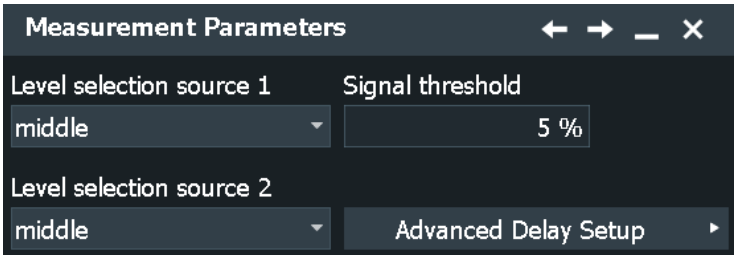
"Either" In multiple measurements, the time is measured both between rising edges and between falling edges.
In single measurements. The first edge is taken for the measurement.

"First edge" Time is measured either between rising edges or between falling edges. The first edge is taken for the measurement. In single measurements, it works the same way as "Either".
Only available for analog measurement sources.

Remote command:
MEASurement<m>:AMPTime:PFSLope on page 1440

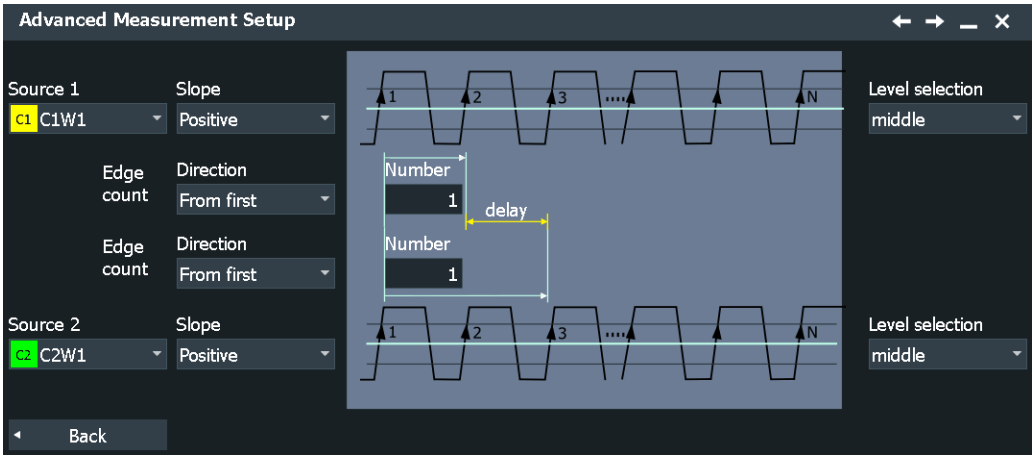
**Settings for delay and phase measurements (analog sources)**
The specific settings for delay measurement allow you to measure the time between any two slopes at any reference level. Therefore, the reference levels and the slopes must be defined for each source individually. The measurement result is negative if the edge of the second source comes before the edge of the first source.

"Level selection source 1/2": For each measurement source, select the reference level on which the time is measured.




"Advanced Delay Setup":

With the settings shown in the picture, the time between the second and the fifth rising edge is measured.




"Level selection" — Selects the reference level on which the time is measured.

"Slope" — Sets the edge of each source, between which the delay is measured: positive, negative, or either of them.

"Direction" — Selects the direction for counting slopes for each source: from the beginning of the waveform, or from the end.

"Number" — Sets the number of the edge that is relevant for delay measurement.

Remote command:

`MEASurement<m>:AMPTime:DELay<n>:LSELect` on page 1442

`MEASurement<m>:AMPTime:DELay<n>:SLOPe` on page 1443

`MEASurement<m>:AMPTime:DELay<n>:DIRection` on page 1441

`MEASurement<m>:AMPTime:DELay<n>:ECOunt` on page 1442

**Settings for delay and phase measurements (digital sources)**

Delay measurement on digital channels is reduced to measure the time between two subsequent rising or two subsequent falling edges.

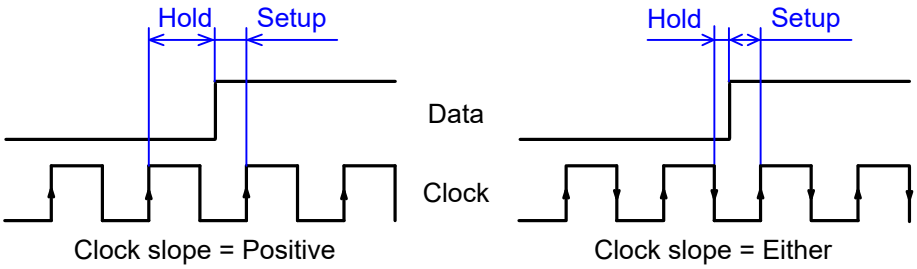
The edge direction is set with Edges slope.

**Setup/Hold measurement settings**

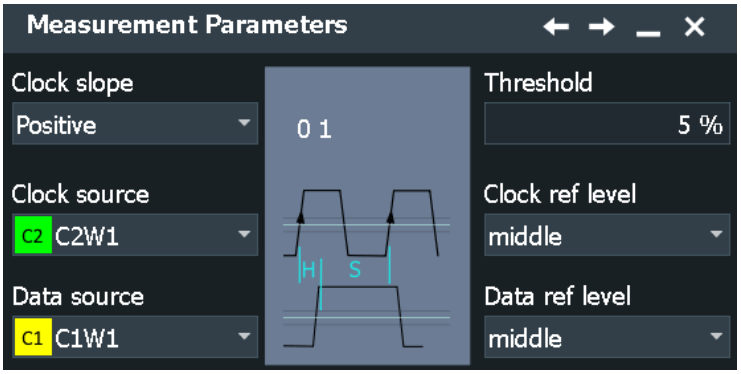
Setup/Hold measurements analyze the relative timing between two signals: a data signal and the synchronous clock signal. Setup time is the time that the data signal is steady before clock edge - the time between a data transition and the next specified clock edge. Hold time is the time that the data signal is steady after clock edge - the time between a data transition and the previous specified clock edge.

"Setup/Hold time" measures and displays the setup and hold durations. "Setup/Hold ratio" measurements return the ratio of the setup time to the sum of hold and setup time: $T_{Setup} / (T_{Setup} + T_{Hold})$.

The clock edge can be defined, the polarity of the data signal does not matter.




If at least one of the setup/hold measurements is selected, more settings appear to specify the measurement.




"Clock slope" Sets the edge of the clock from which the setup and hold times are measured: positive, negative, or either of them. If "Either" is selected, the clock edges next to the data edge are considered regardless of the clock slope.

"Clock source" The "Clock source" is identical to the measurement "Source". It defines the waveform used as clock in the setup/hold measurement.

"Data source" The "Data source" is identical to the "2nd Source" of the measurement. It sets the data signal.

"Clock ref level" See "Clock ref level" on page 350.

"Data ref level" See "Data ref level / Reference level" on page 351.

"Threshold" See "Signal threshold" on page 347.

Remote command:
Clock slope: `MEASurement<m>:AMPTime:CSLope` on page 1444

**Clock ref level**
Selects the reference level of the clock on which the time is measured. The intersection of slope and reference level defines the time point for measurements.

The setting is used for setup and hold measurements, and for jitter measurements (option R&S RTP-K12).

Remote command:
`MEASurement<m>:AMPTime:CLCK<n>:LSELect` on page 1444

**Data ref level / Reference level**
The setting defines the "Data ref level" for setup and hold measurements, and for time-interval error measurements (option R&S RTP-K12). It selects the reference level of the data signal on which the time is measured. The intersection of slope and reference level defines the time point for measurements.

For clock jitter measurements (option R&S RTP-K12), it sets the "Reference level" for the time measurement.

Remote command:
`MEASurement<m>:AMPTime:DATA<n>:LSELect` on page 1445

**Pulse count**
Sets the number N of positive pulses for the "Pulse train" measurement. This measurement measures the duration of N positive pulses from the rising edge of the first pulse to the falling edge of the N-th pulse.

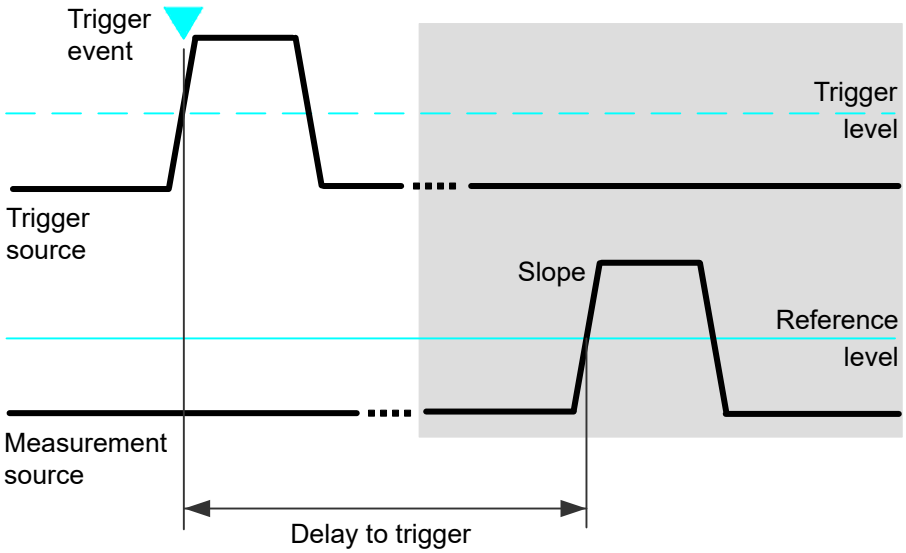
Remote command:
`MEASurement<m>:AMPTime:PTCount` on page 1443

**Edges slope**
Sets the edge direction to be considered. The setting is relevant for edge count measurement and delay measurement on digital channels.

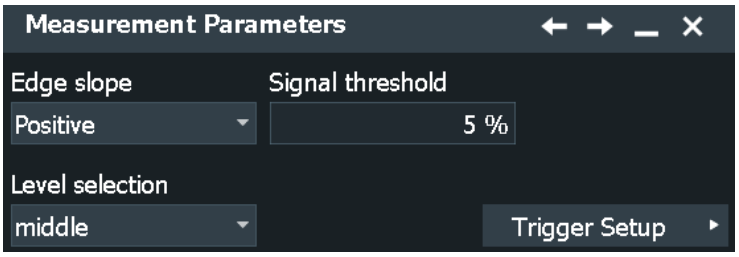
| | |
|---|---|
| "Positive" | Positive edges are considered. |
| "Negative" | Negative edges are considered. |
| "Either" | Both positive and negative edges are counted (edge count). Delay is measured either between rising edges or between falling edges. The first edge is taken for the measurement. |

Remote command:
`MEASurement<m>:AMPTime:ESLope` on page 1444

**Delay to trigger measurement settings**
Delay to trigger measures the time between the trigger point and the following slope of a waveform. The delay between the trigger and the slope can be high compared to the accuracy of the acquisition, and the trigger point can even be outside of the current acquisition.




To configure the trigger conditions, use the trigger setup. To set up the slope, additional settings appear in the measurements "Setup" dialog box.




"Edge slope" Sets the edge direction to be used for delay measurement: positive, negative, or either edge.

"Level selection" Selects the reference level of the measurement source on which the delay is measured: upper, middle, or lower level.

"Signal threshold"
See "Signal threshold" on page 347.

Remote command:
`MEASurement<m>:AMPTime:DTOTrigger<n>:SLOPe` on page 1445
`MEASurement<m>:AMPTime:DTOTrigger<n>:LSELect` on page 1445

### 8.2.5.3 Arithmetic for amplitude/time and spectrum measurements

Access: [Measure] > "Meas Group" tab > "Amp/Time"/"Spectrum" category > "Arithmetic".

In this dialog, you can set up basic arithmetic operations between selected measurement values and/or constant.

You can select only measurement groups and measurements that are already enabled.

**Type**
Selects the type of arithmetic measurement:

"Measurement" Performs an arithmetic operation between two measurement results. You can select from the measurements enabled for "Amp/Time" or "Spectrum".

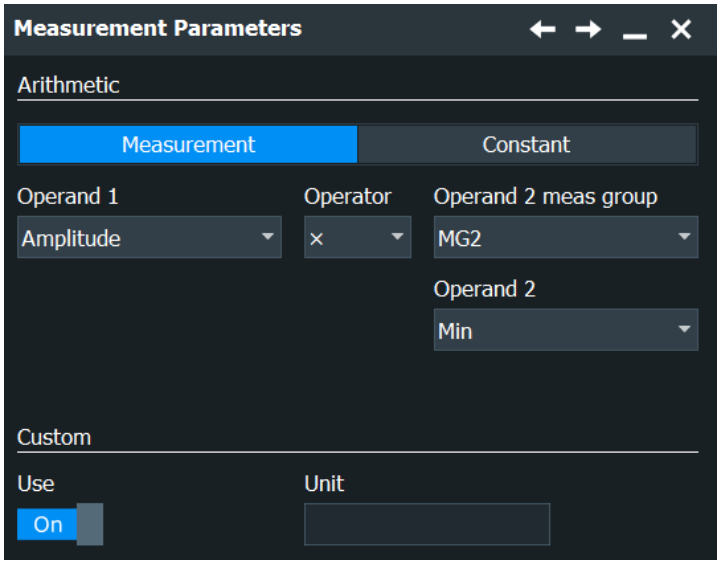



*Figure 8-9: Arithmetic with two measurement results*

"Constant" Performs an operation between a measurement result and a user defined fixed Constant.

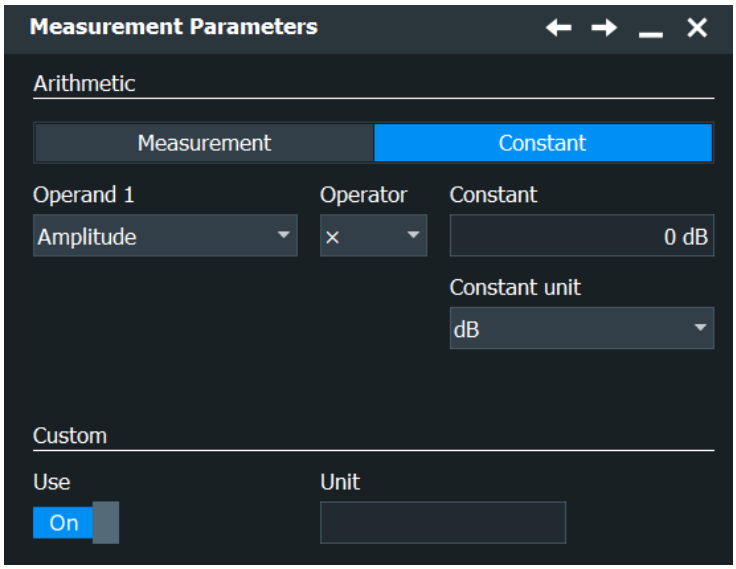



*Figure 8-10: Arithmetic with a measurement result and a constant*

Remote command:
`MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:TYPE` on page 1452

**Operand 1**
Selects the first operand for the arithmetics. You can select one of the measurements, that is enabled for the current measurement group.

Remote command:
MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:AMPTime on page 1454
MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:AMPTime on page 1450
MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:SPECtrum on page 1451
MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:SPECtrum on page 1455

**Operand 2 meas group / Operand 2**
Selects the measurement group and the measurement result for the second operand. You can select one of the measurements, that is enabled for the selected measurement group.

Remote command:
MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:GROup on page 1451
MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:GROup on page 1455

**Operator**
Selects the operation type to be performed on the selected operands. You can choose between +, -, * and /.

| | |
|---|---|
| "+" | Adds up the operands. |
| "-" | Subtracts "Operand 2"/"Constant" from "Operand 1". |
| "*" | Multiplies the operands. |
| "/" | Divides "Operand 1" by "Operand 2"/"Constant". |

Remote command:
MEASurement<m>:AMPTime:ARIThmetic:OPERator on page 1450

**Custom**
Enable "Use" to apply the user-defined value for the "Unit".

Remote command:
MEASurement<m>:AMPTime:ARIThmetic:CUNState on page 1449
MEASurement<m>:SPECtrum:ARIThmetic:CUNState on page 1453

**Constant**
Sets a constant numeric value for the arithmetic operation, for the "Constant" measurement type.

Remote command:
MEASurement<m>:AMPTime:ARIThmetic:CONStant on page 1448
MEASurement<m>:SPECtrum:ARIThmetic:CONStant on page 1452

**Constant unit**
Selects the unit for the Arithmetic for amplitude/time and spectrum measurements.

Remote command:
MEASurement<m>:AMPTime:ARIThmetic:COUNit on page 1448
MEASurement<m>:SPECtrum:ARIThmetic:COUNit on page 1453

#### 8.2.5.4 Measuring the delay to trigger

Delay to trigger measures the time between the trigger point and the following slope of a waveform. If the delay is unknown, it can be measured in two stages - first a coarse and then a precise measurement.

See also: "Delay to trigger measurement settings" on page 351.

1. Set the horizontal scale and horizontal position so that the trigger point and the slope both are visible on the screen.
2. Select the delay to trigger measurement:
   a) In the "Meas Group" dialog, set the "Category" to "Amp/Time".
   b) Press "Add/ Remove Measurements" button.
   c) In the dialog, enable "Delay to trigger".
   d) Tap "Back".
3. Configure the "Delay to trigger" measurement:
   a) In the "Active Measurements" list, tap the "Delay to trigger" measurement.
   b) Select the source, that is the waveform with the delayed slope.
   c) Select the slope and the reference level.
   d) Check the trigger settings.
4. Enable the measurement. Note the result.
5. Turn the horizontal [Position] knob and enter the measured delay as horizontal position.

   Thus, the slope is moved to the center of the screen.
6. Adjust the time scale ("Menu" > "Horizontal" > "Setup" tab) and the sample rate and resolution ("Menu" > "Acquire" > "Setup" tab) to the required resolution.

   The trigger is outside the display and is not part of the current acquisition.
7. Repeat the "Delay to trigger" measurement.

   Now the delay is measured with high accuracy. You can analyze the variance of delay values using statistical evaluation and histogram functions.

### 8.2.6 Eye diagram measurements

The eye diagram is a tool for evaluation of signal quality and shows the combined effects of channel noise and intersymbol interference. It is a significant means of visualizing jitter and allows you to analyze the reasons for it. By creating histograms of the eye diagram, important jitter parameters can be determined.

The eye diagrams are a superposition of repetitively sampled waveforms, which have a length of about 1 bit.

There are two ways to generate eye diagrams on the R&S RTP:

- Standard eye diagrams, which are a superposition of repetitively sampled digital data.
- To use advanced eye diagrams, option R&S RTP-K136/K137 or R&S RTP-K91/K93 is required.

The waveform display style must be set to vectors: "Menu" > "Settings" > "Display" > "Signal" > "Waveform style" = "Vectors".

To obtain optimized settings for an eye measurement, use the "Autoset" function that is provided on the right side of the "Eye" tab.

The following characteristic values can be determined:

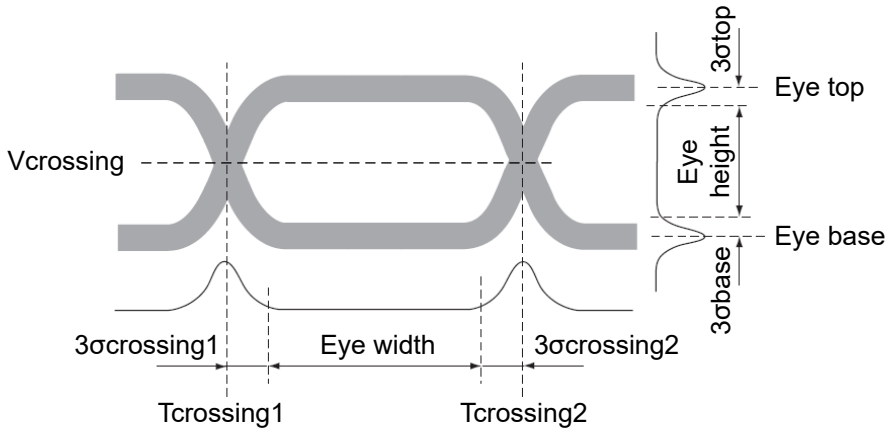



***Figure 8-11: Basic eye diagram characteristics***

Eye top = Mean of the upper vertical histogram
σtop = Standard deviation of the upper vertical histogram
Eye base = Mean of the lower vertical histogram
σbase = Standard deviation of the lower vertical histogram
Tcrossing = First and second mean of the horizontal histogram
σcrossing = Standard deviation of the horizontal histogram

### 8.2.6.1 Overview of eye diagram measurements

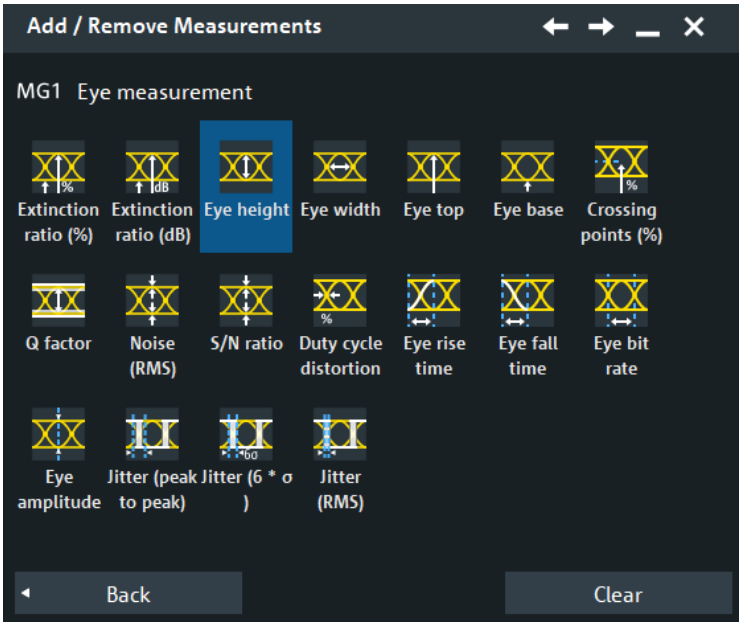



*Table 8-6: Eye measurements*

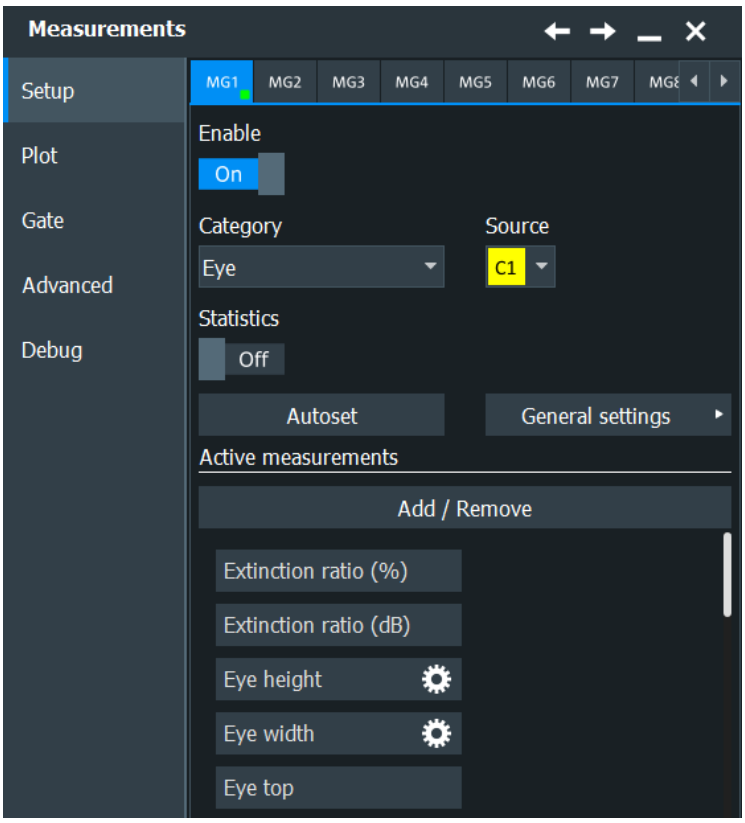
| | Meas. type | Description/Result |
|---|---|---|
| | Extinction ratio (%) | The extinction ratio is an indication of efficiency. It describes the ratio of the power used to transmit a logic level 1, to the energy used to transmit a logic level 0. The R&S RTP provides extinction ratio measurements as a percentage, and in decibels:<br>ER (%) = Eye base / Eye top *100<br>Prerequisite: Eye base > 0 and Eye top > 0 because extinction ratio is defined only for positive values. |
| | Extinction ratio (dB) | ER (dB) = 10*log (Eye top / Eye base) |
| | Eye height | The vertical eye opening indicates the sensitivity of the transmission to noise.<br>(Eye top – 3 * σtop) – (Eye base + 3 * σbase) |
| | Eye width | The horizontal eye opening indicates the time range during which the sampling of the logical state is possible.<br>(Tcrossing2 – 3 * σcrossing2) – (Tcrossing 1 – 3 * σcrossing1) |
| | Eye top | Mean of the upper vertical histogram |
| | Eye base | Mean of the lower vertical histogram |
| | Q factor | Q factor = (Eye top – Eye base) / (σtop + σbase) |

| | Meas. type | Description/Result |
|---|---|---|
| | Noise (RMS) | Quadratic mean of the noise of eye top and eye base<br>Noise (RMS) = (σtop + σbase)/2 |
| | S/N ratio | Signal-to-noise ratio<br>SNR = 10 * log (Eye amplitude / Noise RMS) |
| | Duty cycle distortion | Duty cycle distortion = 20 * log (Eye amplitude / Noise RMS) |
| | Eye rise time | Duration for signal to rise from 10% to 90% of the high signal level |
| | Eye fall time | Duration for signal to fall from 90% to 10% of the high signal level |
| | Eye bit rate | Frequency between two crossings |
| | Eye amplitude | Eye top - Eye base |
| | Jitter (peak to peak) | Average of the jitter for both crossing points.<br>Jitter = (σcrossing1 + σcrossing2) / 2 |
| | Jitter (6*σ) | Jitter (6*σ) = Jitter * 6 |
| | Jitter (RMS) | Quadratic mean of the jitter at both crossing points |
| | Crossing points (%) | Percentage of points crossing the eye |

### 8.2.6.2 Settings for eye diagram measurements

Access: [Measure] > "Meas Group" tab > "Eye" category

Eye diagram measurements are only available for sources in the time domain.

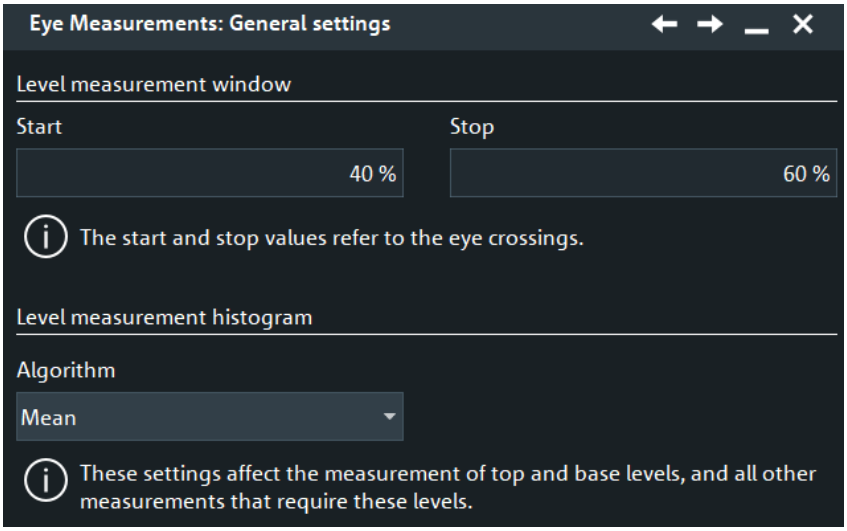



To obtain optimized settings for an eye measurement, use the "Autoset" function that is provided on the right side of the tab.

**Autoset**
Defines optimized settings for eye diagram measurements on the selected source.

Remote command:
`MEASurement<m>:EYEJitter:AUToscale` on page 1458

**General**
The settings in this dialog are relevant for the following eye measurements: "Extinction ratio (%)", "Extinction ratio (dB)", "Q factor", "Eye height", "Noise (RMS)", "S/N ratio", "Eye amplitude".




"Start" Defines the global start for the eye measurements.

"Stop" Defines the global stop for the eye measurements.

"Algorithm" Defines how to compute the top/base level and the corresponding sigma values.

- "Mean": the mean value of the histogram. The Sigma value is the standard deviation of the histogram.
- "Peak": the maximal peak of the histogram. Only values corresponding to this peak are used to compute sigma.
- "Inner peak": the inner peak of the histogram. Only values corresponding to this peak are used to compute sigma.
- "Outer peak": the outer peak of the histogram. Only values corresponding to this peak are used to compute sigma.

"Threshold" Sets the threshold used to separate peaks: two peaks are separated if they are split by a valley of height *Threshold * Maximal_histogram_value*.

Remote command:

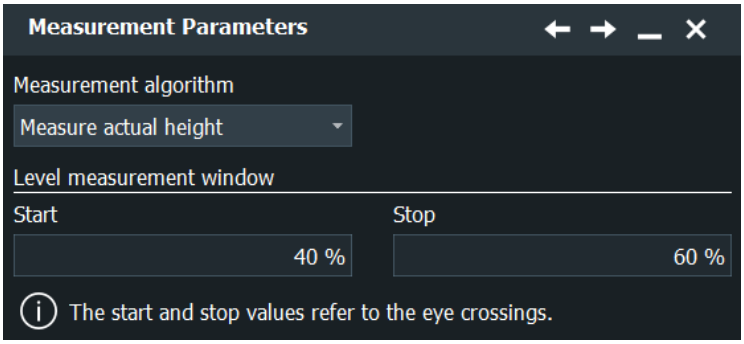
`MEASurement<m>:EYEJitter:GENSettings:RELative:STARt` on page 1460

`MEASurement<m>:EYEJitter:GENSettings:RELative:STOP` on page 1461

`MEASurement<m>:EYEJitter:GENSettings:MLEHistogram:ALGorithm` on page 1460

`MEASurement<m>:EYEJitter:GENSettings:MLEHistogram:THReshold` on page 1460

### Eye height




In this dialog, you can define the settings for the eye height measurements.

"Measurement algorithm" Selects the algorithm for the measurement of the height.
If "Meas actual height" is selected, you can define a measurement window for the measurement. The default window is 40% - 60% of the eye period.
If "Extrapolate using three sigma" is selected, the eye height is measured with the settings defined in the General dialog.
Eye height=(Eye top-3σtop)-(Eye base+3σbase)

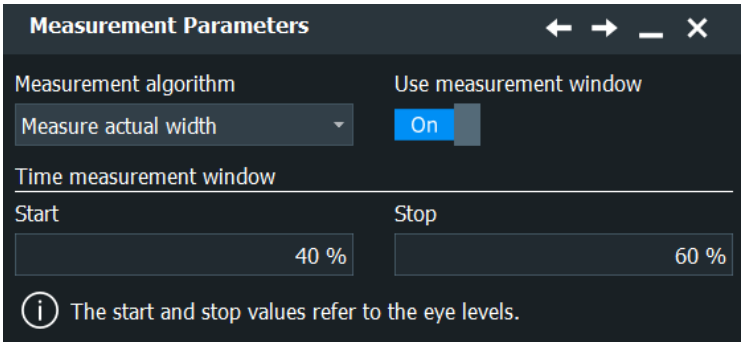
"Start" Defines the start time of the range for the eye measurement height.

"Stop" Defines the stop time of the range for the eye measurement height.

Remote command:
`MEASurement<m>:EYEJitter:EYEHeight:ALGorithm` on page 1463
`MEASurement<m>:EYEJitter:EYEHeight:RELative:STARt` on page 1463
`MEASurement<m>:EYEJitter:EYEHeight:RELative:STOP` on page 1464

### Eye width

In this dialog, you can define the settings for the eye width measurements.




"Measurement algorithm" Selects the algorithm for the measurement of the height.
If "Meas actual height" is selected, you can define the time measurement window. The default window is 40% - 60% of the eye period.
If "Extrapolate using three sigma" is selected, the eye width is measured at the crossing level:
Eye width=(Tcrossing2-3σcrossing2)-(Tcrossing1+3σcrossing1)

"Use measurement level" If enabled, you can define a user defined "Start" and "Stop" for the eye width measurement.

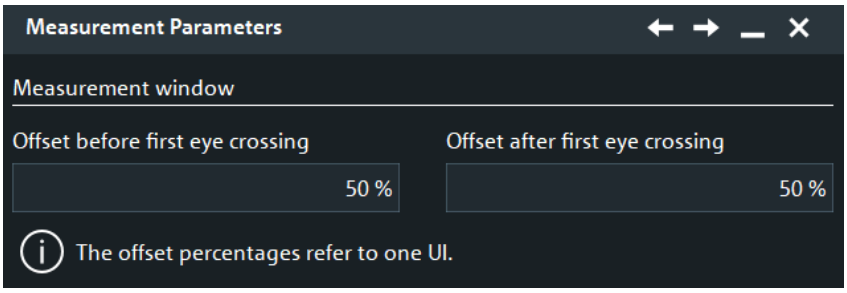
"Start" Defines the start time of the range for the eye measurement width.

"Stop" Defines the stop time of the range for the eye measurement width.

Remote command:
MEASurement<m>:EYEJitter:EYEWidth:ALGorithm on page 1461
MEASurement<m>:EYEJitter:EYEWidth:MEASwindow on page 1462
MEASurement<m>:EYEJitter:EYEWidth:RELative:STARt on page 1462
MEASurement<m>:EYEJitter:EYEWidth:RELative:STOP on page 1463

**Eye rise time,Eye fall time**
In this dialog, you can define the settings for the "Eye rise time" and "Eye fall time" measurements.

Only values inside the specified horizontal range are considered for the measurements. The range is marked with green lines if the result lines are shown.

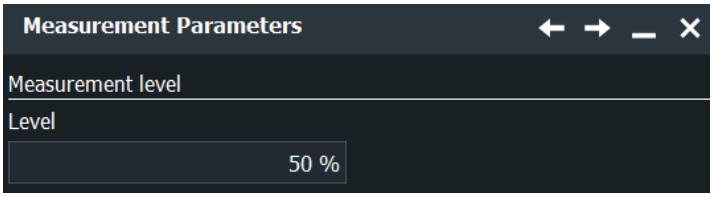



"Offset before first eye crossing" Sets the local start level of the range for eye rise/fall time measurements.

"Offset after first eye crossing" Sets the local stop level of the range for eye rise/fall time measurements.

Remote command:
MEASurement<m>:EYEJitter:RTFTime:RELative:STARt on page 1465
MEASurement<m>:EYEJitter:RTFTime:RELative:STOP on page 1465

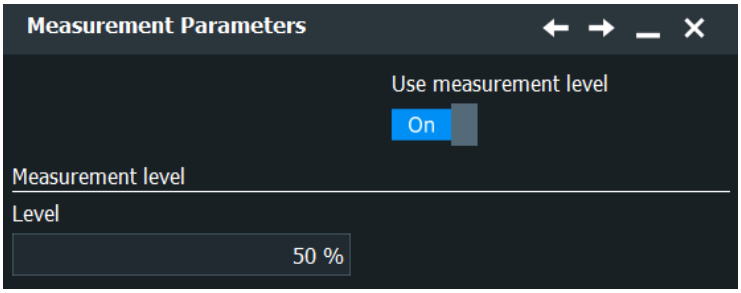
**Duty cycle disortion**
In this dialog, you can define the settings for the duty cycle distortion measurements.




"Level" Defines the level for the eye measurement jitter.

Remote command:
MEASurement<m>:EYEJitter:DCDistortion:RELative:LEVel on page 1461

**Jitter**
The settings in this dialog are available for "Jitter (peak to peak)", "Jitter (RMS)", "Jitter".




"Use measurement level" — If enabled, you can define a user defined "Level" for the eye measurement jitter.

"Level" — Defines the level for the eye measurement jitter.

Remote command:
`MEASurement<m>:EYEJitter:JITTer:MEASlevel` on page 1464
`MEASurement<m>:EYEJitter:JITTer:RELative:LEVel` on page 1465

## 8.2.7 Spectrum measurements

Spectrum analysis determines the frequencies of a given input signal over time. Various measurements can then be performed based on the signal spectrum.

### 8.2.7.1 Overview of spectrum measurements

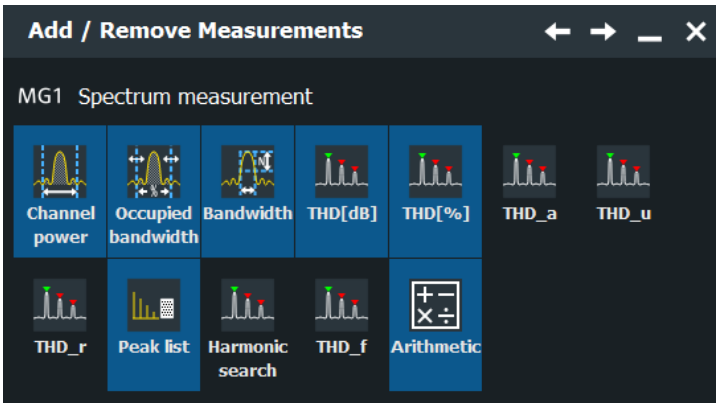


*Table 8-7: Spectrum measurements*

| | Meas. type | Description, result |
|---|---|---|
| | Channel power | Power integrated over the sample values defined by a center frequency and a bandwidth; based on a defined impedance. The result is given in dBm.<br>To get best results, enable "Measure all events in each acquisition" on page 391, and set the "Limit" to the maximum value. In particular, these settings are important when measuring pulsed signals. |
| | Occupied bandwidth | From the defined center frequency, symmetric sample value pairs to the left and right are integrated until a user-defined percentage of the total power is reached. The occupied bandwidth is the difference between the frequencies at which the requested power was reached. |
| | Bandwidth | n dB down bandwidth; the samples to the left and right of the peak value are analyzed until the n dB threshold is exceeded. The frequencies at which the threshold is exceeded define the limits of the requested bandwidth. |
| | THD[dB], THD[%]<br>Total harmonic distortion | Power sum of the harmonic waves divided by the power of the fundamental wave:<br>$THD = \frac{\sum_{n=2}^{\infty} P_n}{P_1}$ |
| | THD_f, THD_a, THD_u, THD_r<br>Total harmonic distortion | These measurements require option R&S RTP-K37 Spectrum analysis.<br>THD_f is the root mean square of the sum of all amplitudes of the harmonic waves in relation to the amplitude at the fundamental frequency (first harmonic):<br>$THD_F = \frac{\sqrt{\sum_{i=2}^{n} U_i^2}}{U_1}$<br>THD_a corresponds to THD[dB]:<br>$THD_a = \frac{\sum_{i=2}^{n} U_i^2}{U_1^2}$<br>THD_u:<br>$THD_u = \frac{\sqrt{U^2 - U_1^2}}{U_1}$<br>Distortion factor:<br>$THD_R = \frac{\sqrt{U^2 - U_1^2}}{U}$<br>Where:<br>• $U_i$: effective value of the harmonic with index i<br>• $U_1$: effective value of the first harmonic<br>• U: effective value of the signal |

| | Meas. type | Description, result |
|---|---|---|
| | Peak list | Table with measured peaks. For each peak, the frequency and the power value are listed in a table row. The number of determined peaks can be defined. You can sort the results by frequency or power value, and the peak labels are adjusted accordingly. |
| | Harmonic search | Table with measured harmonics. For each harmonic, the frequency and the value are listed in a table row. |
| | Arithmetic | Allows you to perform basic aritmetic operations between selected measurement values and/or constant.<br>For details see: Chapter 8.2.5.3, "Arithmetic for amplitude/time and spectrum measurements", on page 352 |

For remote command parameters and suffix types, see Table 24-8.

### 8.2.7.2 Settings for spectrum measurements

Access: [Measure] > "Meas Group" tab > "Spectrum" category

Spectrum measurements require a source in the frequency domain, i.e. a math waveform with an FFT operation.

For spectrum measurements, make sure that the start frequency and other FFT parameters are set correctly, and the fundamentals are not covered by the DC component of the signal. Consider also a gated measurement if the instrument cannot return any result.

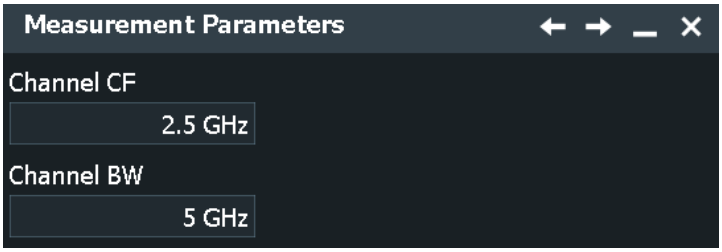



*Figure 8-12: Parameters of channel power measurement*

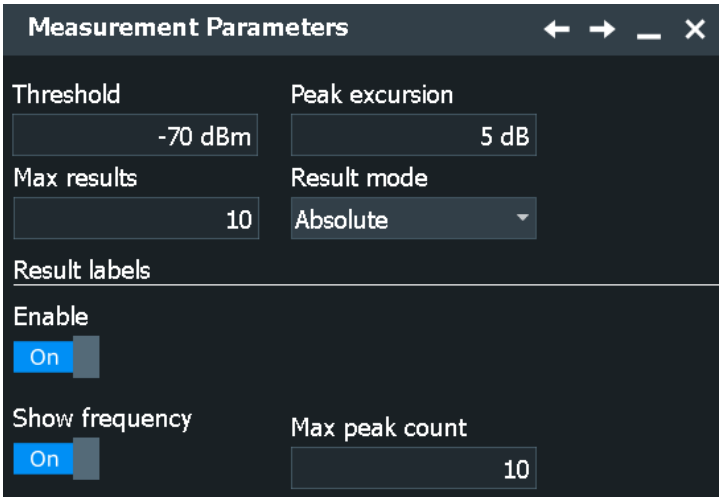



*Figure 8-13: Parameters of peak list measurement*

**Enable**
Enables the peak list measurement for the spectogram.

**N db down**
The threshold until which the samples to the left and right of the peak value are analyzed to determine the "Bandwidth".

Remote command:
`MEASurement<m>:SPECtrum:NDBDown` on page 1468

**Channel BW**
Bandwidth over which the channel power is calculated.

Remote command:
`MEASurement<m>:SPECtrum:CPOWer:BANDwidth` on page 1467

**Channel CF**
Center frequency from which the channel power is calculated over the specified bandwidth.

Remote command:
`MEASurement<m>:SPECtrum:CPOWer:CFRequency` on page 1468

**Occup. BW**
Percentage of the total power used to determine the occupied bandwidth.

Remote command:
`MEASurement<m>:SPECtrum:OBANdwidth` on page 1467

**Threshold**
Defines an absolute threshold as an additional condition for the peak search. Only peaks that exceed the threshold are detected.

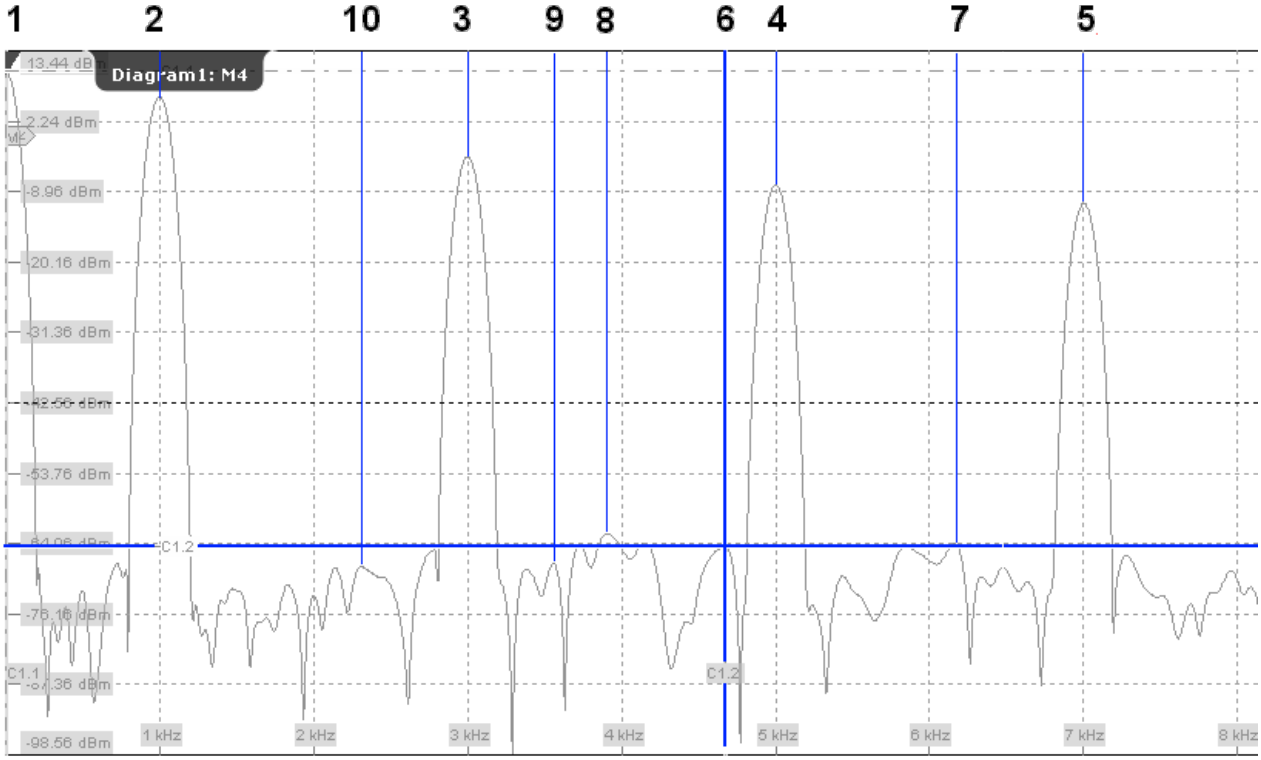
This setting is only available for spectrum waveforms. It is valid for cursor measurements, spectrum measurements and peak search.

Remote command:
CURSor<m>:THReshold on page 1594
MEASurement<m>:SPECtrum:ATHReshold on page 1469

**Peak excursion**
Defines a relative threshold, the minimum level value by which the waveform must rise or fall to be considered as a peak. To avoid identifying noise peaks, enter a peak excursion value that is higher than the noise levels.

This setting is only available for spectrum waveforms. It is valid for cursor measurements, spectrum measurements and peak search.

The following figure shows a cursor measurement on a spectrum waveform:




If "Peak excursion" is 30 dB, the peaks 1 to 5 are found. If "Peak excursion" is 20 dB, also the peaks 6 to 10 are found. The cursor position is on peak 6.

Remote command:
CURSor<m>:PEXCursion on page 1594
MEASurement<m>:SPECtrum:PEXCursion on page 1468

**Max results**
Sets the maximum number of measurement results that are listed in the result table. Available for peak list and harmonic search measurements.

Remote command:
MEASurement<m>:SPECtrum:RESult<n>:COUNt on page 1469

**Result mode**
Selects the the way the measurement results are displayed. Available for peak list and harmonic search measurements.

"Absolute" The harmonics/peaks are shown in absolute value, dBm.

"Relative" The level of the carrier is shown in absolute value dBm. The values the harmonics/peaks are shown relatively to the carrier in dBc.

Remote command:
`MEASurement<m>:SPECtrum:RESult<n>:MODE` on page 1469

**Result labels**
For peak lists only: Use labels to describe the detected peaks in the spectrum diagram.

You can configure what is displayed in the labels. To change the label design, use "Menu" > "Settings" > "Appearance" > "Peak List", see Chapter 4.3.9, "Peak list appearance settings", on page 90.

"Enable" Displays a description for each detected peak in the spectrum diagram.

"Show frequency" Includes the frequency of the detected peak in the diagram labels.

"Max. peak count" Defines the maximum number of peaks that are labeled in the diagram. The result table lists all peaks.

Remote command:
`MEASurement<m>:RESult:SHLabels` on page 1471
`MEASurement<m>:RESult:LABorder` on page 1224
`MEASurement<m>:RESult:INVerse` on page 1225
`MEASurement<m>:RESult:MAXCount` on page 1470
`MEASurement<m>:RESult:SHFRequency` on page 1470

## 8.2.8 Histograms and histogram measurements

### 8.2.8.1 Histogram characteristics

Histograms are used to plot density of data, i.e. to display graphically how often which signal values occur. The histogram can be based on the input signal levels (amplitudes) or the timebase in a time domain measurement, or on frequencies or frequency levels in a spectrum measurement. They are a prerequisite for histogram measurements.

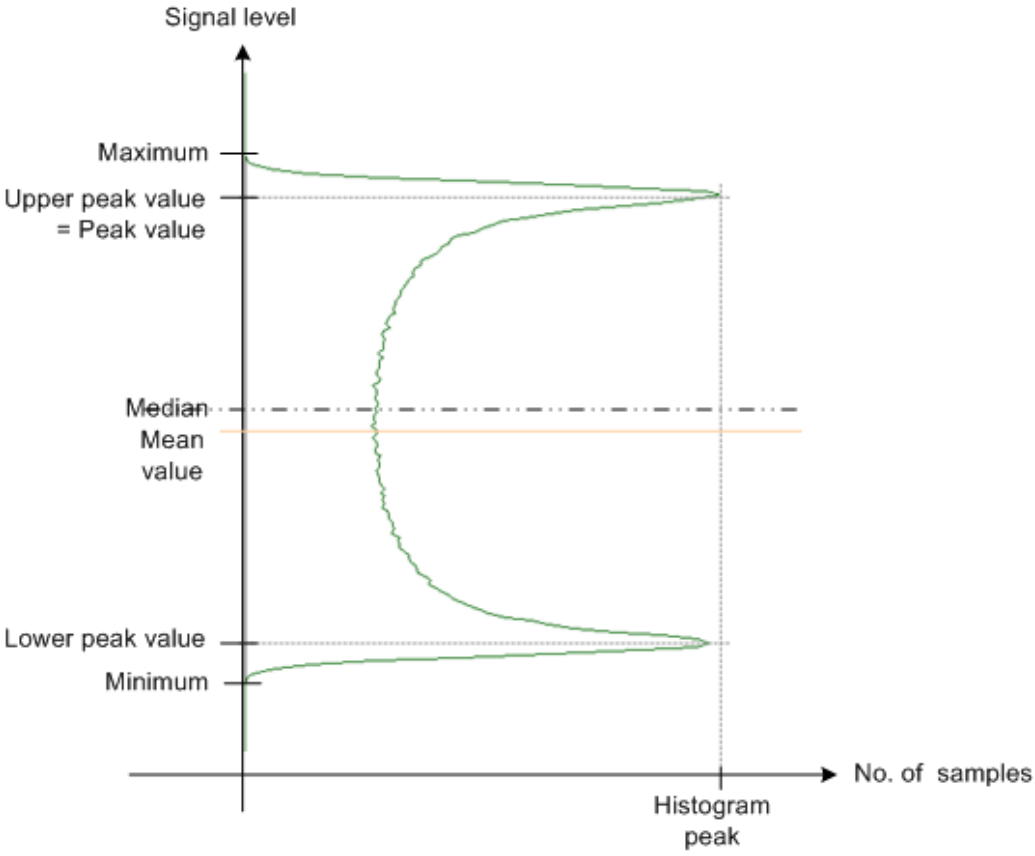
Depending on which data the histogram is based on, a vertical or horizontal histogram can be selected. A vertical, or amplitude, histogram displays horizontal bars across amplitude values. A horizontal or time/frequency histogram displays vertical bars over time/frequencies.

You can define up to 8 histograms in a diagram, one of them is displayed. They can be created quickly using toolbar icons, or in the "Meas" menu > "Histogram" dialog box. To switch the histogram display, tap the required histogram area, or select it in the "Histo-

gram" dialog box. For histogram measurements, the measured histogram is selected independently in the measurement setup.

In a histogram, the maximum count of a waveform value is assigned to the full height (histogram peak). All other count values are displayed relative to the maximum.

The following characteristic values can be determined for histograms (illustrated for a vertical histogram):




### 8.2.8.2 Overview of histogram measurements

There are various measurements available to analyze histograms.

Note that graphical changes is the diagram, e.g. changing the diagram size or the size of the result table, reset the histogram measurements in run and stop mode.

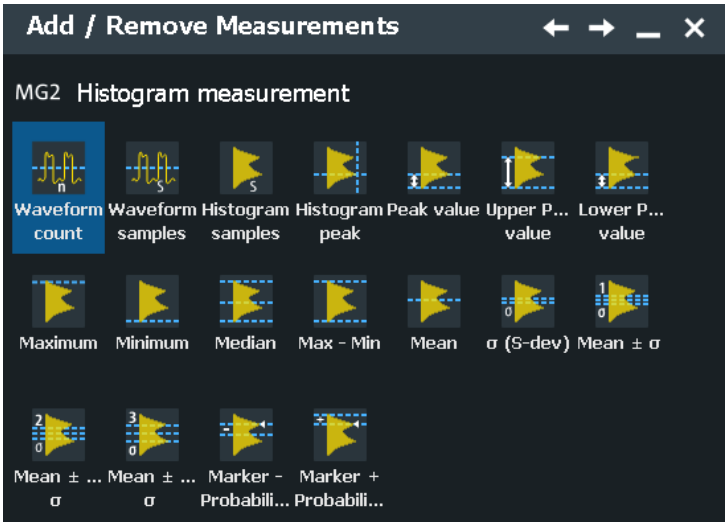


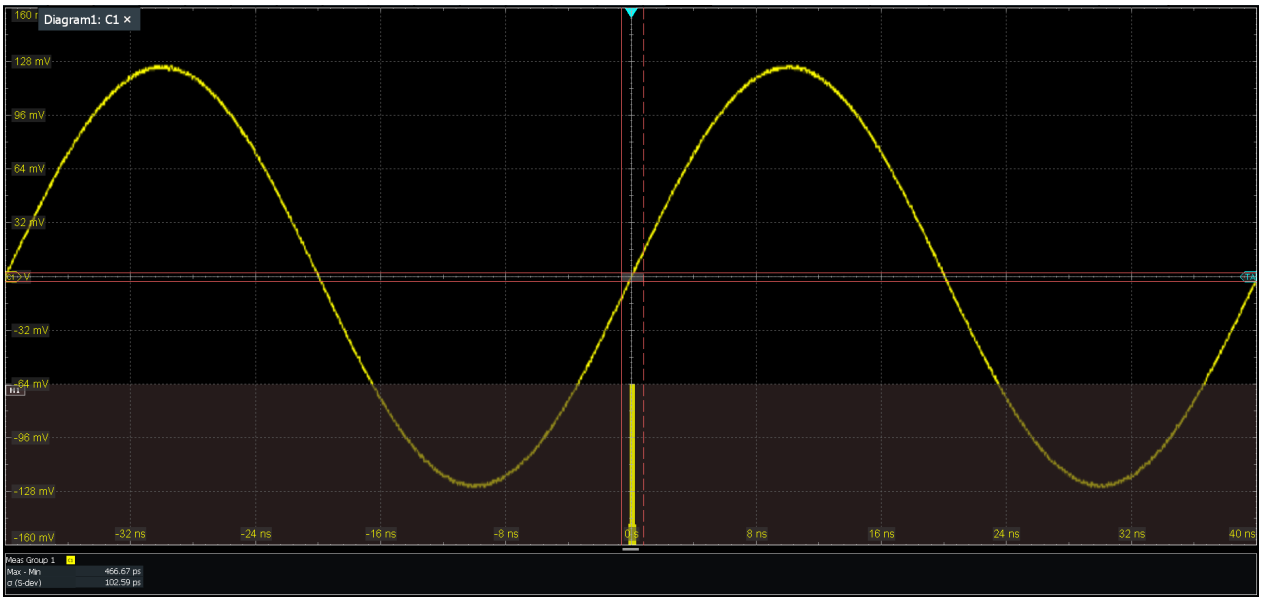

*Table 8-8: Histogram measurements*

| | Meas. type | Description/Result |
|---|---|---|
| | Waveform count | The number of acquisitions (waveforms) the histogram is based on |
| | Waveform samples | The number of samples from the most recent acquisition included in the current histogram |
| | Histogram samples | The number of samples from all acquisitions included in the current histogram |
| | Histogram peak | The maximum count value in the histogram |
| | Peak value | The signal value at the histogram peak |
| | Upper peak value | The signal value at the maximum count value in the upper half of the histogram |
| | Lower peak value | The signal value at the maximum count value in the lower half of the histogram |
| | Maximum | The highest signal value with a probability > 0 |
| | Minimum | The lowest signal value with a probability > 0 |

| | Meas. type | Description/Result |
|---|---|---|
| | Median | The signal value for which half the samples lie above, the other half below in the histogram<br>The sample count of one signal value after the other are accumulated until half the total number of samples in the histogram is reached. The signal value for which 50% of the samples are accumulated is the median. |
| | Max - Min | The range of signal values with a probability > 0 |
| | Mean | The weighted arithmetic average of the histogram |
| | σ (S-dev) | Standard deviation of the sample numbers |
| | Mean ±σ | The range between (mean value + standard deviation) and (mean value - standard deviation) |
| | Mean ±2*σ | The range between (mean value + 3 * standard deviation) and (mean value - 2 * standard deviation) |
| | Mean ±3*σ | The range between (mean value + 3 * standard deviation) and (mean value - 2 * standard deviation) |
| | Marker + Probability % | The marker value (according to the selected probability domain marker type) plus the defined limit.<br>Note that the value is restricted to the histogram range. |
| | Marker - Probability % | The marker value (according to the selected probability domain marker type) minus the defined limit.<br>Note that the value is restricted to the histogram range. |

**Rough jitter evaluation using a histogram**

You can use a horizontal histogram to perform a rough jitter measurement. Define a histogram for a narrow amplitude range close to the trigger time. The "Max-Min" value indicates the peak jitter, while the "StdDev" value indicates the RMS jitter.



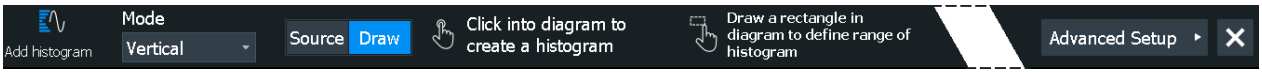

In addition to histograms on channel, math and reference waveforms, histograms can be created based on statistic measurement results. These histograms are enabled in the "Result Analysis" tab, see Chapter 8.2.12, "Result analysis", on page 384.

### 8.2.8.3 Creating histograms

Histograms can be used to evaluate the sample value occurrences directly. They are a prerequisite for histogram measurements.

**To create a histogram quickly with toolbar icons**

1. Select the waveform for which you need a histogram.
2. Tap the histogram icon on the toolbar.
3. Select the histogram type on the overlay menu: "Mode" = vertical for an amplitude histogram, or horizontal for a time-based histogram.




4. Tap the diagram with the waveform to be measured, or draw a rectangle on the screen to define the area for histogram calculation.
   Alternatively, tap "Source" and select the waveform.

   The histogram range is indicated in the diagram and a histogram with the selected waveform as a source is defined and displayed.

5. To adjust the histogram range:
   a) Double-tap the histogram.
   b) Tap "Advanced Settings" on the overlay menu.
   c) Adjust the start and stop values.

**To create and configure a histogram in the dialog box**

1. Select "Menu" > "Apps" > "Analysis" tab > "Histogram".

   The "Histogram" dialog box is displayed.

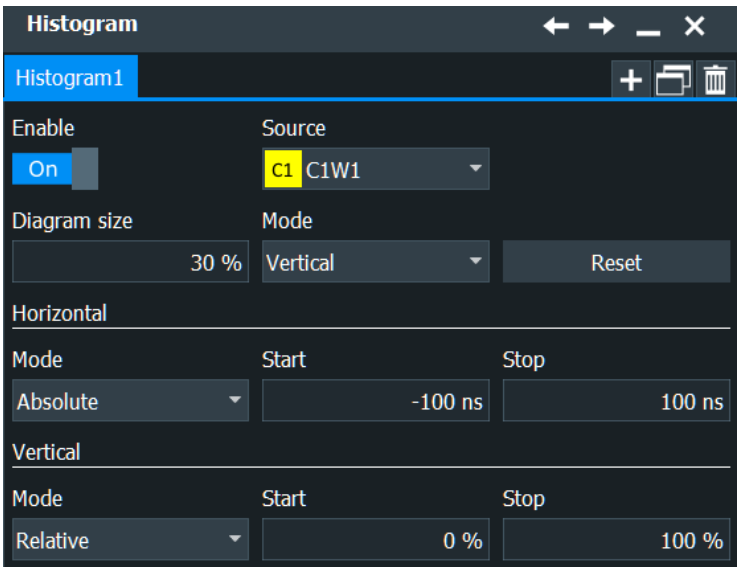
2. To create a histogram, tap the "Add" icon in the upper right corner of the dialog box.

3. To copy an existing histogram and configure a new one based on those settings, tap the "Copy" icon.
4. To change the name of a histogram, double-tap the tab label. Enter a name for the histogram using the on-screen keyboard.
5. Select a "Source" for the histogram. The source can be any input signal, math or reference waveform.
6. Define the histogram "Mode": vertical for an amplitude, horizontal for a time-based histogram.
7. Define the range of the waveform for which the histogram is to be generated. Enter the start value and the stop value in x and in y direction, either as absolute or relative values.
8. Enable the histogram.

### 8.2.8.4 Histogram setup

"Menu" > "Apps" > "Analysis" tab > "Histogram"

In this dialog box, you configure histograms on which you can perform further measurements.




**Enable**
Enables or disables the histogram evaluation and display. The histogram settings are kept until the histogram is deleted.

Remote command:
`LAYout:HISTogram:ENABle` on page 1477

**Source**
Defines the source of the histogram. Any analog channel waveform, math or reference waveform can be selected. Also measurements can serve as histogram source. In this case, the density distribution of the results of the main measurement is displayed.

Remote command:
`LAYout:HISTogram:SOURce` on page 1478

**Diagram size**
Defines the size of the histogram in percent of the diagram.

**Mode**
Defines the type of histogram.

"Vertical" Amplitude histogram (horizontal bars across amplitude)

"Horizontal" Time histogram (vertical bars over time). For spectrum waveforms, horizontal histograms over spectrum are not available.

Remote command:
`LAYout:HISTogram:MODE` on page 1478

**Reset**
Resets the values to begin a new histogram.

Remote command:
`LAYout:HISTogram:RESet` on page 1482

**Mode (Horizontal / Vertical)**
Defines whether the value range limits are entered as absolute or relative values.

Remote command:
LAYout:HISTogram:HORZ:MODE on page 1479
LAYout:HISTogram:VERTical:MODE on page 1480

**Start, Stop (Horizontal)**
Defines the horizontal value range of the histogram.

If advanced eye is enabled, the unit of the horizontal "Start", "Stop" (Horizontal) depends on the selected Time units. The unit can be absolute time or unit interval.

Remote command:
LAYout:HISTogram:HORZ:ABSolute:STARt on page 1479
LAYout:HISTogram:HORZ:ABSolute:STOP on page 1479
LAYout:HISTogram:HORZ:RELative:STARt on page 1480
LAYout:HISTogram:HORZ:RELative:STOP on page 1480

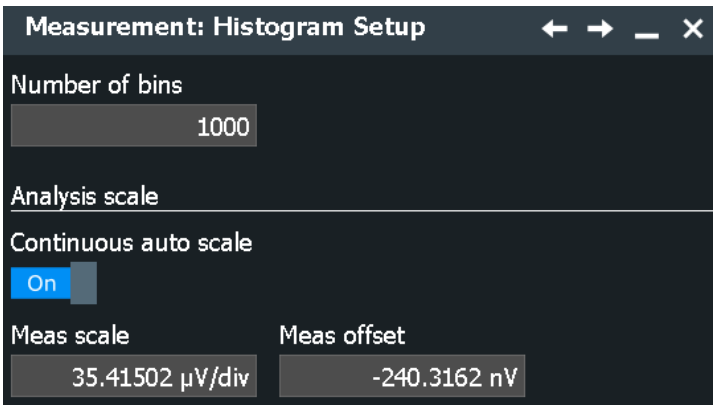
**Start, Stop (Vertical)**
Defines the vertical value range of the histogram.

Remote command:
LAYout:HISTogram:VERTical:ABSolute:STARt on page 1481
LAYout:HISTogram:VERTical:ABSolute:STOP on page 1481
LAYout:HISTogram:VERTical:RELative:STARt on page 1481
LAYout:HISTogram:VERTical:RELative:STOP on page 1482

### 8.2.8.5 Settings for histogram measurement

Access: [Measure] > "Meas Group" tab > "Histogram" category

You can perform measurements on histograms. Before, you have to define a histogram, see Chapter 8.2.8.3, "Creating histograms", on page 372.




The measurement parameters are common for all histograms measurements, you find them directly on the "Meas Group" tab.

**Source**
Selects the histogram on which the measurement is based. Before, you have to define a histogram, see Chapter 8.2.8.3, "Creating histograms", on page 372.

Remote command:
`MEASurement<m>:HISTogram:SELect` on page 1484

**Probability domain marker, Reference**
Defines the marker reference in the probability domain.

| | |
|---|---|
| "Peak" | The y-value with the maximum sample value in the histogram |
| "Upper Peak" | The y-value at the maximum sample value in the upper half of the histogram |
| "Lower Peak" | The y-value at the maximum sample value in the lower half of the histogram |
| "Maximum" | The highest y-value with a probability > 0 |
| "Minimum" | The lowest y-value with a probability > 0 |
| "Median" | The y-value for which half the samples lie above, the other half below in the histogram. |
| "Mean" | The weighted arithmetic average of the histogram |

Remote command:
`MEASurement<m>:HISTogram:PROBability:TYPE` on page 1484

**Probability domain marker, Delta**
Defines a range around the marker.

Remote command:
`MEASurement<m>:HISTogram:PROBability:LIMit` on page 1485

## 8.2.9 Jitter measurements

Jitter measurements are availabe if one of the following options is installed:

- R&S RTP-K12
- R&S RTP-K133
- R&S RTP-K134

See Chapter 18.1, "Jitter measurements (option R&S RTP-K12)", on page 1024.

## 8.2.10 Optical measurements

This dialog provides settings for testing optical signals. Optical measurements require an optical power signal provided by an optical probe.

For accurate measurements, a dark calibration of the optical probe is required.

### 8.2.10.1 Overview of optical measurements

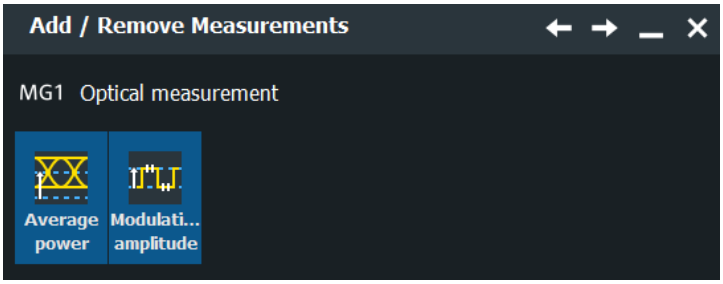
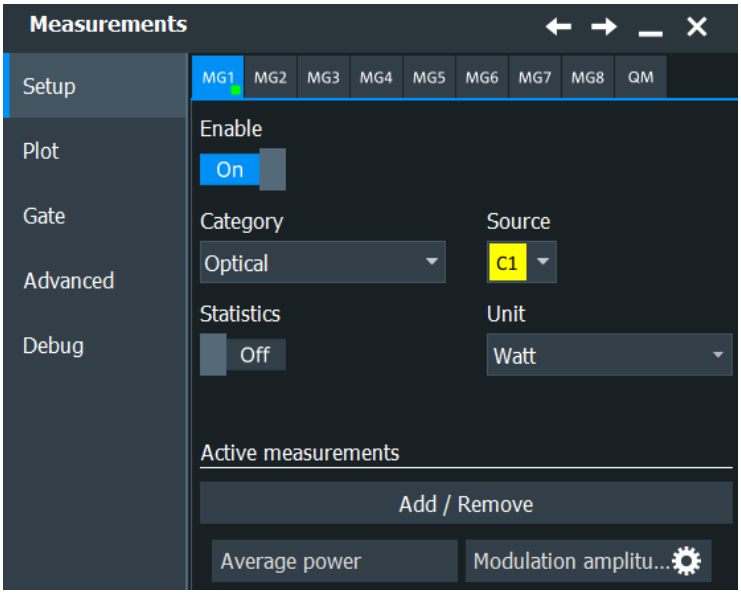



*Table 8-9: Optical measurements*

| | Meas. type | Description/Result |
|---|---|---|
| | Average power | Measure for the true average component of an optical signal.<br>$X_{Average} = \frac{1}{N_{Eval}} * \sum_{i=1}^{N_{Eval}} x(i)$ |
| | Optical modulation amplitude (OMA) | Measure of the difference between the optical power of a non-return-to-zero (NRZ) one pulse and the optical power of an NRZ zero pulse.<br>OMA requires an NRZ pattern.<br>The one pulse and zero pulse levels are measured in the level measurement window defined with Start and Stop.<br>OMA is based on norm IEC 61280-2-2. |

### 8.2.10.2 Settings for optical measurements

Access: [Measure] > "Meas Group" tab > "Optical" category



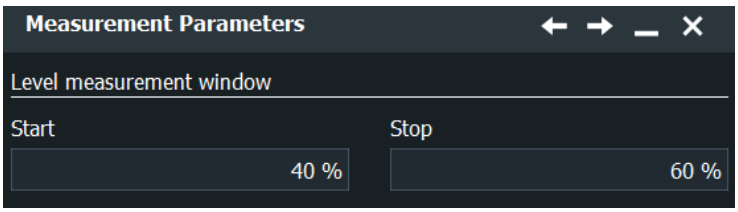



*Figure 8-14: Modulation amplitude settings*

**Unit**
Selects the unit of the optical measurement.

Remote command:
MEASurement<m>:OPTical:UNIT on page 1474

**Start**
Sets the start value of the range for the optical modulation amplitude.

Remote command:
MEASurement<m>:OPTical:OMAMplitude:RELative:STARt on page 1474

**Stop**
Sets the end value of the range for the optical modulation amplitude.

Remote command:
MEASurement<m>:OPTical:OMAMplitude:RELative:STOP on page 1474

## 8.2.11 Protocol measurements (option R&S RTP-K35)

Option R&S RTP-K35 provides specific measurements on automotive and Ethernet protocols.

### 8.2.11.1 Overview of protocol measurements

If option R&S RTP-K35 is installed, enhanced measurements on the supported serial buses are possible. With it, you can measure various dependencies between the frames.

Additional to option R&S RTP-K35 you need one of the following serial protocol options:

- I2C (R&S RTP-K1)
- SPI (R&S RTP-K1)
- UART/RS232 (R&S RTP-K2)
- CAN / CAN-FD (R&S RTP-K3/R&S RTP-K9)
- LIN (R&S RTP-K3)
- RFFE (R&S RTP-K40)
- Ethernet (100BASE-Tx) (R&S RTP-K8)
- 100BASE-T1 (R&S RTP-57)

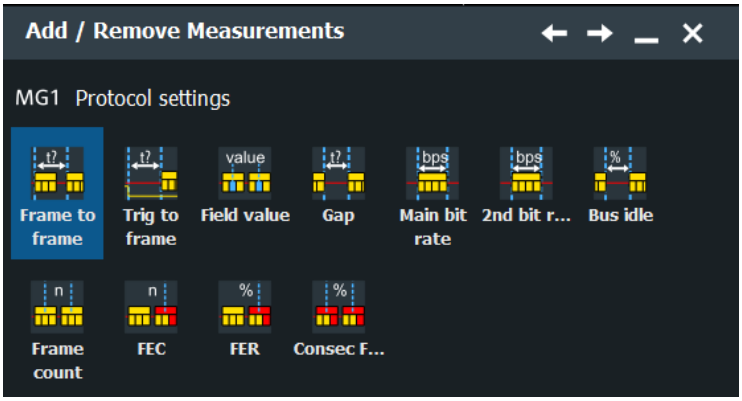


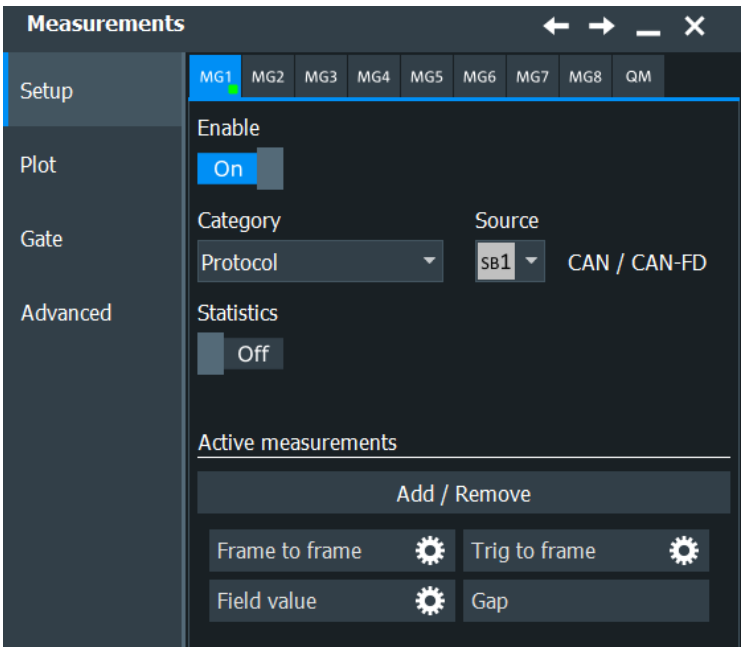

*Table 8-10: Protocol measurements*

| Meas. type | Description / result |
|---|---|
| Field value | Value of a field over time |
| Frame to frame | The distance between two frames |
| Trigger to frame | The distance between a defined frame and the next trigger signal |
| Gap | Measures a gap, period at which the bus is idle. The distance of a gap can only be measured between two identified frames. |
| Main bit rate | Bit rate as defined by the standard |
| 2nd bit rate | Additional bit rate, protocol dependent<br>For example: "Data rate" for the CAN-FD protocol. |
| Bus idle | Calculates the bus idle time |
| Frame count | Number of all frames within the acquisition window |
| Frame error count | Sum of all frames with errors within the acquisition window |
| Frame error rate | Sum of all frames with errors divided by all frames within the acquisition window |
| Consecutive frame error rate | Measures the rate at which at least two consecutive frames have an error |

### 8.2.11.2 Settings for protocol measurements

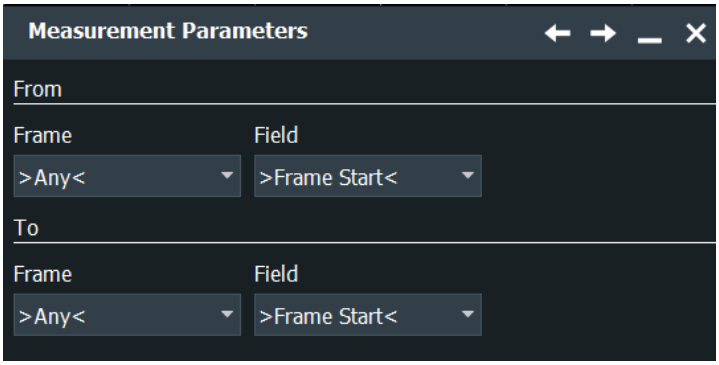
Access: [Measure] > "Meas Group" tab > "Protocol" category

Protocol measurements require that a supported protocol is enabled first.




If further settings are available for a measurement, a settings icon is shown beside the measurement's name in the "Active Measurements" list. Tap the icon to configure the measurement.

**Frame to frame**
Opens a dialog to set up the parameters for the frame to frame measurement. You can define a start frame ("From") and an end frame ("To") for the measurement.




The types of available frames and fields are depending on the enabled protocol.

***Table 8-11: Available protocol frames***

| Protocol | Frame | Field |
|---|---|---|
| I2C | W<br>R | Address |
| | | 2. Addr |
| | | ACK |
| | | R/W |
| | | Data |

| Protocol | Frame | Field |
|---|---|---|
| SPI | MISO<br>MOSI | Word |
| CAN/CAN FD | Remote<br>Data | ID<br>FDF<br>BRS<br>ESI<br>DLC<br>SCV<br>P<br>SC<br>Data<br>CRC<br>ACK |
| | Overload | - |
| LIN | WAKE | - |
| | Data | ID<br>Data |
| RFFE | Register 0 Write<br>Register Write<br>Register Read<br>Extended Register Write<br>Extended Register Read<br>Extended Register Write Long<br>Extended Register Read Long<br>Interrupt Summary and Identification<br>Masked Write<br>Master Ownership Handover<br>Master Write<br>Master Read<br>Master Context Transfer Write<br>Master Context Transfer Read | SA<br>MID<br>BC<br>Addr<br>Mask<br>Data |
| Ethernet | MAC<br>Sleep<br>EOS | Preamble<br>SFD<br>Dest<br>Src<br>Len/Type<br>Data<br>FCS |
| | Idle | Triplet |

| Protocol | Frame | Field |
|---|---|---|
| 100BASE-T1 | MAC<br>Fill | Preamble<br>SFD<br>Dest<br>Src<br>Len/Type<br>Data<br>FCS |
| | Idle | Triplet |

Also, if a label list is loaded and enabled for the protocol, the symbolic label names can be selected from the "Frame"/"Field" value list.

"Frame" Selects the type of start frame (from) / end frame (to).

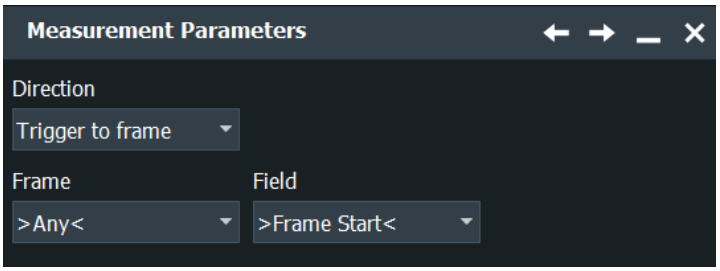
"Field" Selects the type of start field (from) / end field (to).

"Value" Sets a value for the selected field.

Remote command:
`MEASurement<m>:PROTocol:F2FRame:FRMFrom` on page 1497
`MEASurement<m>:PROTocol:F2FRame:FRMTo` on page 1497
`MEASurement<m>:PROTocol:F2FRame:FLDFrom` on page 1496
`MEASurement<m>:PROTocol:F2FRame:FLDTo` on page 1496
`MEASurement<m>:PROTocol:F2FRame:VALFrom` on page 1497
`MEASurement<m>:PROTocol:F2FRame:VALTo` on page 1497

**Trigger to frame**
Opens a dialog to set up the parameters for the trigger to frame measurement.

The types of available frames are depending on the enabled protocol, see Table 8-11.




"Direction" Selects the direction for the measurement, from the trigger to the next frame (start),

"Frame" Selects the type of frame.

"Field" Selects the type of field.

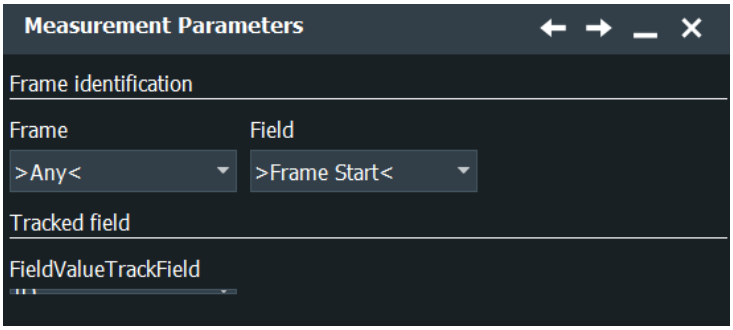
"Value" Sets a value for the selected field.

Remote command:
`MEASurement<m>:PROTocol:T2FRame:DIRection` on page 1500
`MEASurement<m>:PROTocol:T2FRame:FLD` on page 1501
`MEASurement<m>:PROTocol:T2FRame:FRM` on page 1501
`MEASurement<m>:PROTocol:T2FRame:VALue` on page 1501

**Field value**
Opens a dialog to set up the parameters for the field value measurement. You can first identify a frame with specific field value and then track a selected field of this frame.

The types of available frames and fields depend on the enabled protocol, see Table 8-11.

If a label list is loaded and enabled for the protocol, sometimes there are defined calculations and formatting for the displayed field value. In this case, the measurement displays the value according to the definition in the label list.




"Frame" Selects the type of frame.

"Field" Selects the type of field.

"Value" Sets a value for the selected field.

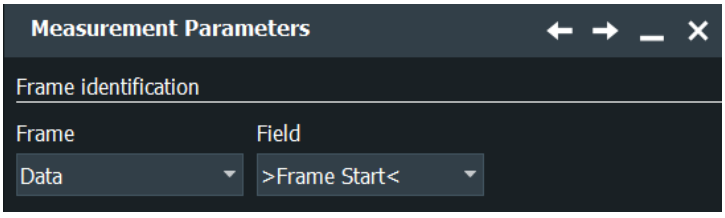
"Field to be tracked" Selects the type of field which value is tracked over time.

Remote command:
`MEASurement<m>:PROTocol:FLDValue:FLD` on page 1498
`MEASurement<m>:PROTocol:FLDValue:FRM` on page 1498
`MEASurement<m>:PROTocol:FLDValue:TRCK` on page 1498
`MEASurement<m>:PROTocol:FLDValue:VAL` on page 1499

**Main bit rate / 2nd bit rate**
Opens a dialog to set up the parameters for the main bit rate/ 2nd bit rate measurement. You can identify a frame and a specific field value.

The types of available frames and fields depend on the enabled protocol, see Table 8-11.

If a label list is loaded and enabled for the protocol, sometimes there are defined calculations and formatting for the displayed field value. In this case, the measurement displays the value according to the definition in the label list.




"Frame" Selects the type of frame.

"Field" Selects the type of field.

"Value" Sets a value for the selected field.

Remote command:
MEASurement<m>:PROTocol:MBITrate:FLD on page 1499
MEASurement<m>:PROTocol:MBITrate:FRM on page 1499
MEASurement<m>:PROTocol:MBITrate:VAL on page 1499
MEASurement<m>:PROTocol:SBITrate:FLD on page 1500
MEASurement<m>:PROTocol:SBITrate:FRM on page 1500
MEASurement<m>:PROTocol:SBITrate:VAL on page 1500

## 8.2.12 Result analysis

The behavior of measurement results over time can be evaluated in different ways:

- Statistics
- Long-term measurements
- Histograms on measurement results
- Track

### 8.2.12.1 Statistics

Statistics can be compiled for each measurement group separately, and also for long-term measurements. If enabled, statistical results are shown in the result box.

By default, all events in each acquisition are measured and incuded in the statitics: the measurement result is not only determined once within one acquisition, but repeatedly, if available. More results provide a larger basis for statistical evaluation.

To obtain precise results, additional measurement settings can be useful:

- Reference/signal levels: configuring user-defined levels can compensate for irregular data, see Chapter 8.2.4, "Reference levels", on page 332.
- Gate areas: restricting the waveform range for measurement can eliminate irregular data, see Chapter 8.2.3, "Measurement gates", on page 329.
- Defining a "Signal threshold" for time, area and counting measurements can eliminate noise from the evaluation, see "Signal threshold" on page 347.
- Spectrum measurements: you can eliminate noise from the evaluation, see Threshold and "Peak excursion" on page 319

► To enable statistics, use one of the following ways.
Make sure to select the subtab of the measurement group for which you want to compile statistics.
  - On the toolbar assist, enable "Statistics".
  - On the "Meas Group" tab, enable "Statistics".

If statistics are enabled, the following results are calculated. You can select which results are shown in the result table, see Chapter 8.2.2.3, "Display settings for results", on page 327.

| Label | Description |
|---|---|
| +Peak | Positive peak value (maximum) |
| -Peak | Negative peak value (minimum) |
| µ (Avg) | Average |
| RMS | Root mean square |
| σ (S-dev) | Standard deviation |
| Event count | Number of measured events (e.g. rising edges, pulses etc.) |
| Wave count | Number of waveforms (acquisitions) the measurement is based on |

Remote commands:

- `MEASurement<m>:RESult[:ACTual]?` on page 1434
- `MEASurement<m>:RESult:AVG?` on page 1434
- `MEASurement<m>:RESult:COUNt?` on page 1436
- `MEASurement<m>:RESult:EVTCount?` on page 1434
- `MEASurement<m>:RESult:NPEak?` on page 1434
- `MEASurement<m>:RESult:PPEak?` on page 1434
- `MEASurement<m>:RESult:RMS?` on page 1434
- `MEASurement<m>:RESult:STDDev?` on page 1434
- `MEASurement<m>:RESult:WFMCount?` on page 1434
- `MEASurement<m>:RESult:STARt?` on page 1436
- `MEASurement<m>:RESult:STOP?` on page 1436
- `MEASurement<m>:ARNames` on page 1434
- `MEASurement<m>:ARES?` on page 1433

The peak and average values and the standard deviation of the long-term points are also shown in the graph of the long-term measurement.

Stopping and starting the acquisition does not reset statistics but only stops and continues them.

The instrument only resets statistical evaluation if you change measurement setup:

- Select measurement
- Create or modify gate
- Enable/disable long-term measurement and histogram
- Enable continuous autoscale with enabled histogram
- Switch on/off channels
- Enable/disable cursors
- Tap "Reset" or "Clear sreen results"

After a reset, new statistics are compiled beginning with the next acquired waveform.

If limit and margin check is enabled, the icons in the result table indicate if statistical results exceed a limit or margin. These violations do not initiate an action.

#### 8.2.12.2 Long-term measurements

Long-term measurements show the behavior of measurement results over a longer time or for many samples. Therefore the measurement results of a specified time period are summarized into one long-term point. For each point, the current value measured at the end of the time period is written to the long-term waveform. In addition, statistical results for each time period are calculated, saved, and reset. This reset avoids constantly rising maximum or constantly falling minimum values until the end of the measurement.

You can define the number of long-term points and export the long-term data, including statistical results. The measurement histogram is a vertical histogram shown in the long-term diagram.

Long-term measurements are done on the selected "Measurement for analysis / math".

See also: "Enable (Long term)" on page 388.

**Performing long-term measurements**

1. On the "Meas" menu, select "Result Analysis".
2. Select the subtab for the measurement group you want to configure.
3. Under "Long term", tap "Enable".
4. Since the waveform can change in the process of time, enable "Continuous auto scale" to adapt the scaling automatically.
   Alternatively, tap the "Auto scale" to adjust the scale once and to see the long-term waveform.
5. Tap "Horizontal scaling".
6. Define the "Number of points" to be shown in the long-term diagram.
7. Set the "Scale mode" that defines the period of time from which one long-term point is created.
   See "Scale mode" on page 391 for setting details.

   For each long-term measurement point, the current measurement value is added to the long-term waveform.
8. If you need the statistical data of the long-term points:
   a) Tap "Result Analysis" to return to the measurement settings.
   b) Enable statistics.
   c) Let the measurement run and export the data when finished.

#### 8.2.12.3 Histograms on measurement results

Histograms are available not only for channel, math and reference waveforms, but also on measurement results. These histograms cannot be configured, and they are shown in a separate diagram. The source is a measurement, and all events in all acquisitions are are measured and included in the histogram. If the histogram is based on long-term measurement, it is shown in the long-term diagram.

The histogram is built from the results of the selected "Measurement for analysis / math".

See also: "Enable (Histogram)" on page 389.

#### 8.2.12.4 Track

The track is a waveform that shows measurement values in time-correlation to the measured signal. It is the graphical interpretation of all measurement values of a single acquisition.

The track is available for most amplitude/time measurements (except for High, Low, Amplitude, Max, Min, Peak to peak, Mean, RMS, S-dev, Pos. and Neg. overshoot, and Area), and for jitter measurements.

Enabling the track enables also the Continuous auto scale and Measure all events in each acquisition.

To analyze the track, you can use cursor measurements and zoom.

**Creating a track waveform**

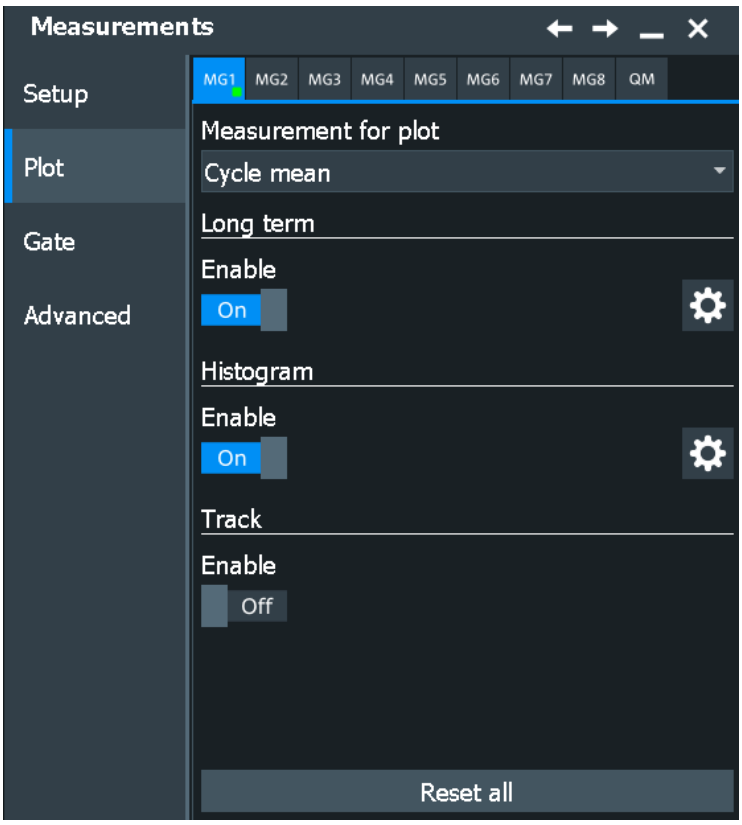
1. Configure the measurement group.
2. Activate the measurement group.
3. Select the "Result Analysis" tab.
4. If several measurements are enabled, select the "Measurement for analysis / math".
5. Enable the track.

See also: "Enable (Track)" on page 390.

#### 8.2.12.5 Settings for result analysis

The settings in the "Result Analysis" tab activate and configure long-term measurements, statistical calculations, and the histogram of measurement results over a longer period of time.

For scaling settings of the long-term diagram, see Chapter 8.2.12.6, "Horizontal long-term scaling", on page 391.



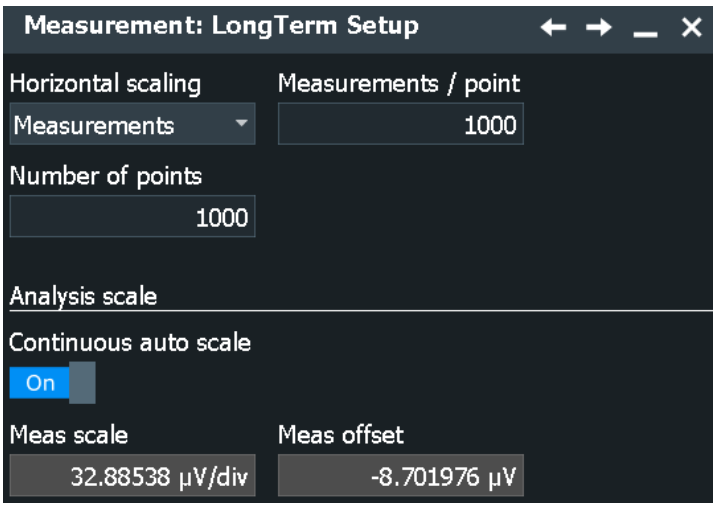

**Measurement for plot**
Selects the measurement that is used as source for mathematic calculations, long-term measurements, and histograms. These evaluations are performed on one measurement only, not on all measurements of the group. Statistics are calculated for all measurements.

The selected measurement cannot be disabled in the "Add / Remove Measurement" dialog box.

Remote command:
`MEASurement<m>:MAIN` on page 1430

**Enable (Long term)**
Enables long-term measurement of the main measurement.

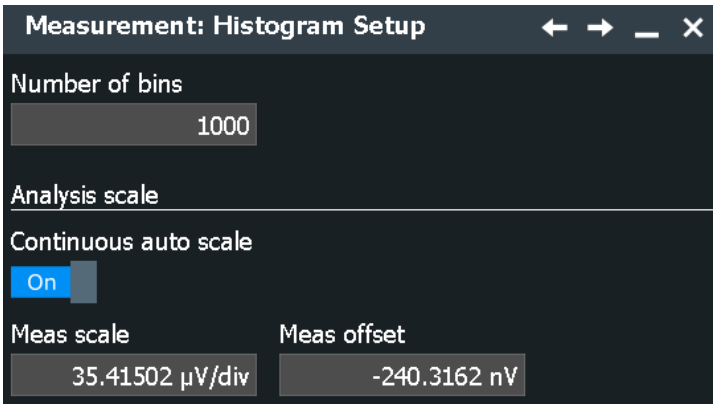
Long-term measurements are performed on the "Measurement for plot".




Remote command:
MEASurement<m>:LTMeas[:STATe] on page 1494

**Enable (Histogram)**
Displays a histogram of measurement results - the cumulative occurrence distribution of measurement results in a graphic. Enabling the histogram enables also the statistics and "Measure all events in each acquisition".

Long-term measurements are performed on the "Measurement for plot".




Remote command:
MEASurement<m>:STATistics:HISTogram on page 1490

**Number of bins**
Sets the number of bins - the number of vertical bars that build the histogram.

If "Continuous auto scale" is enabled, the instrument determines the number of bins based on the timebase, the current measurements, and other settings. To set the number of bins manually, disable "Continuous auto scale".

Remote command:
MEASurement<m>:STATistics:HBINs on page 1490

**Enable (Track)**
Enables the track of measurement results over time and displays the track waveform. It is the graphical interpretation of all measurement values of a single acquisition.

The track is available for most amplitude/time measurements (except for High, Low, Amplitude, Max, Min, Peak to peak, Mean, RMS, S-dev, Pos. and Neg. overshoot, and Area), and for jitter measurements.

Enabling the track enables also the Continuous auto scale and Measure all events in each acquisition.

Before you can enable the track, activate the appropriate measurement.

If option R&S RTP-K12 basic jitter analysis is installed, you can use tracks to display the jitter measurement results as a time-correlated waveform, see Chapter 18.1.4, "Track of jitter measurement results", on page 1032.

Remote command:
`MEASurement<m>:TRACk[:STATe]` on page 1502

**Reset**
Immediately resets the histogram, the long-term measurement and the statistics.

Stopping and starting the acquisition does not reset these analyses but only stops and continues them.

To delete all results, waveforms and history, select "Display" menu > "Clear all".

Remote command:
`MEASurement<m>:STATistics:RESet` on page 1492

**Analysis scale**
The measurement scale of a long-term measurement diagram or measurement histogram can be set automatically by the instrument, or manually.

Use automatic scaling if the measurement is running and you cannot see the expected results.

"Continuous auto scale"
: Performs an automatic scaling whenever the long-term waveform or the histogram does not fit in the diagram during the measurement period.

"Vertical"
: Defines the vertical scaling per division for long-term measurement period and the measurement histogram.

"Offset"
: Defines an offset for the long-term measurement and the measurement histogram.

Remote command:
`MEASurement<m>:VERTical:CONT` on page 1493
`MEASurement<m>:VERTical:AUTO` on page 1493
`MEASurement<m>:VERTical:SCALe` on page 1494
`MEASurement<m>:VERTical:OFFSet` on page 1493

**Measure all events in each acquisition**
Normally, only one measurement is performed for each acquired waveform to get best performance. If "Measure all events in each acquisition" is enabled, more than one result is taken from one acquired waveform and the results are included evaluation. For example, the rise time is measured on all pulses in the waveform, not only on the first.

The result box shows only the first result of the waveform, the following results are used only for evaluation.

All event results are also considered in limit and margin checks and can initiate an action. However, the icons in the result box only indicate violations of the first result.

Measuring all events is enabled automatically when calculating statistics and histograms. It can also be useful when generating tracks; however, it reduces the performance of the instrument.

The number of considered results can be restricted: "Limit" sets the maximum number of measurement results per acquisition.

Remote command:
`MEASurement<m>:MULTiple` on page 1489
`MEASurement<m>:MNOMeas` on page 1490

### 8.2.12.6 Horizontal long-term scaling

In this dialog box, you define the horizontal scale of long-term measurement diagrams. The length of the long-term measurement is defined by the number of points.

**Number of points**
Defines the total number of points to be displayed in the long-term measurement diagram.

Remote command:
`MEASurement<m>:LTMeas:COUNt` on page 1494

**Scale mode**
Defines when the points of a long-term measurement are created.

If statistics are enabled, each long-term measurement point shows the statistical mean and standard deviation of the results measured during the defined period.

If statistics are disabled, the first measurement result of each period is taken as long-term measurement point.

| | |
|---|---|
| "Time" | Sets one long-term measurement point for the time defined in "Time/point". |
| "Waveforms" | Sets one long-term measurement point for several acquired waveforms. The number is defined in "Waveforms/point". |
| "Measurements" | Sets one long-term measurement point for several measurement results. The number is defined in "Measurements/point". |

Remote command:
`MEASurement<m>:STATistics:MODE` on page 1491

**Time / point**
Defines the time to create one point of the long-term measurement. The "Time / point" value is a lower time limit. The actual time between two points depends on the acquisition and postprocessing time.

The long-term measurement is not a data logger with equidistant points as the time between two points varies.

This setting is only available if "Scale mode" is set to "Time".

Remote command:
`MEASurement<m>:STATistics:RTIMe` on page 1492

**Measurement time**
Defines the total duration of the long-term measurement: *Time/point * Number of points*.

This setting is only available if "Scale mode" is set to "Time".

Remote command:
`MEASurement<m>:LTMeas:TIME` on page 1495

**Wfms / point**
Defines the number of measured waveforms from which one point of the long-term measurement is created.

This setting is only available if "Scale mode" is set to "Waveforms".

Remote command:
`MEASurement<m>:STATistics:RCOunt` on page 1491

**Measurements / point**
Defines the number of measurement results from which one point of the long-term measurement is created.

This setting is only available if "Scale mode" is set to "Measurements".

Remote command:
`MEASurement<m>:STATistics:RMEascount` on page 1492

## 8.2.13 Limit and margin checks

Limit and margin checks evaluate whether the measurement result exceeds a specified value. Violations are indicated by icons in the result box. Furthermore, you can define actions that are performed on limit or margin violation, like saving the waveform or measurement results.

The following results are considered in limit and margin checks:

- All selected measurements.
- All measured events in an acquisition: all results can initiate an action. However, the icons in the result box indicate only violations of the first result.
- Statistical results. Limit and margin violations of statistical results are indicated by icons in the result box. These violations do not initiate an action.

See also: Chapter 8.2.2.1, "Measurement status", on page 326

### 8.2.13.1 Performing limit checks

1. Open the menu and select "Measure".
2. Select the "Advanced" tab.
3. Select the subtab of the measurement group that you want to configure.
4. Under "Type", select "Limit only" to distinguish only between valid and non-valid values.
   Select "Margin&Limit" to check two values, where the margin is still valid, while the limit is not.
5. Tap "Setup". Define the valid value range for each active measurement.
   The margins must always be within the valid value range. If necessary, the limit or margin values are adapted to match the selected valid range.
   See also "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394
6. Define what happens when the defined limits and margins are exceeded.

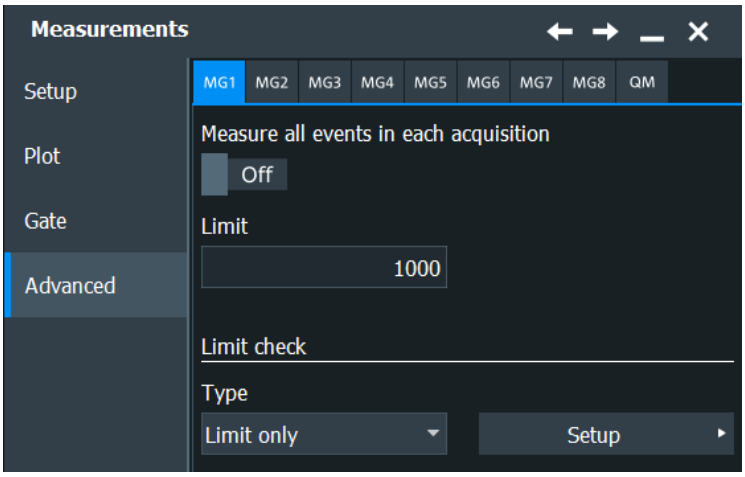
   For each action, define when the instrument starts it:
   - If the limits or margins are exceeded
   - If the measurement is completed without limit violations
   - Not at all

   As a result of the limit check, the specified actions are performed and an icon indicates the status in the result box.

### 8.2.13.2 Limit and margin settings

Access: [Measure] > "Advanced" tab

If a check is selected, tap "Setup" to set the limit and margin values and the range of valid measurement results.



**Type**
Selects and enables the limit or margin check.

| | |
|---|---|
| "Off" | No limit check is performed. |
| "Limit only" | Limits are checked for violation. |
| "Margin & Limit" | Margins and limits are checked for violation. |

**Upper limit, Lower limit, Upper margin, Lower margin, Valid range**
Set the limits and margins for each measurement, and also specify the valid range.

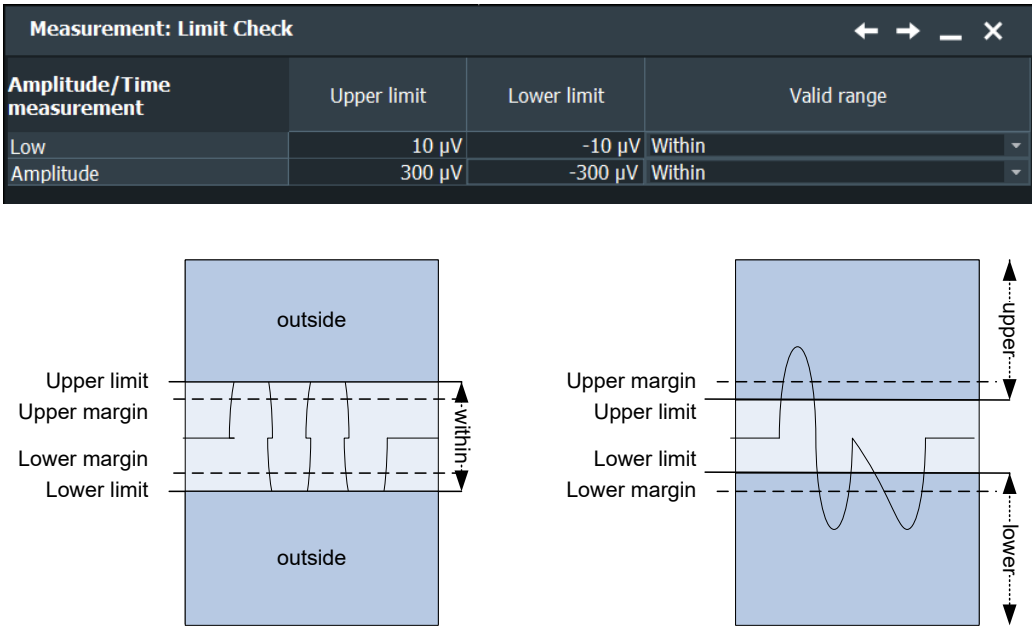



*Figure 8-15: Limit and margin definition*

Limits are stricter than the margins for the value check. Thus, the margins must be within the valid range. If necessary, the limit and margin values are adapted according to the selected valid range.

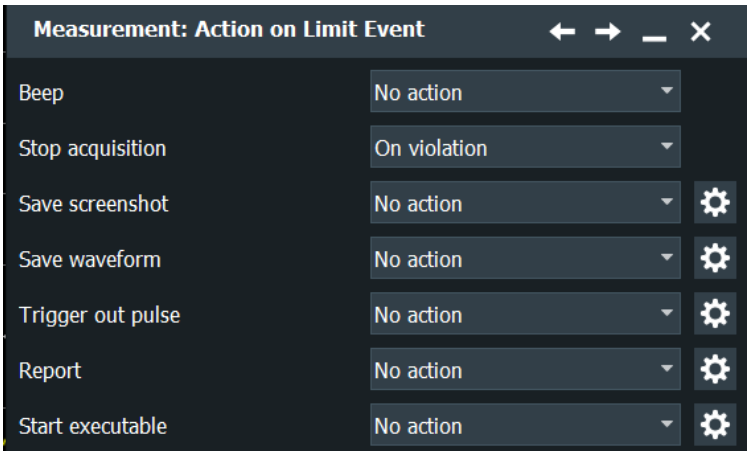
The settings are only visible if "Limit check" is enabled.

Remote command:
`MEASurement<m>:AMPTime:LCHeck<n>:LOWer:LIMit` on page 1446
`MEASurement<m>:AMPTime:LCHeck<n>:LOWer:MARGin` on page 1447
`MEASurement<m>:AMPTime:LCHeck<n>:UPPer:LIMit` on page 1446
`MEASurement<m>:AMPTime:LCHeck<n>:UPPer:MARGin` on page 1447
`MEASurement<m>:AMPTime:LCHeck<n>:VALid` on page 1446
To check limits and margins of jitter measurements, use the AMPTime remote commands.
`MEASurement<m>:EYEJitter:LCHeck<n>:LOWer:LIMit` on page 1459
`MEASurement<m>:EYEJitter:LCHeck<n>:LOWer:MARGin` on page 1459
`MEASurement<m>:EYEJitter:LCHeck<n>:UPPer:LIMit` on page 1459
`MEASurement<m>:EYEJitter:LCHeck<n>:UPPer:MARGin` on page 1459
`MEASurement<m>:EYEJitter:LCHeck<n>:VALid` on page 1458

MEASurement<m>:HISTogram:LCHeck<n>:LOWer:LIMit on page 1486
MEASurement<m>:HISTogram:LCHeck<n>:LOWer:MARGin on page 1486
MEASurement<m>:HISTogram:LCHeck<n>:UPPer:LIMit on page 1486
MEASurement<m>:HISTogram:LCHeck<n>:UPPer:MARGin on page 1486
MEASurement<m>:HISTogram:LCHeck<n>:VALid on page 1485
MEASurement<m>:SPECtrum:LCHeck<n>:LOWer:LIMit on page 1472
MEASurement<m>:SPECtrum:LCHeck<n>:LOWer:MARGin on page 1473
MEASurement<m>:SPECtrum:LCHeck<n>:UPPer:LIMit on page 1472
MEASurement<m>:SPECtrum:LCHeck<n>:UPPer:MARGin on page 1473
MEASurement<m>:SPECtrum:LCHeck<n>:VALid on page 1471

### 8.2.13.3 Actions on limit check results

To define what happens when the limits and margins are exceeded, tap Action on limit event below the limit and margin ranges.




Actions are initiated by all measurements and all results measured on one acquisition.

Note that the violation actions do not distinguish between a margin violation and a limit violation. However, different icons are displayed in the result box.

For each action, you can define the event on which the action is initiated:

- On violation
  The action is initiated when the limits or margins are exceeded during the measurement.
- On successful completion
  The action is initiated when a defined number of acquisitions has been captured, and the limits or margins were not exceeded.

Independent of these actions, an icon is displayed in the result box, see Chapter 8.2.2, "Measurement results", on page 325.

**Beep**
Generates a beep sound.

Remote command:
MEASurement<m>:ONViolation:BEEP on page 1507

**Stop acq**
Stops data acquisition on violation.

Remote command:
MEASurement<m>:ONViolation:ACQStop on page 1507

**Save Wfm**
Saves the waveform data to the file.

Remote command:
MEASurement<m>:ONViolation:WFMSave on page 1508

**Trigger Out Pulse**
Creates a pulse on the Trigger Out connector on limit violation.

When "Trigger Out Pulse" is used, the trigger control option "Enable trigger out" is disabled. Thus, the trigger-out pulse is created only on limit violation but not when a trigger occurs. The pulse is provided always with the minimum delay of 800 ns, the "Delay" cannot be set.

See also: "Trigger out signal setup" on page 230.

Remote command:
MEASurement<m>:ONViolation:TRIGgerout on page 1508

**Report**
Creates and saves a report using the settings defined in "File" menu > "Report Setup".

Remote command:
MEASurement<m>:ONViolation:REPort on page 1508

**Start Executable**
Starts an external application. Tap "Config Executable" to set the application path and parameters.

Remote command:
MEASurement<m>:ONViolation:RUNexec on page 1509

## 8.3 Quick measurements

Quick measurement performs a set of up to eight amplitude/time measurements on one source, simply by tapping the "Quick measurement" toolbar icon. The results are displayed in a results box. You can configure the measurement to be included in quick measurement. The current configuration can be saved to repeat the measurement quickly.

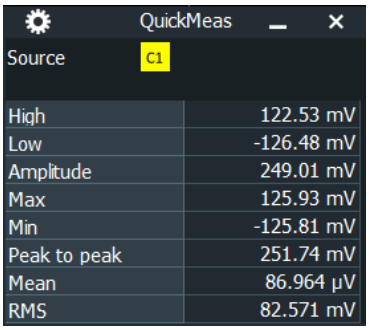
### 8.3.1 Starting the quick measurement

If the "Quick meas" icon is not visible on the toolbar, add it to the toolbar: see Chapter 3.3.6.2, "Configuring the toolbar", on page 56.

1. Tap the waveform that you want to measure.
2. Tap the "Quick measurement" icon on the toolbar.

3. Tap the diagram.

   The result box shows the results of the default quick measurement.




### 8.3.2 Configuring the quick measurement

The default configuration of the quick measurement includes already 8 amplitude measurements. If these measurements do not fit the measurement task, you can modify the selection.

1. Open "Menu" > "Measure" > "Meas Group".
2. Select the "QM" subtab.

3. Tap "Add/Remove Measurements".
4. Disable all measurements that you do not need.
5. Select the measurements that you want to add to the quick measurement.
6. Tap "OK".
7. A "Settings" icon indicates whether further settings are required. Tap the measurement. A dialog box opens, where you can adjust the measurement.
8. Tap "Set as QuickMeas" to save the setup.

   The saved iQuickMeas setup is used when you start a new quick measurement.
9. "Enable" the measurement.

**Set as QuickMeas**
Saves the current QuickMeas setup. The saved QuickMeas configuration is used when you start a new quick measurement. It remains until you save another QuickMeas setup, or until you reset the instrument to "Factory defaults". The "Set as QuickMeas" is not available if the current configuration already has been saved.

# 9 Spectrum analysis

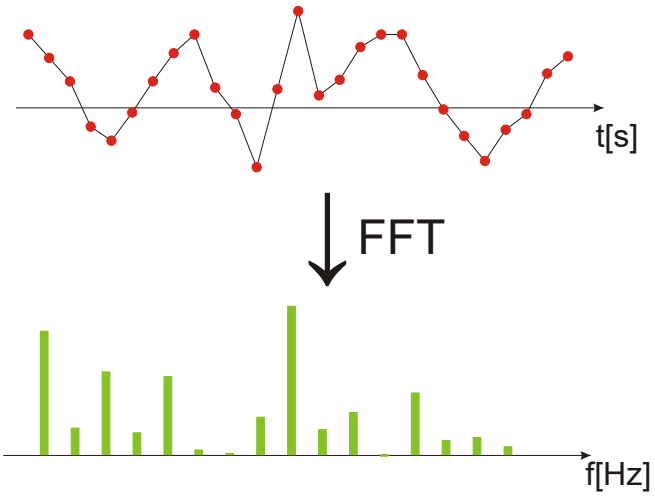
The R&S RTP provides two ways of spectrum analysis:

- Basic FFT calculation, which is included in the firmware
- Spectrum analysis option R&S RTP-K37, which provides a wide range of analysis possibilities, for example, spectrogram, cursor and automatic measurements.

## 9.1 FFT analysis

### 9.1.1 Fundamentals of FFT analysis

During FFT analysis, a signal in the time domain is converted to a spectrum of frequencies. As a result, either the magnitude or the phase of the determined frequencies can be displayed. FFT analysis can be restricted to an extract of the original time base, and the results display can be restricted to a specified frequency range.

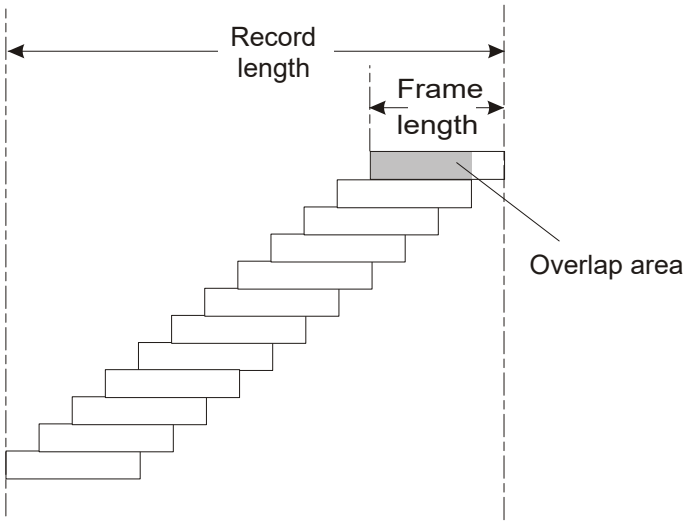



**Frames/Segments**

To convert the time domain signal to a frequency spectrum, an FFT (Fast Fourier Transformation) unit is used which converts a vector of input values into a discrete spectrum of frequencies.

Conventional oscilloscopes calculate one FFT per capture. The R&S RTP can calculate multiple FFTs per capture by dividing one capture into several *segments*, or *frames*. Thus, the R&S RTP can visualize how the frequency content of a signal changes over time which helps to detect intermittent or sporadic signal details. Furthermore, the R&S RTP allows consecutive frames to overlap. This is especially useful in conjunction with window functions since it enables a gap-free frequency analysis of the signal.

The overlapping factor can be set freely. The higher the overlap factor, the more frames are used. This leads to more individual results and improves detection of transient signal effects. However, it also extends the duration of the calculation. The size of the frame depends on the number of input signal values (record length), the overlap factor, and the FFT size (number of samples used for FFT calculation).




### Window functions

Each frame is multiplied with a specific window function after sampling in the time domain. Windowing helps minimize the discontinuities at the end of the measured signal interval and thus reduces the effect of spectral leakage, increasing the frequency resolution.

There are several window functions that can be used in FFT analysis. Each of the window functions has specific characteristics, including some advantages and some trade-offs. Consider these characteristics carefully to find the optimum solution for the measurement task.

For details, see "Window type" on page 410.

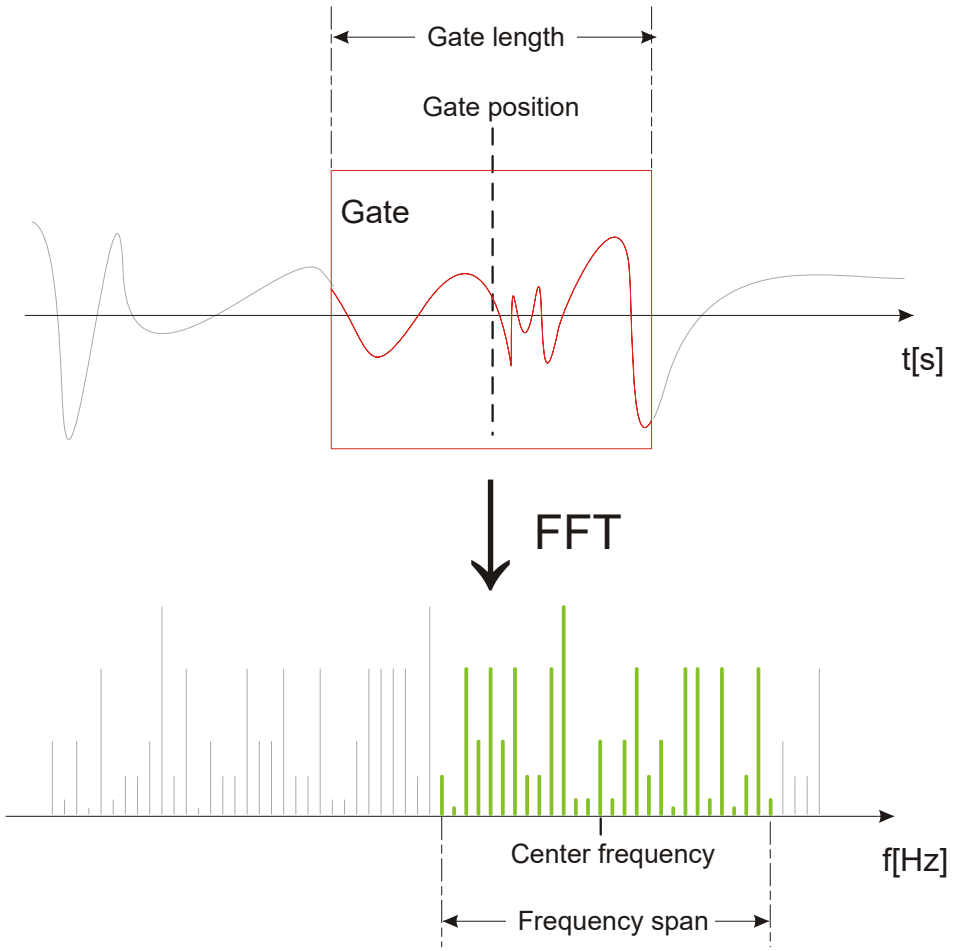
### Combining FFT results

After the window function has been applied to the FFT results for each segment, the results for all segments of the data acquisition must be combined to obtain the final waveform. Various arithmetic functions are available for FFT segments, such as averaging, enveloping, or minimum and maximum calculation.

### Gating functions

You can restrict the time base of the input signal for which FFT analysis is to be performed. There are various methods to do so:

- Define absolute start and stop times for the time base extract
- Define relative start and stop values that define a percentage of the original time base
- Couple the time base extract for FFT to an active zoom area.




### Restricting the result range

You can restrict the results of the FFT analysis to a specified frequency range. The frequency range can be defined in two ways:

- Define a center frequency and frequency span
- Define start and stop frequencies

Using the new cursor functions for spectrum waveforms you can easily determine the results for the current center frequency by moving the cursor to that frequency ("C1 to Center"). If you detect a point of interest in the spectrum diagram, you can place the cursor on it and then move the center frequency to the position of the cursor automatically ("Center to C1").

See Chapter 8.1, "Cursor measurements", on page 308.

### Magnitude vs. phase display

The result of an FFT analysis is a spectrum of frequencies. Either the magnitudes or the phases of those frequencies are displayed, depending on the used FFT function. In "Basic" mode, and for the "Advanced" mode FFT functions |FFT|, FFT (re) and FFT

(im), the magnitude is displayed. For the "Advanced" mode FFT (φ) function, the phase is displayed.

For magnitude display, you can select the scale and range of magnitudes to be displayed. For linear scaling, the vertical value range of the input signal is used. For logarithmic scaling, the logarithmic power of the frequency is displayed. In this case, the input signal must be given in either Volt or Watt. The resulting value range is defined by a maximum value and a range size. Logarithmic scaling can also be set in relation to a given reference value.

For phase display, you can select the unit and suppress phases beneath a threshold value which are most likely caused by noise. The value range [-π, + π] or [-180°, +180°] is used. Phase shifts due to a limitation of the value range can be eliminated using the "Unwrap" function.

#### Dependencies between FFT parameters

FFT analysis in the R&S RTP is highly configurable. Several parameters, including the resolution bandwidth, frequency span and center frequency, can be defined according to your requirements. Note, however, that several parameters are correlated and not all can be configured independently of the others.

The **resolution bandwidth** defines the minimum frequency separation at which the individual components of a spectrum can be distinguished. Small values result in high precision, as the distance between two distinguishable frequencies is small. Higher values decrease the precision, but increase measurement speed.

The minimum achievable RBW depends on the integration time which is equivalent to the number of samples available for FFT calculation. If a higher spectral resolution is required, the number of samples must be increased by using a higher sample rate or longer record length. To simplify operation some parameters are coupled and automatically calculated, such as record length and RBW.

The **frequency span** and **center frequency** define the start and stop frequency of the spectral diagram. By default, a suitable frequency range according to the resolution bandwidth is selected, in respect to performance and precision. Span and RBW settings are coupled, so that the parameters can be adjusted automatically as necessary.

With a **Span/RBW ratio** of 100 and a screen resolution of 1000 pixels, each frequency in the spectrum is displayed by 10 pixels. A span/RBW ratio of 1000 provides the highest resolution. For full flexibility, the span/RBW coupling can also be disabled. Note, however, that a higher span/RBW ratio (i.e. low RBW values and large frequency spans) result in large amounts of data and extend the duration of the calculation.

#### Advanced FFT functions

In "Advanced" math definition mode, other FFT results than the basic frequency magnitude can be displayed.

- **FFT (φ)**: phase display
- **FFT (im)**: imaginary part of FFT value (magnitude)
- **FFT (re)**: real part of FFT value (magnitude)

- **FFT -dφ*df** (group delay): the negative derivative of the phase with respect to frequency; useful to measure phase distortion

## 9.1.2 Configuring spectrum waveforms

During FFT analysis, a signal in the time domain is converted to a spectrum of frequencies. A basic spectrum waveform can be displayed quickly. By defining additional FFT parameters, the waveform can be configured in more detail.

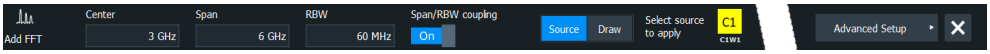
As a result, either the magnitude or the phase of the determined frequencies can be displayed, or more complex FFT functions. Analysis can be restricted to an extract of the original time base, and the results display can be restricted to a specified frequency range.

**To display a basic spectrum waveform using the toolbar**

1. Add the "FFT" icon to the toolbar, see Chapter 3.3.6.2, "Configuring the toolbar", on page 56 .

2. Tap the "FFT" icon on the toolbar.

   The FFT overlay menu opens.




3. Tap the waveform for which the FFT is performed.

   The first available math waveform is configured to use the selected waveform as a source and the "Mag(FFT(x))" operator, and is enabled. The spectrum waveform is displayed in a new diagram.

**To display a basic spectrum waveform using the [Spectrum] key**

1. Press the [Spectrum] key.

   The "FFT" tab of the "Math" dialog opens.

2. Set the "Source" to the input signal.
3. Enable "Display".
4. If necessary, edit the spectrum waveform parameters as described in the following procedures.

**To display advanced spectrum waveforms**

Using the "Formula Editor", you can display other FFT results than the basic frequency magnitude, and use more FFT sources.

1. Open the "Menu" > "Math" dialog.
2. Select the Equation tab.

3. Double-tap the editing area.

   The "Formula Editor" is displayed.

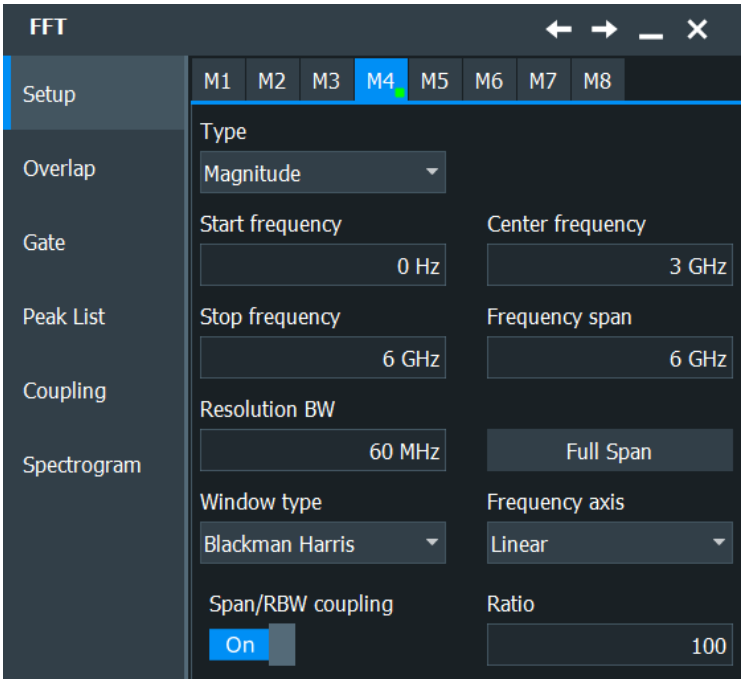
4. Delete the contents of the edit field.
5. Tap the "More" key to display further functions in the editor.
6. Tap the required FFT function key.
7. Select the source channel.
8. Close the parenthesis.
9. Tap "Enter"
10. Select the "Scale" tab to adjust the vertical and horizontal scale of the spectrum.
11. On the "Setup" tab, select "FFT setup" to configure the spectrum. See "To configure the spectrum of FFT analysis" on page 404.

### To configure the spectrum of FFT analysis

By default, a suitable frequency range for the expected horizontal values according to the resolution bandwidth is selected, in respect to performance and precision. Span and RBW settings are coupled. If a more precise evaluation is required, for example for postprocessing in a different application, disable the coupling and change the frequency ranges and resolution bandwidth values as required.

1. Open [Spectrum] > "FFT setup".

   The "FFT" dialog opens.




2. In the "Setup" tab, tap the "Frequency axis" button to select the type of scaling you want to use: linear or logarithmic.

3. Disable the "Span/RBW coupling".
4. Specify the frequency range you want to display using one of the following methods:
   - Enter a "Center frequency" and a "Frequency span" that define the spectrum.
   - Enter a "Start frequency" and "Stop frequency" that define the spectrum.
   - Tap the "Full Span" button to display the complete spectrum resulting from the FFT analysis.
5. Define the resolution bandwidth for the FFT result.
   The resolution bandwidth defines how precise the results are, i.e. how close together the individual frequencies can be. Small values result in high precision, as the distance between two distinguishable frequencies is small. Higher values decrease the precision, but increase performance.

   You can define the RBW manually, or couple it to other FFT settings. Do one of the following:
   - To couple the RBW to the span, enable the "Span/RBW coupling" option and define the "Ratio". The smaller the ratio, the higher the RBW becomes to display the same frequency span.
   - Enter the "Resolution BW" manually. The "Span/RBW coupling" option is automatically disabled.
   - To couple the RBW to the specified record length, in the " Gate" tab of the "Advanced FFT" dialog box, select the "Record length controlled" option. This option is only available if no gate is being used ("Use Gate" disabled).
6. Select the most suitable "Window type" for your source data. Window functions are multiplied with the input values and thus can improve the FFT display. For details, see "Window type" on page 410.
7. Optionally, select an arithmetic mode for the FFT segments. This mode defines how the individual segment results are combined to a final spectrum waveform.
   In the "Overlap" tab of the "Advanced FFT" dialog box, tap "Segment arithmetic" and select the required mode from the list.
8. If you use an arithmetic mode, increase the "Overlap factor" for neighboring segments to increase the accuracy of the results.

**To restrict the input values (gating)**

By default, the FFT is calculated for the entire record length as defined for the data acquisition. However, you can restrict the time range for which the FFT is calculated, resulting in a restricted spectrum. Alternatively, the record length can be determined automatically according to the selected RBW.

1. Open [Spectrum] > "FFT setup" > "Gate" tab.
2. Determine how the input length is configured by selecting one of the following options:
   - To ensure that the FFT is calculated for the full defined record length, select the "Record length controlled" option. This option is only available if no gate is

being used ("Use Gate" disabled). The RBW is adapted so that the record length can be acquired in the specified acquisition time. However, the RBW is restricted, so that data acquisition may fail if the record length is too long for the specified acquisition time.

- To couple the used record length to the required RBW, select the "RBW controlled" option. This option is only available if no gate is being used ("Use Gate" disabled).
  The required acquisition time for the defined RBW value is indicated.
- To restrict the basis of the FFT calculation to a certain time base, configure a time gate, that is: an extract of the time base in the original diagram. To do so, enable the "Use Gate" option, then do one of the following:
  – Select the "Absolute" mode and enter the "Start" and "Stop" times that define the gate area.
  – Select the "Relative" mode and enter the percentages of the total time base that define the "Relative Start" and "Relative Stop" times.
  – If a zoom area has already been defined in the original diagram and you want to use the same time base for FFT analysis, select "Zoom coupling".

The spectrum waveform displays the spectrum for the specified time span.

**To configure magnitude results**

1. Open [Spectrum] > "FFT setup" > "Setup" tab.
2. Set "Type" = "Magnitude".
3. In the "Scale" tab of the "Math" dialog, select the "Unit".
4. Select the "Vertical scale mode" to configure the value range manually or use the automatic settings.
5. For "Vertical scale mode" > "Manual", define the size of the "Vertical range" and the "Vertical maximum" to be displayed.
   For "Vertical scale mode" > "Auto", define the size of the "Vertical range" to be displayed.
   For logarithmic scaling ("Unit = dB"), also define the "Reference level" to be used.

**To configure phase results**

1. Open [Spectrum] > "FFT setup" > "Setup" tab.
2. Set "Type" = "Phase".
3. In the "Scale" tab of the "Math" dialog, select the "Unit".
4. To eliminate phase shifts due to a limitation of the value range, enable the "Unwrap" function.
5. To suppress small phase values due to noise, enable the "Suppression" function and enter a "Threshold" value.

**To couple spectrum displays**

The settings for one or more spectrums can be coupled. Thus, if any FFT setting for any of the coupled spectrums are changed, they are changed for all coupled spectrums.

1. Open [Spectrum] > "FFT setup" > "Overlap" tab.
2. Select the spectrums that you need to couple. You cannot select the spectrum for the currently selected math waveform. Its settings are applied to the selected spectrums.
3. If necessary, define an FFT function to be used for the coupled math waveforms so that a spectrum is displayed. See Chapter 7.3.1, "Displaying math waveforms", on page 279.

## 9.1.3 FFT configuration settings


- FFT setup....407
- FFT overlap....411
- FFT gating....412
- Peak list....415
- FFT coupling....415

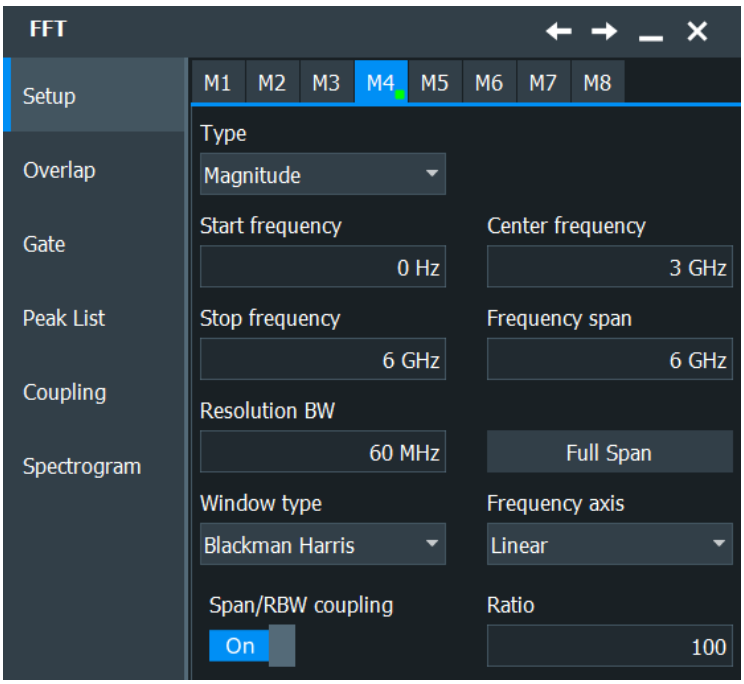

### 9.1.3.1 FFT setup

Access: [Spectrum] > "FFT setup" > "Setup" tab

In this tab, you define the settings for the FFT window. The display can be restricted to the results for a certain time base extract and to a specified frequency range.

Additional settings are available if the Spectrum analysis option (R&S RTP-K37) is installed.

See Chapter 9.2.4, "Spectrogram configuration settings", on page 421.




Type ..... 408
Display ..... 408
Frequency axis ..... 408
Center frequency ..... 409
Frequency span ..... 409
Full span ..... 409
Start frequency ..... 409
Stop frequency ..... 409
Span/RBW Coupling ..... 409
Span/RBW Ratio ..... 410
Resolution BW ..... 410
Window type ..... 410
Use color table ..... 410

**Type**
Selects between the magnitude and phase spectrum type.

Remote command:
`CALCulate:MATH<m>:FFT:TYPE` on page 1527

**Display**
If activated, a diagram for the defined math waveform is displayed on the touchscreen.

Remote command:
`CALCulate:MATH<m>:STATe` on page 1401

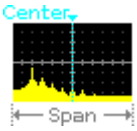
**Frequency axis**
Defines the scaling method for the frequency (x-) axis of the spectrogram.

"Logarithmic" Logarithmic scaling

"Linear Unit" Linear scaling

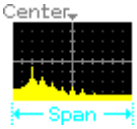
Remote command:
CALCulate:MATH<m>:FFT:LOGScale on page 1525

**Center frequency**
Defines the position of the displayed frequency range, which is (Center - Span/2) to (Center + Span/2). The width of the range is defined using the "Frequency span" setting.




Remote command:
CALCulate:MATH<m>:FFT:CFRequency on page 1526

**Frequency span**
The span is specified in Hertz and defines the width of the displayed frequency range, which is (Center - Span/2) to (Center + Span/2). The position of the span is defined using the "Center frequency" setting.




Remote command:
CALCulate:MATH<m>:FFT:SPAN on page 1526

**Full span**
Displays the full frequency span.

Remote command:
CALCulate:MATH<m>:FFT:FULLspan on page 1526

**Start frequency**
Defines the start frequency of the displayed frequency span.

Remote command:
CALCulate:MATH<m>:FFT:STARt on page 1525

**Stop frequency**
Defines the stop frequency of the displayed frequency span.

Remote command:
CALCulate:MATH<m>:FFT:STOP on page 1525

**Span/RBW Coupling**
Couples the frequency span to the "Resolution BW" setting.

Remote command:
CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:AUTO on page 1527

**Span/RBW Ratio**
Defines the coupling ratio for Span/RBW. This setting is only available if `CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:AUTO` is ON.

Remote command:
`CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:RATio` on page 1528

**Resolution BW**
Defines the resolution bandwidth. Note that the resolution bandwidth is correlated with the span, record length and acquisition time. If a constant record length is to be used, the RBW may be adapted if the required number of samples cannot be acquired. If span and RBW values are coupled, changing the span will also change the RBW.

For details see Chapter 9.1.1, "Fundamentals of FFT analysis", on page 399.

Remote command:
`CALCulate:MATH<m>:FFT:BANDwidth[:RESolution][:VALue]` on page 1528
`CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:ADJusted?` on page 1527

**Window type**
Windowing helps minimize the discontinuities at the end of the measured signal interval and thus reduces the effect of spectral leakage, increasing the frequency resolution.

Various different window functions are provided in the R&S RTP to suit different input signals. Each of the window functions has specific characteristics, including some advantages and some trade-offs. Consider these characteristics carefully, to find the optimum solution for the measurement task.

| Window type | Frequency resolution | Magnitude resolution | Measurement recommendation |
|---|---|---|---|
| Rectangular | Best | Worst | Separation of two tones with almost equal amplitudes and a small frequency distance |
| Hamming<br>Hann | Good | Poor | Frequency response measurements, sine waves, periodic signals and narrow-band noise |
| Blackman Harris (default) | Worst | Best | Mainly for signals with single frequencies to detect harmonics<br>Accurate single-tone measurements |
| Gaussian | Good | Good | Weak signals and short duration |
| Flattop2 | Poor | Best | Accurate single-tone measurements |
| Kaiser Bessel | Poor | Good | Separation of two tones with differing amplitudes and a small frequency distance |

Remote command:
`CALCulate:MATH<m>:FFT:WINDow:TYPE` on page 1529

**Use color table**
If enabled, the spectrum waveform (and a spectrogram, if available) is displayed according to the assigned color table. For information on the available color tables, see Chapter 4.3.2, "Color tables", on page 79.

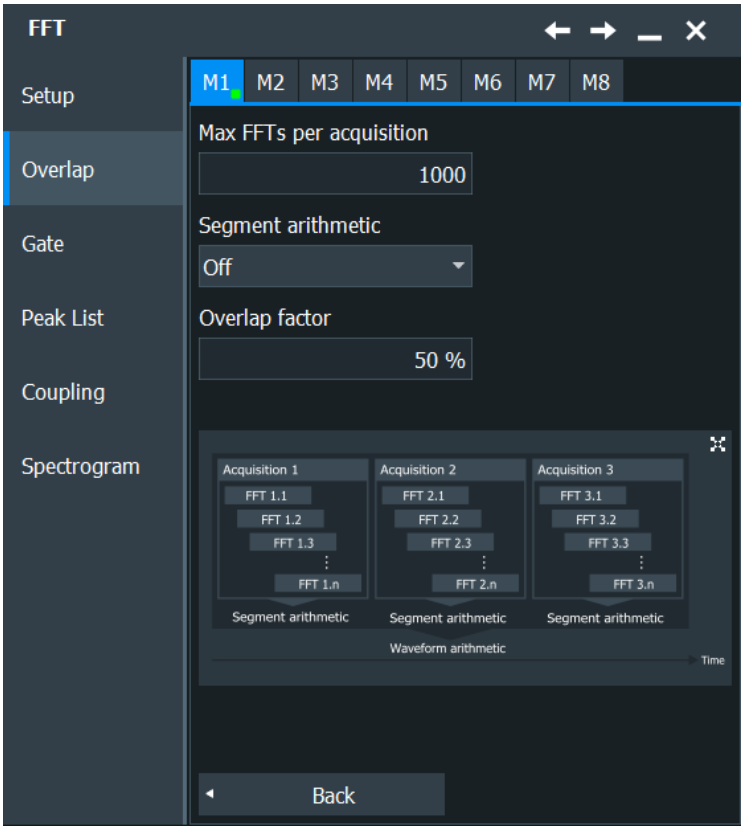
If this option is disabled, the preset color of the selected channel source is displayed, and the intensity of the specific signal color varies according to the cumulative occurrence of the values. For spectrum diagrams, this setting corresponds to the common waveform display. The spectrogram, on the other hand, is then displayed in gray tones, which is not useful.

Remote command:
`CALCulate:MATH<m>:FFT:USEColtab` on page 1540

### 9.1.3.2 FFT overlap

Access: [Spectrum] > "FFT setup" > "Overlap" tab

In this tab, you define the settings for the magnitude and phase of the frequencies.




Max FFTs per acquisition....411
Segment arithmetic....412
Overlap Factor....412

**Max FFTs per acquisition**
Restricts the maximum number of FFTs to be calculated for each data acquisition. Due to the other parameter settings, the required number of FFTs may become very high, thus slowing performance. By restricting the number of FFTs, you can avoid performance loss without changing the other parameters.

However, if the maximum number of FFTs is lower than the required number to cover the entire waveform, the waveform may only be analyzed partially. In this case, the "Frame coverage" indicates the percentage of the waveform that was analyzed, i.e. which part of the data was included in the FFT calculation.

Remote command:
`CALCulate:MATH<m>:FFT:FRAMe:MAXCount` on page 1530
`CALCulate:MATH<m>:FFT:FRAMe:COVerage?` on page 1530

**Segment arithmetic**
FFT analysis can only be performed on a maximum number of values at once. If more values must be calculated, the input signal is divided into segments, each of which is calculated separately. The segments need not be disjunct, that is: they may overlap, so that some values have several FFT results. In this case, the arithmetic mode defines how the final result is calculated from the individual results.

The following methods are available:

"Off" The data of only one segment is considered. In effect, no arithmetics are processed.

"Envelope" Detects the minimum and maximum values for FFT calculation over all segments. The resulting diagram shows two envelope waveforms: the minimums (floor) and maximums (roof). These envelopes indicate the range of all FFT values that occurred.

"Average" The average is calculated over all segments.

"RMS" The root mean square is calculated over all segments. The result is the average power spectrum. If you measure the channel power on this RMS spectrum, you get the same result as for the average channel power measurement on segments.

"MinHold" Determines the minimum result for each input value from the data of the current acquisition and the acquisitions before.

"MaxHold" Determines the maximum result for each input value from the data of the current acquisition and the acquisitions before.

Remote command:
`CALCulate:MATH<m>:FFT:FRAMe:ARIThmetics` on page 1530

**Overlap Factor**
Defines the minimum factor by which two neighboring segments overlap. If the required number of segments to cover the input values allows for more overlap, the factor is increased.

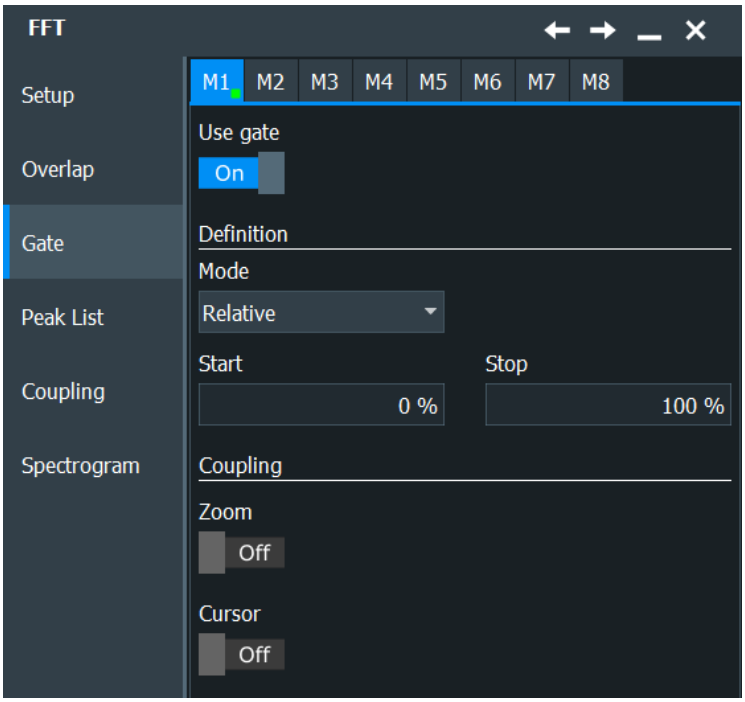
The higher the overlap factor, the more segments are used. This leads to more individual results and improves detection of transient signal effects. However, it also extends the duration of the calculation.

Remote command:
`CALCulate:MATH<m>:FFT:FRAMe:OFACtor` on page 1531

### 9.1.3.3 FFT gating

Access: [Spectrum] > "FFT setup" > "Gate" tab

FFT gating allows you to restrict FFT analysis to a certain time base of the input signal.




If no gate is used, you can define the record length as dependent on the RBW, or the RBW as dependent on the record length (which is defined by the acquisition time).

Use Gate ..... 413
Gate Definition ..... 413
└ Zoom ..... 414
└ Gate description ..... 414
└ (Relative) Start ..... 414
└ (Relative) Stop ..... 414
Record Length/RBW Coupling ..... 415
Required acquisition time ..... 415

**Use Gate**
Enables FFT gating and shows the gate.

If enabled, the "Gate Definition" settings are used.

If disabled, the relation between the record length and the RBW can be defined manually in the "Setup" tab instead.

When a gate is used, the RBW is adapted, if necessary. The smaller the gate, the higher the RBW.

For details, see Chapter 9.1.1, "Fundamentals of FFT analysis", on page 399.

**Gate Definition**
Defines the gate settings for FFT gating.

**Zoom ← Gate Definition**
Zoom coupling is available if a zoom is defined. As long as "Zoom coupling" is enabled, the gate area is defined identically to the zoom area - if you change the zoom, the gate changes as well.

If several zoom diagrams are defined, select the zoom diagram to be used for gating. The "Start" and "Stop" values of the gate are adjusted accordingly.

Zoom coupling can be set for measurement gates, FFT gates, and search gates. The zoom must be defined on the diagram that contains the signal source of the measurement, FFT, or search.

Remote command:
`MEASurement<m>:GATE:ZCOupling` on page 1505
`MEASurement<m>:GATE:ZDIagram` on page 1506
`CALCulate:MATH<m>:FFT:GATE:ZCOupling` on page 1534
`SEARch:GATE:ZCOupling` on page 1597
`SEARch:GATE:ZDIagram` on page 1597

**Gate description ← Gate Definition**
Defines whether the gate settings are configured using absolute or relative values.

"Absolute" The gate is defined by absolute start and stop values.

"Relative" The gate's start and stop values are defined by a percentage of the value range.

Remote command:
`CALCulate:MATH<m>:FFT:GATE:MODE` on page 1533
`MEASurement<m>:GATE:MODE` on page 1503
`SEARch:GATE:MODE` on page 1595

**(Relative) Start ← Gate Definition**
Defines the starting value for the gate.

Remote command:
`CALCulate:MATH<m>:FFT:GATE:ABSolute:STARt` on page 1532
`CALCulate:MATH<m>:FFT:GATE:RELative:STARt` on page 1533
`MEASurement<m>:GATE:ABSolute:STARt` on page 1504
`MEASurement<m>:GATE:RELative:STARt` on page 1504
`SEARch:GATE:ABSolute:STARt` on page 1596
`SEARch:GATE:RELative:STARt` on page 1596

**(Relative) Stop ← Gate Definition**
Defines the end value for the gate.

Remote command:
`CALCulate:MATH<m>:FFT:GATE:ABSolute:STOP` on page 1532
`CALCulate:MATH<m>:FFT:GATE:RELative:STOP` on page 1533
`MEASurement<m>:GATE:ABSolute:STOP` on page 1504
`MEASurement<m>:GATE:RELative:STOP` on page 1504
`SEARch:GATE:ABSolute:STOP` on page 1596
`SEARch:GATE:RELative:STOP` on page 1597

**Record Length/RBW Coupling**
The record length and resolution bandwidth are coupled during FFT analysis. If you change one value, the other must be adapted accordingly. You can keep either value constant, thus preventing automatic adaptation when the other parameter is changed. However, this may cause the FFT analysis to fail.

This setting is only available if gating is not enabled (otherwise the gate determines the RBW automatically).

For details, see Chapter 9.1.1, "Fundamentals of FFT analysis", on page 399.

| | |
|---|---|
| "Record length controlled" | The record length remains constant. If not enough samples are available for the selected RBW, the RBW is decreased. |
| "RBW controlled" | The RBW is not adapted, i.e. remains as defined. The required acquisition time for this RBW is indicated. If necessary and possible, the record length is extended to acquire the required number of samples. |

Remote command:
`CALCulate:MATH<m>:FFT:GATE:COUPling` on page 1531

**Required acquisition time**
The required acquisition time is calculated for the defined RBW value if "RBW constant" is selected, and is displayed for information only. If the required acquisition time is not available (e.g. because acquisition has already been stopped), an error message is displayed in the FFT setup tab indicating that not enough samples are available for the defined RBW.

Remote command:
`TIMebase:RACTime?` on page 1532

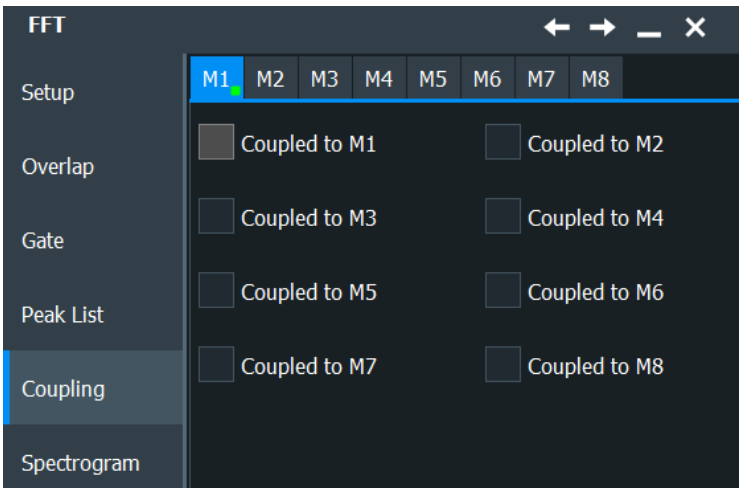
### 9.1.3.4 Peak list

For a description of the "Peak List" settings, see Chapter 8.2.7, "Spectrum measurements", on page 363.

### 9.1.3.5 FFT coupling

Access: [Spectrum] > "FFT setup" > "Coupling" tab

Up to four spectrum displays can be shown simultaneously, one for each math waveform. The settings for one or more spectrums can be coupled. Thus, if any FFT setting for any of the coupled spectrums are changed, they are changed for all coupled spectrums.




**Coupled to M1...M8**
Copies the current FFT settings of the selected math waveform (M1...M8) to the other selected math waveforms, and couples those waveforms. Thus, if any FFT setting for any of the coupled spectrums are changed, they are changed for all coupled spectrums.

Two different sets of spectrums can be coupled at the same time, for instance "M1" can be coupled to "M2", while "M3" is coupled to "M4".

Note that the formula of the coupled math waveforms is not changed. If necessary, you must select an FFT function for the math waveform manually before the FFT settings of the coupled waveform are applied. See Chapter 7.3.1, "Displaying math waveforms", on page 279.

Remote command:
`CALCulate:MATH<m>:FFT:COUPled:WITH<1..8>` on page 1537

### 9.1.4 FFT scale settings

In this section, you can find functions to set the vertical parameters of the FFT math waveform.

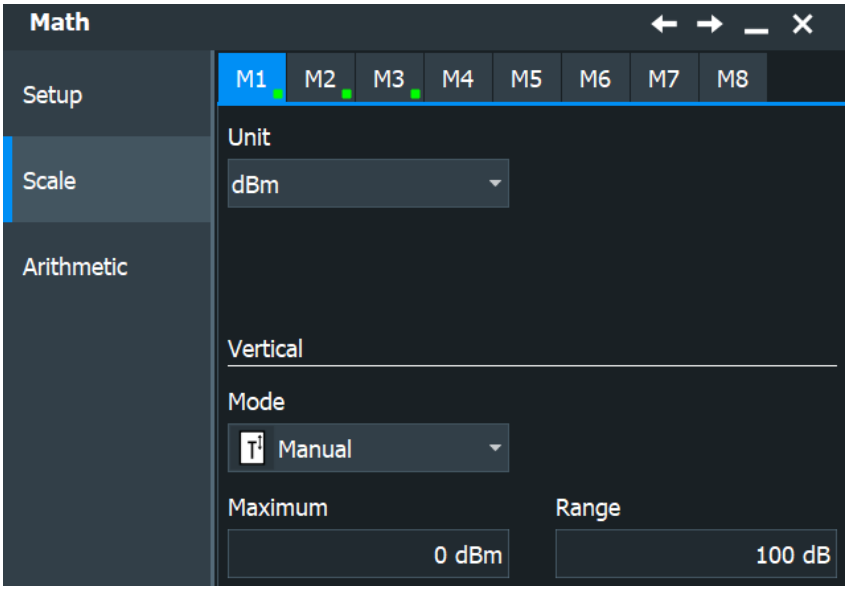



*Figure 9-1: Scale settings of magnitude spectrum*

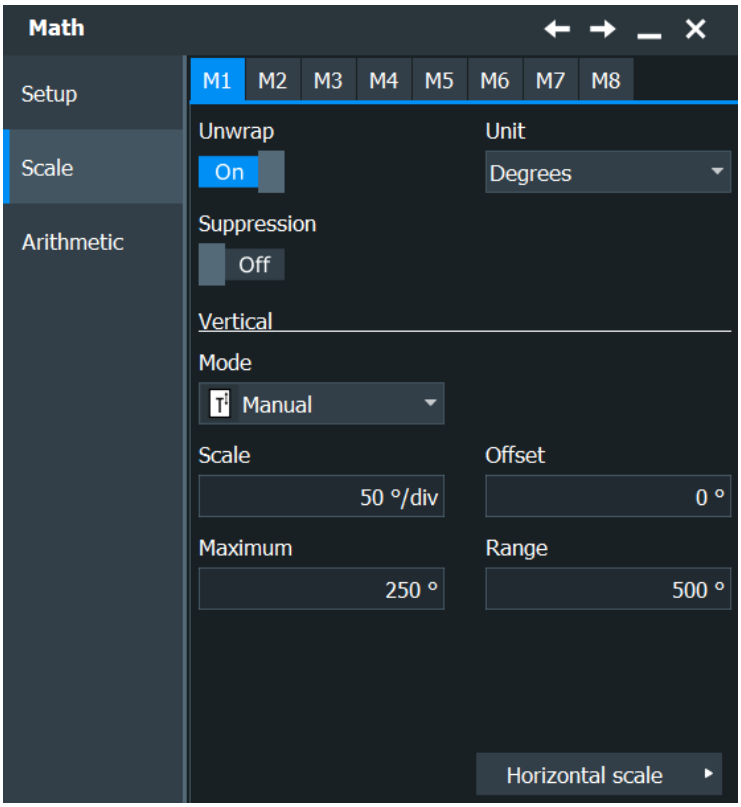



*Figure 9-2: Scale settings of phase spectrum*

See also:

- Vertical scaling mode (Manual/Auto)
- Vertical Offset

- Vertical Scale

**Unit**
Defines the scaling unit for magnitude display. Available units depend on the unit of the selected source waveform.

"Linear" displays the RMS value of the voltage.

Remote command:
CALCulate:MATH<m>:FFT:MAGNitude:SCALe on page 1535

**Reference level**
Defines the reference level if "dB" is the selected unit.

Remote command:
CALCulate:MATH<m>:FFT:MAGNitude:LEVel on page 1534

**Vertical maximum**
Defines the maximum value on y-axis for spectrum displays. Only available for "Manual" scale mode.

Remote command:
CALCulate:MATH<m>:VERTical:MAXimum on page 1403

**Vertical range**
Defines the range of FFT values to be displayed.

Remote command:
CALCulate:MATH<m>:VERTical:RANGe on page 1403

**Unwrap**
If enabled, phase shifts due to a limitation of the value range are eliminated.

Remote command:
CALCulate:MATH<m>:FFT:PHASe:UNWRap on page 1537

**Unit**
Defines the scaling unit for phase display.

- Radians
- Degrees

Remote command:
CALCulate:MATH<m>:FFT:PHASe:SCALe on page 1536

**Suppression**
Enables noise suppression. Phase calculation is restricted to frequencies with a minimum magnitude, the threshold value.

Remote command:
CALCulate:MATH<m>:FFT:PHASe:SUPPression on page 1536

**Threshold**
Defines the minimum frequency magnitude for which phases are calculated. This setting is only available if "Suppression" is enabled.

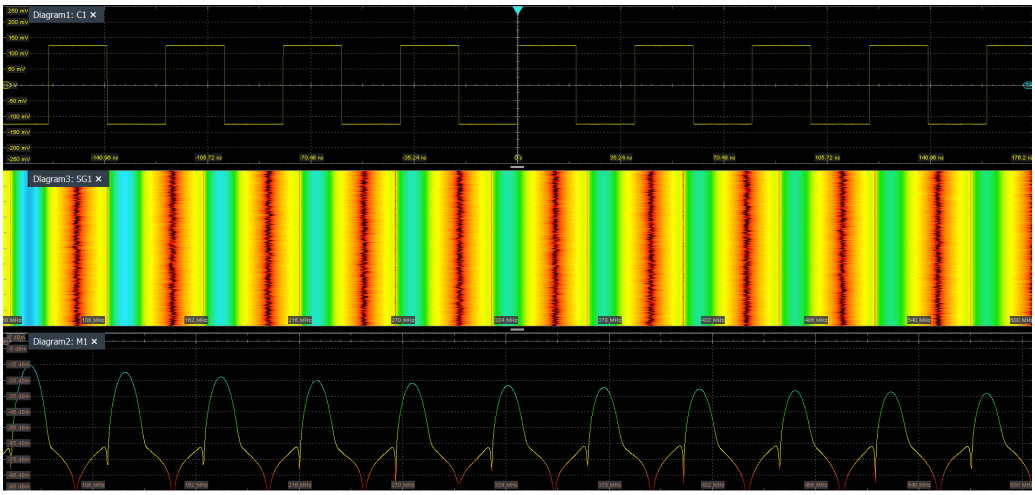
Remote command:
CALCulate:MATH<m>:FFT:PHASe:THReshold on page 1537

## 9.2 Spectrum analysis (option R&S RTP-K37)

This chapter describes the Spectrum analysis option R&S RTP-K37.

### 9.2.1 Spectrogram display

The Spectrum analysis option provides a new diagram for spectrum waveforms: a spectrogram. When you enable a spectrogram, three windows are displayed: the power vs. time diagram at the top, the spectrogram in the middle (labeled "SG") and the power vs. frequency (=spectrum) diagram at the bottom.




A spectrogram shows how the spectral density of a signal varies over time. The x-axis shows the frequency, the y-axis shows the time. A third dimension, the power level, is indicated by different colors. Thus you can see how the strength of the signal varies over time for different frequencies.

The spectrogram is updated with each data acquisition, from top to bottom, so that the most recent trace is at the bottom. Up to two time lines can be shown at a specified position so that you can analyze the spectrum at a specific point in time.

The spectrum diagram indicates the power vs. frequency values for a single data acquisition. If a time line is enabled, the spectrum shows the results at the selected time. Otherwise, the spectrum shows the results of the most recent data acquisition.

### 9.2.2 Spectrum analysis functions

In addition to spectrograms, the Spectrum analysis option also provides some new automatic measurements based on spectrum waveforms.

- A peak list measurement detects all peaks above a user-definable threshold and optionally indicates the peaks in the spectrum diagram.

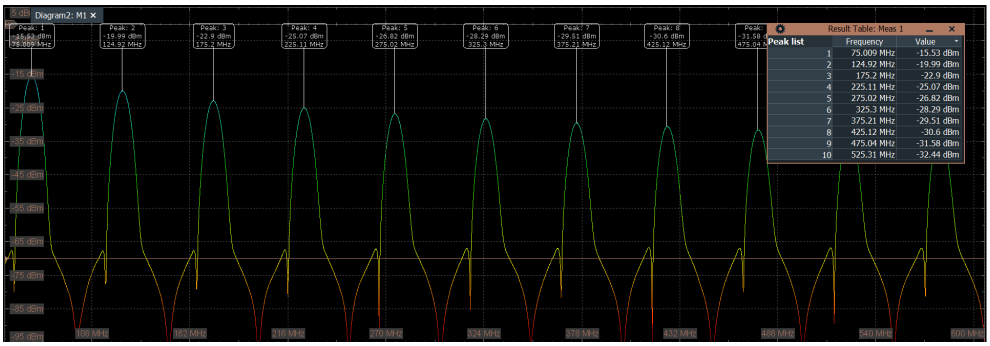



*Figure 9-3: Peak list with labels for spectrum waveform*

For a description of the measurement settings, see Chapter 8.2.7.2, "Settings for spectrum measurements", on page 365.

- The THD measurements are an extension to the basic THD measurement. See Chapter 8.2.7, "Spectrum measurements", on page 363 for details.
- Cursor measurements on spectrum waveforms provide easy center definition and peak search functions, see Chapter 8.1.1.2, "Cursor measurements on spectrum waveforms", on page 309.

### 9.2.3 Configuring spectrograms

Spectrograms are only available if the Spectrum analysis option R&S RTP-K37 is activated.

1. Tap the "Spectrogram" icon on the toolbar.

   The "Spectrogram" overlay menu opens.

2. Select the source of the spectrogram.

   A spectrogram diagram is displayed. A new signal icon for the spectrogram is displayed on the signal bar ("SGx").
   If the selected source is a channel waveform, an FFT is started, on which the spectrogram is created.
   Additional settings for time lines become available in the dialog box.

3. Optionally, to display a time line and thus mark a specific waveform in the spectrogram, select "Enable" for one of the two time lines.

A small arrow icon labeled "T1" / "T2" indicates the position of the time line in the spectrogram.
The spectrum diagram displays the results for the selected waveforms. A new signal icon is displayed on the signal bar for each time line ("SGxTL1|2").

4. To view the spectrum for each time line in a separate diagram, drag the signal icon for one time line to the diagram area and drop it.

   A new window is displayed for the selected time line, and the original diagram displays the other time line.

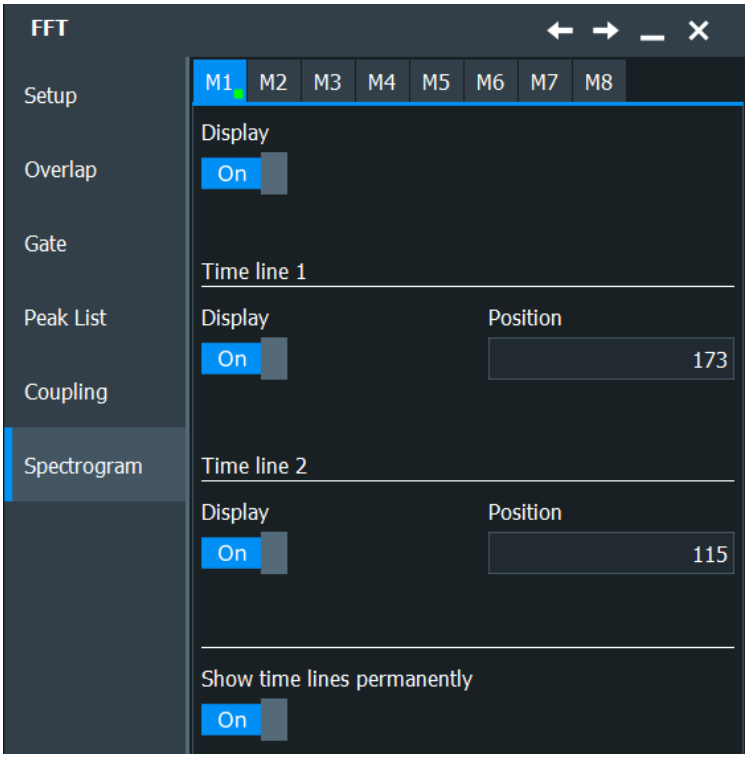
5. To view a different waveform from the spectrogram, move the time lines in the spectrogram.

### 9.2.4 Spectrogram configuration settings

Access: [Spectrum] > "FFT setup" > "Spectogram" tab

Spectrograms are only available if the option R&S RTP-K37 is activated. Furthermore, a math (FFT) waveform must be configured and enabled.

See Chapter 9.1.2, "Configuring spectrum waveforms", on page 403.




Display ..... 422
Time line 1/2 ..... 422
Show lines permanently ..... 422

**Display**
Enables the spectrogram display.

If enabled, a new signal icon for the spectrogram is displayed on the signal bar ("SGx").

Remote command:
`CALCulate:MATH<m>:FFT:SPECtrogram:STATe` on page 1540

**Time line 1/2**
A time line marks a single spectrum in the spectrogram, that is: the power vs frequency results for the data acquired at a specific time. After enabling a time line, the results for that time are displayed in the spectrum diagram. A small arrow icon labeled "T1" / "T2" indicates the position of the time line in the spectrogram. A new signal icon is displayed on the signal bar for each time line ("SGxTL1|2").

You can enable and display two time lines at the same time. This allows you to compare the results at different times.

The position of the time line is defined by the index of the data acquisition in the history. How many acquisitions are available depends on the history settings.

See "Player" on page 300.

Remote command:
`CALCulate:MATH<m>:FFT:SPECtrogram:TIMeline<n>:STATe` on page 1541
`CALCulate:MATH<m>:FFT:SPECtrogram:TIMeline<n>:POSition` on page 1541

**Show lines permanently**
Displays the spectrogram time lines in the diagrams until you disable this option.

If disabled, only the small arrow icons are permanently visible. The line is only displayed temporarily when you touch the arrow.

# 10 Mask testing

## 10.1 About mask testing

Masks are used to determine whether the signal remains within specified limits, e.g. to uncover signal anomalies or test compliance and stability of digital signals. The limits are specified as "mask", which is laid over the input signal in the display. Thus you can easily detect where the signal violates the mask.

Mask testing with R&S RTP has only a minor impact on the acquisition rate, thus mask violations are detected fast and reliably.

With R&S RTP, you can define own masks easily. Specific actions can be executed when mask violations occur. For error analysis, you can stop the acquisition on a failed test and use the history view to look at the previous waveforms.

**Mask test**

A mask test consists of:

- Mask definition
- Waveform to be tested
- Fail criteria for test
- Actions to be taken on violation or successful completion

**Mask Definition**

A mask can be created in several ways:

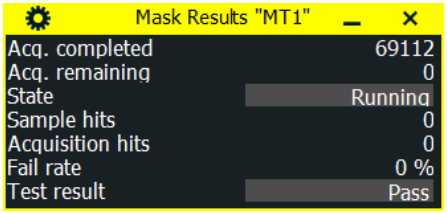
- The individual mask points are defined, either on the touch screen or as numerical values. This mask type is called *user mask*.
  For details, see Chapter 10.2.2.1, "Mask definition: user mask", on page 427
- The mask is derived from an existing waveform. This mask type is called *waveform mask*.
  For details, see Chapter 10.2.2.2, "Mask definition: waveform mask", on page 430.

**Fail Criteria for Testing**

The fail criteria for a mask test is set by two parameters: "Fail condition" and "Violation tolerance". Fail condition defines if sample hits or the number of acquisitions with sample hits are considered. Violation tolerance sets the number of tolerable sample hits or acquisition hits. A test has failed if the number of sample hits or acquisition hits exceeds the limit of violation tolerance hits.

### 10.1.1 Results of a mask test

The result box of a mask test shows the following test results:




**Acq. completed**
Number of tested acquisitions.

Remote command:
`MTESt:RESult:COUNt:WAVeforms?` on page 1559

**Acq. remaining**
Remaining acquisitions until "Average count / Nx Single count" is reached.

The value is useful if you test a specified number of acquisitions with action "Stop acquisition" on violation. Also if the acquisition has been stopped manually before the required number of acquisitions has been acquired.

Remote command:
`MTESt:RESult:COUNt:REMaining?` on page 1559

**State**
Shows if the test has been completed. The state is set to "Finished" when "Nx Single count" acquisitions are tested and the number of "Acq. remaining" is 0. as long as the number of tested acquisitions is less the "Nx Single count" number, the state is "Running".

If you run the acquisition with [Run / Stop], or the number of played history acquisitions exceeds "Nx Single count", the mask testing is performed according to fail criteria settings independently of the test state. The testing is not stopped when the state is set to "Finished".

Remote command:
`MTESt:RESult:STATe?` on page 1558

**Sample hits**
Number of samples that hit the mask.

Remote command:
`MTESt:RESult:COUNt:FAILures?` on page 1560

**Acquisition hits**
Number of acquisitions that contained at least one sample hit.

Remote command:
`MTESt:RESult:COUNt:FWAVeforms?` on page 1559

**Fail rate**
Ratio of acquisition hits to the number of tested acquisitions.

Remote command:
`MTESt:RESult:FRATe?` on page 1560

**Test result**
A test has failed if the number of sample hits or acquisition hits exceeds the limit of "Violation tolerance" hits.

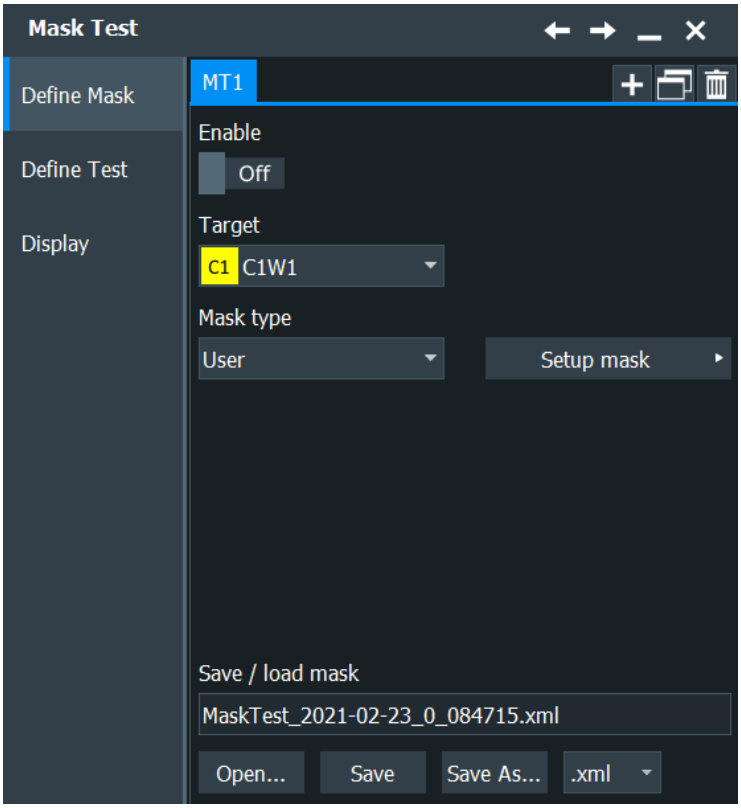
Remote command:
`MTESt:RESult[:RESult]?` on page 1558

## 10.2 Mask test settings

### 10.2.1 Define mask

Access: [Apps] > "Analysis" tab > "Mask" > "Define Mask"

The "Define Mask" tab provides all settings for the mask test itself: the waveform to be tested,and saving/loading the mask definition.




The content of the "Test Definition" tab depends on the selected definition type. If "Waveform" is selected, the main mask settings can be set directly on the "Test Definition" tab. For a description of these settings, see Chapter 10.2.2.2, "Mask definition: waveform mask", on page 430.

Make sure that the correct "Mask Test" tab is selected on the left side before you enter the settings.

Remote commands:

MTESt:ADD on page 1542

MTESt:REMove on page 1543

**Enable**
Activates and deactivates the mask test. If the acquisition is running, the test starts immediately. Otherwise, the test starts when acquisition is started.

The testing is stopped when acquisition is stopped, or if a stop action is configured with "Stop acq." on page 435.

Closing the result box also disables the mask test.

Remote command:
MTESt[:STATe] on page 1543

**Target**
Selects the waveform to be tested against the mask. All channel waveforms can be tested.

Remote command:
MTESt:SOURce on page 1544

**Mask Type**
Sets the method of mask definition.

| | |
|---|---|
| "User" | The mask is created manually by tapping the mask points on the touch screen and/or by entering the numerical x- and y-values of the mask points. |
| "Waveform" | The mask is created from an existing waveform. The waveform builds the upper and lower limit line of the mask, and the limits are moved and stretched. The result is a tolerance tube around the waveform that is used as mask. |
| "Eye" | Requires jitter option R&S RTP-K12.<br>The mask is created by selecting the shape and setting its dimensions according to the test standard. |

Remote command:
MTESt:CTYPe on page 1545

**Setup mask**
Opens a dialog for defining the mask. See Chapter 10.2.2, "Mask definition", on page 427.

**Save / load mask**
Provides all functions to store and recall a mask test. The mask definition, defined actions and fail conditions are stored in an R&S RTP-specific .xml file.

"Open" Opens a dialog box where you can select the file name.
See also: Chapter 12.5, "File selection dialog", on page 513.

"Save, Save As" Stores the specified file.

Remote command:
`MTESt:FILE:NAME` on page 1545
`MTESt:FILE:SAVE` on page 1546
`MTESt:FILE:OPEN` on page 1546
`MTESt:FILE:DELete` on page 1546

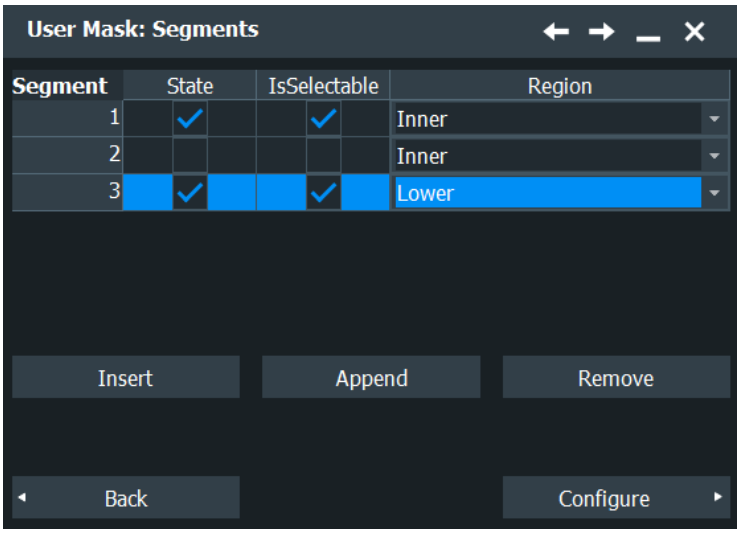
## 10.2.2 Mask definition

With mask definition, you define the shape of the mask - the form and position of its limit lines. The content of the "Mask Definition" tab depends on the selected Mask Type: "User" or "Waveform".

Below, you find the specific settings:

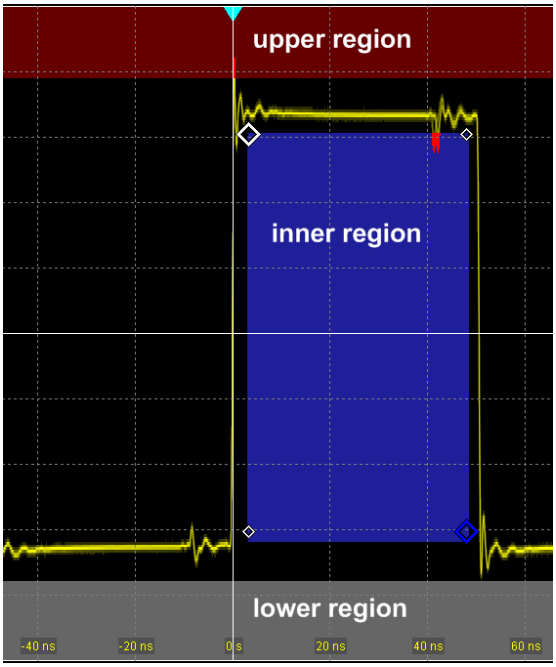
### 10.2.2.1 Mask definition: user mask

Access: [Apps] > "Analysis" tab > "Define Mask" > "Mask Type" = "User" > "Setup mask"




A user mask is defined by entering the time and voltage values for all corner points of the mask segments. A user mask has at least one segment. Complex masks can have up to 16 segments.

An inner segment is an area defined by three or more points. Upper and lower segments limit the signal on top and bottom of the screen. They are defined by a line, the region above or below the line is set automatically as mask segment.




Alternatively, you can set the corner points on the touch screen and adjust the values in the "Mask Definition" tab.

To save the mask, select the "Define Mask" tab and save the mask test.

Make sure that the correct "Mask Test" tab is selected on the left side before you enter the settings.

**Mask segments**
Defines the number and state of mask segments for the selected mask test. Here you can:

- Insert a new segment before the selected segment.
- Append a new segment at the end of the list.
- Remove the selected mask segment from the list.
- Select the region that builds the mask.
  - Inner region: the segment points form a closed geometrical shape, which is the mask segment.
  - Upper region: the segment points are connected to a line, the display area above this line is the mask segment.
  - Lower region: the segment points are connected to a line, the display area below this line is the mask segment.
- Enable and disable the mask segments individually. Disabled segments are not considered by running tests.

Remote command:
`MTESt:SEGMent:STATe` on page 1547
`MTESt:SEGMent:ADD` on page 1547
`MTESt:SEGMent:REMove` on page 1548
`MTESt:SEGMent:INSert` on page 1548

MTESt:SEGMent:REGion on page 1549
MTESt:SEGMent:COUNt? on page 1548

**Configure**
Opens a dialog to define the selected segment.

**Definition of segment**
The number of the selected segment is shown above the table. In the definition table, the individual points of the selected mask segment are listed with exact horizontal and vertical numerical coordinates. Here you can:

- Insert a new point before the selected point.
- Append a new point at the end of the list.
- Remove the selected point from the list.
- Change the x- and y-values of each point. To scale or move the complete segment, use offset and factor values, see Rescale.

Remote command:
MTESt:SEGMent:POINt:ADD on page 1549
MTESt:SEGMent:POINt:REMove on page 1550
MTESt:SEGMent:POINt:INSert on page 1549
MTESt:SEGMent:POINt:X on page 1550
MTESt:SEGMent:POINt:Y on page 1551
MTESt:SEGMent:POINt:COUNt? on page 1550

**Rescale**
You can rescale and move mask segments by numerical input of factors and offsets.

The values change the selected mask segment and take effect on "Apply".

**Offset X ← Rescale**
Moves the mask segment horizontally. The specified offset is added to the x-values of all points of the selected mask segment.

To take effect, tap "Apply".

Remote command:
`MTESt:SEGMent:RESCale:XOFFset` on page 1552

**Factor X ← Rescale**
Stretches or compresses the selected mask segment in horizontal direction. The x-values of all points of the selected mask segment are multiplied with this factor. Factors >1 stretch the mask segment, while factors between 0 and 1 compress it. Negative values are possible and change the algebraic sign.

To take effect, tap "Apply".

Remote command:
`MTESt:SEGMent:RESCale:XFACtor` on page 1552

**Offset Y ← Rescale**
Moves the mask segment vertically. The specified offset is added to the y-values of all points of the selected mask segment.

To take effect, tap "Apply".

Remote command:
`MTESt:SEGMent:RESCale:YOFFset` on page 1553

**Factor Y ← Rescale**
Stretches or compresses the selected mask segment in vertical direction. The y-values of all points of the selected mask segment are multiplied with this factor. Factors >1 stretch the mask segment, while factors between 0 and 1 compress it. Negative values are possible and change the algebraic sign.

To take effect, tap "Apply".

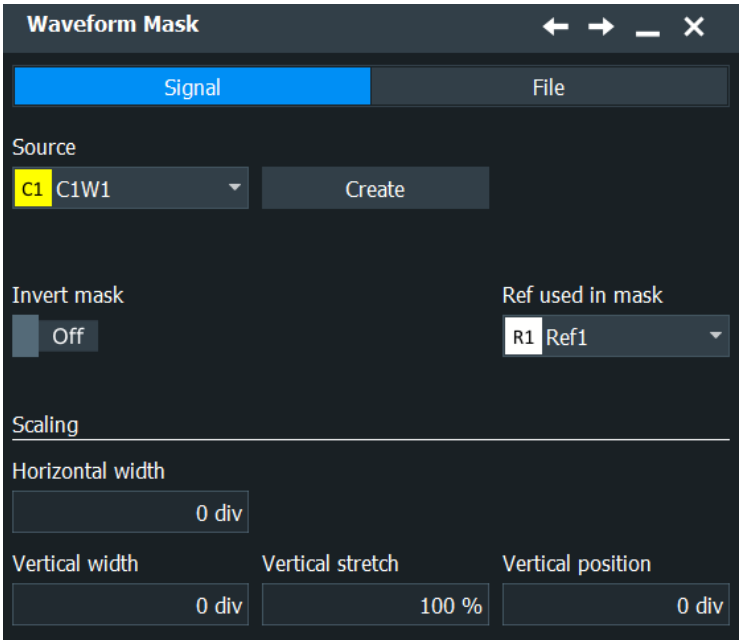
Remote command:
`MTESt:SEGMent:RESCale:YFACtor` on page 1552

**Apply ← Rescale**
Multiplies and adds the given x- and y-factors and offsets to the coordinates of all points of the selected mask segment.

Remote command:
`MTESt:SEGMent:RESCale:RECalculate` on page 1551

#### 10.2.2.2 Mask definition: waveform mask

Access: [Apps] > "Analysis" tab > "Define Mask" > "Mask Type" = "Waveform" > "Setup mask"




A waveform mask is created from an existing waveform. The waveform builds the upper and lower limit line of the mask, and the limits are moved and stretched. The result is a tolerance tube around the waveform that is used as mask.

During mask testing using a waveform mask, the record length is limited to 1 MSample.

The source for a waveform mask is a reference waveform. The reference waveform can be defined before mask definition, or loaded from a file, or it is created from the waveform to be tested.

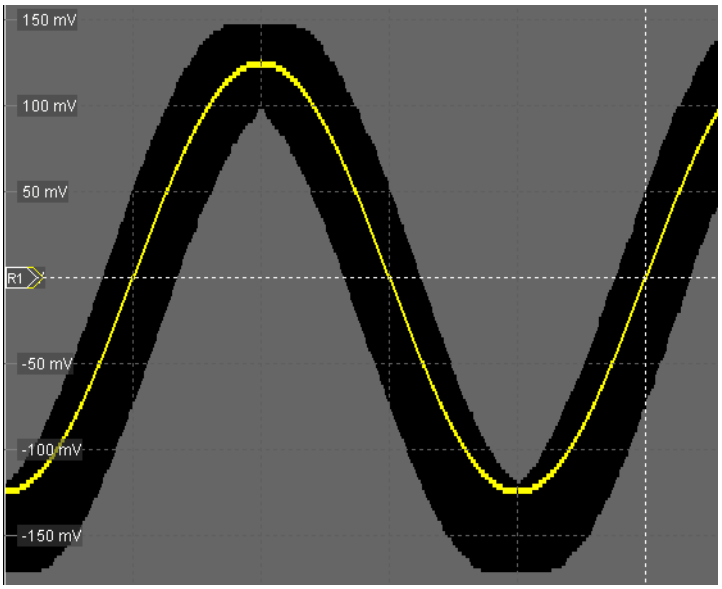



*Figure 10-1: Waveform mask*

**Signal**
The reference waveform is created from the selected "Source".

Sets the reference waveform from which the mask is created.

**Create ← Signal**
Creates the upper and lower mask limit from the "Source" and "Ref used in mark". If the reference waveform was not defined before, it is created automatically from the mask test "Source" waveform.

**File**
Loads the waveform from the selected file to the "Reference" and creates the mask immediately.

**Invert mask**
If enabled, the area inside the mask is highlighted, which the signal must not exit. If disabled, the area outside the mask limits is highlighted.

**Horizontal width**
Sets the width of the mask in horizontal direction. The specified number of divisions is added to the positive x-values and subtracted from the negative x-values of the mask limits in relation to the source waveform of the mask. The overall mask width is twice the specified horizontal width.

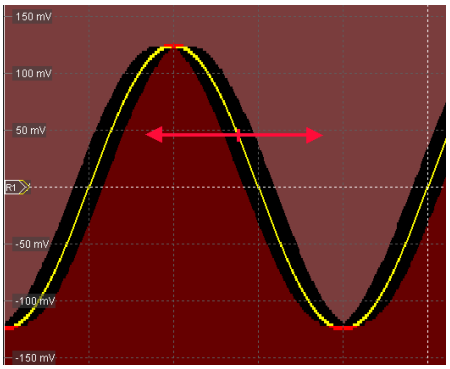



*Figure 10-2: Waveform mask with horizontal width = 0.2 div*

Remote command:
`MTESt:WFMRescale:XWIDth` on page 1554

**Vertical width**
Sets the width of the waveform mask in vertical direction. The specified number of divisions is added to the y-values of the upper mask limit and subtracted from the y-values of the lower mask limit. Thus, the upper half of the mask is pulled upwards, the lower half is pulled down, and the overall height of the mask is twice the vertical width.

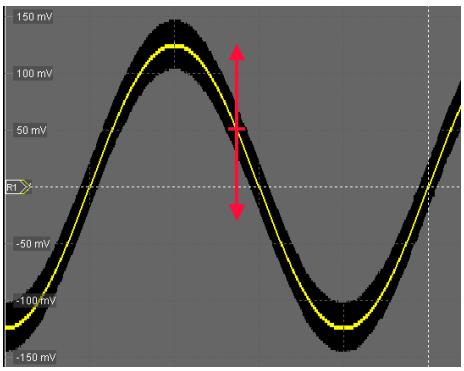



*Figure 10-3: Waveform mask with vertical width = 0.5 div*

Remote command:
MTESt:WFMRescale:YWIDth on page 1554

**Vertical stretch**
Sets the vertical scaling to stretch the mask in y-direction. The scaling axis is the horizontal line through the lowest value of the lower mask limit. Values > 100% stretch the mask, and values < 100% compress it.

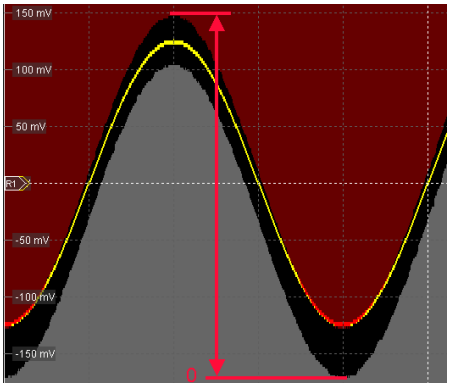



*Figure 10-4: Waveform mask with vertical width = 0.5 div, vertical position = -0.5 div, vertical stretch = 110%*

Remote command:
MTESt:WFMRescale:YSTRetch on page 1555

**Vertical position**
Moves the mask vertically within the display.

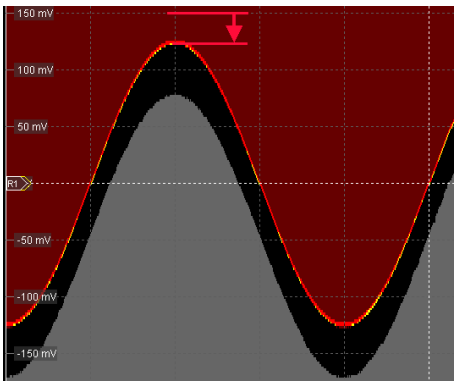
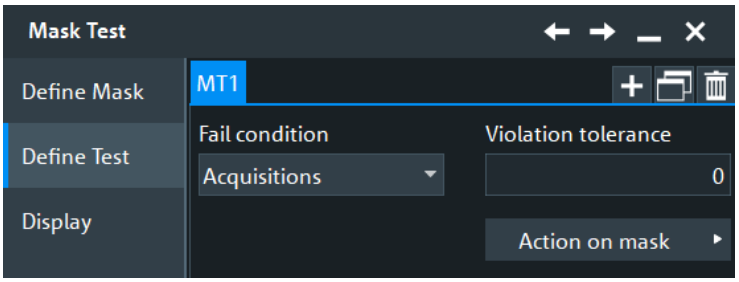



*Figure 10-5: Waveform mask with vertical width = 0.5 div and vertical position = -0.5 div*

Remote command:
`MTESt:WFMRescale:YPOSition` on page 1555

### 10.2.3 Define Test

Access: [Apps] > "Analysis" tab > "Mask" > "Define Test"




There are two usual test practices:

- Testing a defined number of waveforms against the mask and initiate an action when the acquisition cycle has been completed without failure:
  - Set the number of acquisitions to be tested: "Average count (N-single count)"
  - Start [Single]
- Testing a continuous acquisition or a defined number of waveforms against the mask and initiate an action when the fail criteria is fulfilled.

Make sure that the correct mask test tab is selected before you enter the settings.

**Fail condition, Violation tolerance**
The fail criteria for a mask test is set by two parameters: "Fail condition" and "Violation tolerance".

"Fail condition" defines the kind of hits to be considered for test evaluation:

- "Samples": Considers the number of samples that hit the mask.

- "Acquisitions": Considers the number of acquisitions that contain at least one sample hit. How many samples hit the mask in that acquisition is not relevant.

"Violation tolerance" sets the number of tolerable sample hits or acquisition hits.

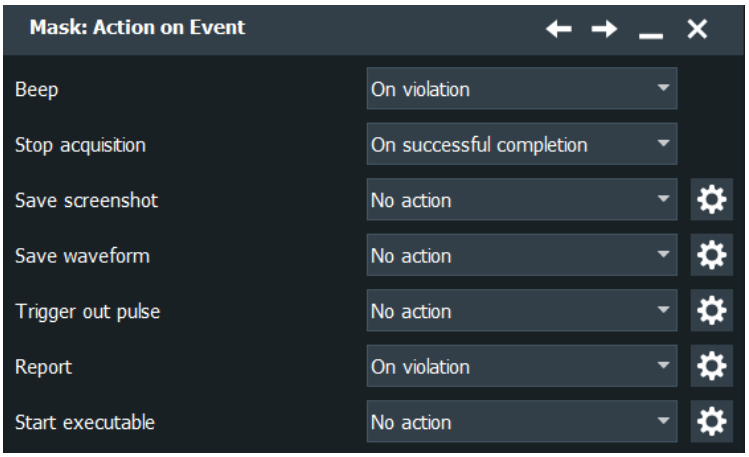
A test has failed if the number of sample hits or acquisition hits exceeds the limit of violation tolerance hits.

Remote command:
`MTESt:CONDition` on page 1544
`MTESt:TOLerance` on page 1545

**Action on mask**
The settings in this dialog define what happens when the mask test has failed or when it has passed successfully.

Most actions can be initiated either on failure or on success:

- On violation
  The action is initiated when the fail criteria is fulfilled.
- On successful completion
  The action is initiated when the [Single] acquisition has finished and the fail criteria is not fulfilled - the fail condition and violation tolerance limit have not been reached.




**Beep ← Action on mask**
Generates a beep sound.

Remote command:
`MTESt:ONViolation:BEEP` on page 1555

**Stop acq. ← Action on mask**
Stops the waveform acquisition on mask violation.

Remote command:
`MTESt:ONViolation:STOP` on page 1556

**Save screenshot ← Action on mask**
Saves a screenshot including the mask test results according to settings in "Menu" > "Save/Recall" > "Save" tab > "Screenshot".

Remote command:
MTESt:ONViolation:PRINt on page 1556

**Save waveform ← Action on mask**
Saves the waveform data to a file according to settings in "Menu" > "Save/Recall" > "Save" tab > "Waveforms".

Remote command:
MTESt:ONViolation:SAVewaveform on page 1556

**Trigger out pulse ← Action on mask**
Sends a pulse to the [Trigger Out] connector on the rear panel.

You can adjust the pulse by tapping the settings icon, see "Trigger out signal setup" on page 230.

Remote command:
MTESt:ONViolation:TRIGgerout on page 1557

**Report ← Action on mask**
Creates and saves a report using the settings.

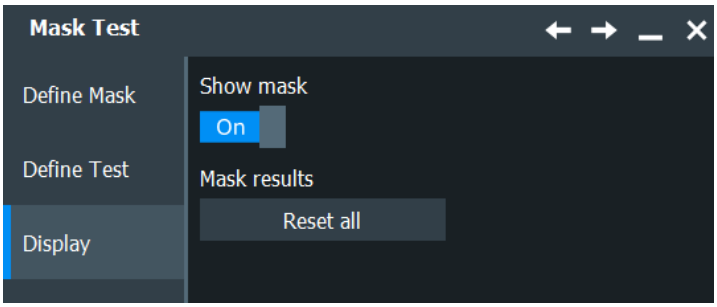
Remote command:
MTESt:ONViolation:REPort on page 1557

**Start Executable ← Action on mask**
Starts an external application. Tap "Config Executable" to set the application path and parameters.

Remote command:
MTESt:ONViolation:RUNexec on page 1557

### 10.2.4 Mask display

Access: [Apps] > "Analysis" tab > "Mask" > "Display".




In this dialog, you can reset the mask results and display the mask.

**Show mask**
Switches the display of all mask segments on or off.

Remote command:
MTESt:SHOW on page 1543

**Mask results > Reset all**
Clears all totals and results in all "Mask Test" result boxes.

Remote command:
`MTESt:RST` on page 1543

### 10.2.5 Mask appearance

You can define additional mask appearance settings in the "Settings" > "Appearance" > "Mask" dialog.

See Chapter 4.3.10, "Mask appearance settings", on page 90

## 10.3 Mask testing on eye diagrams

To perform mask testing on eye diagrams, the option R&S RTP-K12 adds a special mask definition type to the common mask definitions: the definition type "Eye".

Using the "Eye" mask type, you can easily define eye masks of various shapes. You can also use "User" masks to define free mask shapes.

### 10.3.1 Results of mask tests on advanced eye diagrams

If option R&S RTP-K136/K137 is installed, you can perform mask tests on advanced eye diagrams Eye1 to Eye4. The R&S RTP tests each waveform slice that contributes to the eye against the mask. The test results are listed in the mask test result box. Some results differ from the usual mask test results.

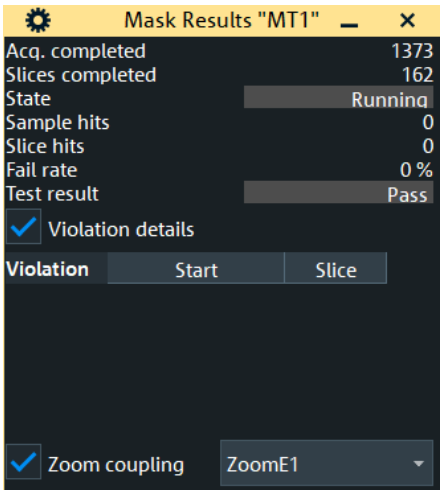



*Figure 10-6: Mask test results of eye testing*

Slices completed = Number of slices that contributed to the eye diagram
Sample hits = Number of samples that hit the mask
Slice hits = Number of violated slices
Violation details = Shows the table with violations of the last acquisition
Zoom coupling = Shows the selected violation in an existing zoom diagram

All other results are the same as in normal mask tests, see Chapter 10.1.1, "Results of a mask test", on page 423.

Remote commands are described in Chapter 24.23.7, "Results of mask tests on eye diagrams", on page 2429.

**To analyze mask violation details**

Prerequisites: The advanced eye diagram is defined, and the eye mask is configured.

1. Press [Run / Stop] to run the acquisition and the mask test on the advanced eye diagram.
2. Stop the mask test. You can also use the event action "Stop acq." = "On violation".
3. Create a zoom on the data waveform.
4. In the mask test result box, enable "Violation details".

   All violated segments are listed under "Violation".
5. In the mask test result box, enable "Zoom coupling". The function is available if at least one zoom diagram is active.
6. If you have several zooms, select the zoom that you want to couple.
7. In the "Violation details" table, tap a line.

   The zoom area is moved to the violated slice, and you can see the signal details in the zoom diagram.

## 10.3.2 Test definition for eye mask tests

Access: [Apps] > "Analysis" tab > "Define Mask" tab

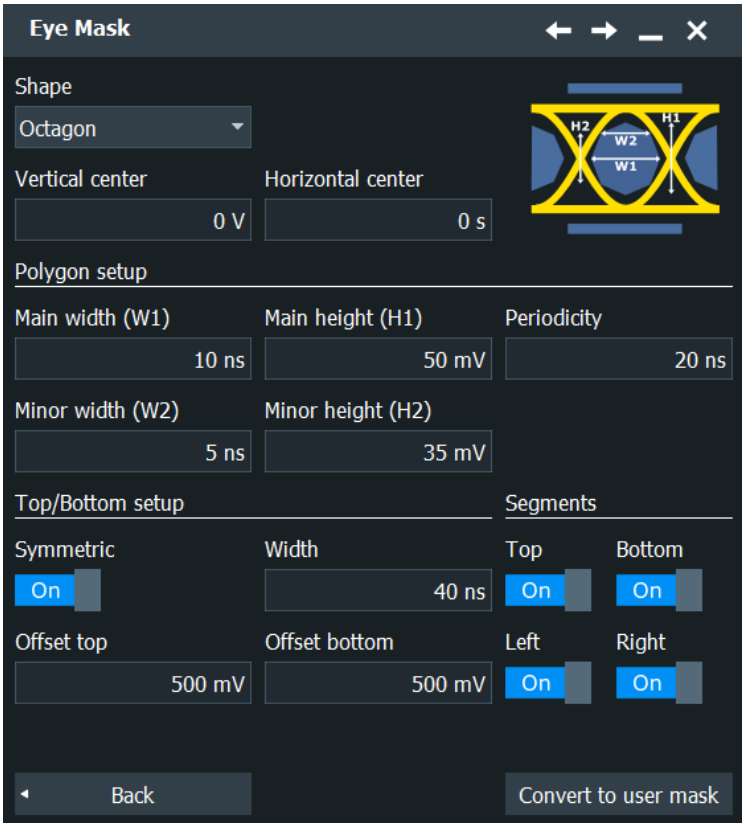
1. Select the "Source": the channel waveform that is analyzed for jitter.
2. Select the "Mask Type" = "Eye".
3. In the "Define Test" tab, adjust the "Fail condition" and "Violation tolerance".
4. Define the eye mask. See Chapter 10.3.3, "Eye mask definition", on page 439.

The general mask test settings are described in Chapter 10.2, "Mask test settings", on page 425.

### 10.3.3 Eye mask definition

Access: [Apps] > "Analysis" tab > "Define Mask" > "Mask Type" = "Eye" > "Setup mask"

The "Eye Mask " dialog provides all settings to define masks of different shapes. It is available if option R&S RTP-K12 is activated on the instrument.




**Shape**
Defines the outline of the eye mask: square, diamond, hexagon or octagon.

Remote command:
`MTESt:EYEMask:TYPE` on page 1561

**Horizontal center/Vertical center**
Set the horizontal (time) and vertical (voltage) values of the eye shape enter and thus define the position of the eye shape on the display.

Remote command:
`MTESt:EYEMask:HPOSition` on page 1564
`MTESt:EYEMask:VPOSition` on page 1565

**Main width (W1), Minor width (W2)**
Main width defines the width of all eye mask shapes. Minor width defines the secondary width for hexagon and octagon mask shapes.

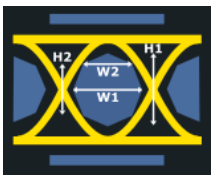



*Figure 10-7: Main and minor widths and heights of an octagon eye mask*

Remote command:
`MTESt:EYEMask:WIDTh<m>[:VALue]` on page 1562

**Main height (H1), Minor height (H2)**
Main height defines the height of all eye mask shapes. Minor height defines the secondary height for octagon mask shapes.

If an octagon shape is selected, and both values are the same, the eye masks looks like a hexagon rather than an octagon.

Remote command:
`MTESt:EYEMask:HEIGht<m>[:VALue]` on page 1561

**Periodicity**
Sets the time distance between the shape centers.

Remote command:
`MTESt:EYEMask:HPERiod` on page 1562

**Top/Bottom setup**
Defines the settings for the top and bottom segments of the mask. You have to enable "Top" / "Bottom" for this settings to be relevant.

**Symmetric ← Top/Bottom setup**
Sets bottom and top offsets to the same value so that the outer regions are symmetric to the eye shape.

Remote command:
`MTESt:EYEMask:TBSYmmetric` on page 1564

**Width ← Top/Bottom setup**
Sets the time width of the outer regions, symmetric to the eye shape center.

Remote command:
`MTESt:EYEMask:TBWidth` on page 1564

**Offset top, Offset bottom ← Top/Bottom setup**
Sets the voltage distance from the eye shape center that limits the upper and lower regions.

Remote command:
`MTESt:EYEMask:BOFFset` on page 1563
`MTESt:EYEMask:TOFFset` on page 1563

**Segments**
Enable the different regions of the mask: "Top", "Bottom", "Left", "Right".

The icons copy the eye shape to the left and to the right.

Remote command:
`MTESt:EYEMask:MSKBottom` on page 1563
`MTESt:EYEMask:MSKTop` on page 1563
`MTESt:EYEMask:MSKLeft` on page 1562
`MTESt:EYEMask:MSKRight` on page 1562

**Convert to user mask**
Converts the test and mask definitions of the current mask test to a new mask test of type "User".

The new test is opened in a separate mask test tab, in the "Define Mask" tab.

This is useful, for example, if want to run variants of a mask test in parallel.

Remote command:
`MTESt:CEMask` on page 1561

## 10.4 Working with masks

This chapter explains step-by-step how mask tests are setup and preformed. For the explanation of the individual settings, see Chapter 10.2, "Mask test settings", on page 425.

- Setting up user masks....441
- Setting up a mask test....446
- Configuring the mask and hit display....446
- Running a mask test....447
- Saving and loading masks....447
- Mask testing on history acquisitions....448

### 10.4.1 Setting up user masks

#### 10.4.1.1 Creating user masks

There are two ways to create a new mask:

- Graphical way by tapping the mask points on the touchscreen,
- Numerical entry of the x- and y-values of the mask points.

You can combine both methods. For example, at first you enter the mask quickly on the touchscreen, and then modify the point coordinates with precise values.

**To create a mask graphically on the touch screen**

1. Tap the "Masks" icon on the toolbar.

The "Masks" overlay menu opens.

2. Tap the corner points of the mask segment on the touch screen.

   **Tip:** To create an exact rectangle, draw the diagonal of the rectangle on the screen.

3. Tap "Finish segment".

   Now you can enter another segment to the current mask test or add a new mask test.

4. To finish the mask definition, close the overlay menu.
   If the acquisition is running, the mask test starts automatically and immediately.

**To create a mask numerically in the dialog box**

The settings mentioned here are described in detail in Chapter 10.2.1, "Define mask", on page 425.

1. Open [Apps] > "Analysis" tab > "Mask" > "Define Mask" tab.
2. Create a mask test:
   a) Tap the "+"-icon in the upper right corner.
   b) Enter a name for the new mask test.

   A new, empty tab for the mask test appears.
3. Set "Mask type"= "User".
4. Tap "Setup mask".
5. In the "User Mask: Segments" area, tap "Insert" to create a new mask segment.
6. Set the corner points of the mask segment:
   a) Select the "Segment".
   b) Tap "Configure". The "User Mask: Points" dialog opens.
   c) In the "Definition of segment" area, tap "Insert".
      Point 1 appears.
   d) Tap the X-cell and enter the X-value of the point.
   e) Tap the Y-cell and enter the Y-value of the point.
   f) To insert the next point:
      - Tap "Insert" to add a point before the selected point.
      - Tap "Append" to add a point at the end of the list.
   g) Set the X- and Y-values for this point.
   h) Repeat the last two steps until you define all points.
7. In the "Define Mask" tab, tap "Enable".
8. Adjust the horizontal and vertical units if necessary.

### 10.4.1.2 Modifying user masks

To change an existing mask definition, you can also use the graphical method on the touch screen, or the numerical way, or combine both.

With the graphical method, you can:

- Move, add, and delete segments
- Move and delete points

Adding points to an existing segment graphically is not possible.

With the numerical method, in the "Define Mask" tab, you have all modification possibilities. You can delete and add points and segments, change the coordinates, and also stretch a segment, or move it by adding an offset.

**To add a mask segment on the touch screen**

1. Tap a mask segment of the mask test that you want to complement.
2. Tap the "Masks" icon on the toolbar.

3. Tap the corner points of the new mask segment on the touch screen.
4. To finish the segment and mask definition, double-tap the last point.

**To delete a mask segment on the touch screen**

1. On the toolbar, tap the "Delete" icon.

2. Tap the mask segment you want to delete.

**To delete a point on the touch screen**

1. Tap the mask segment from which you want to delete a point.

   The selected segment is now in definition mode, shown with blue color.
2. On the toolbar, tap the "Delete" icon.

   The "Delete" overlay menu opens.
3. Tap the point you want to delete.

**To move a segment on the touch screen**

1. Drag&drop the segment to the new position.
2. Tap outside the mask to deselect the mask segment.

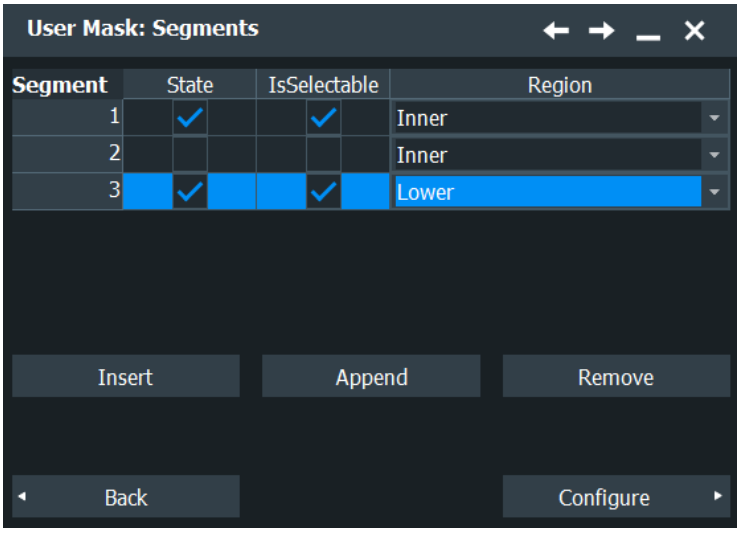
**To move a point on the touch screen**

1. Tap the mask segment that you want to change.
2. Drag&drop the point to the new position.
3. Tap outside the mask to deselect the mask segment.

**To change the mask definition numerically**

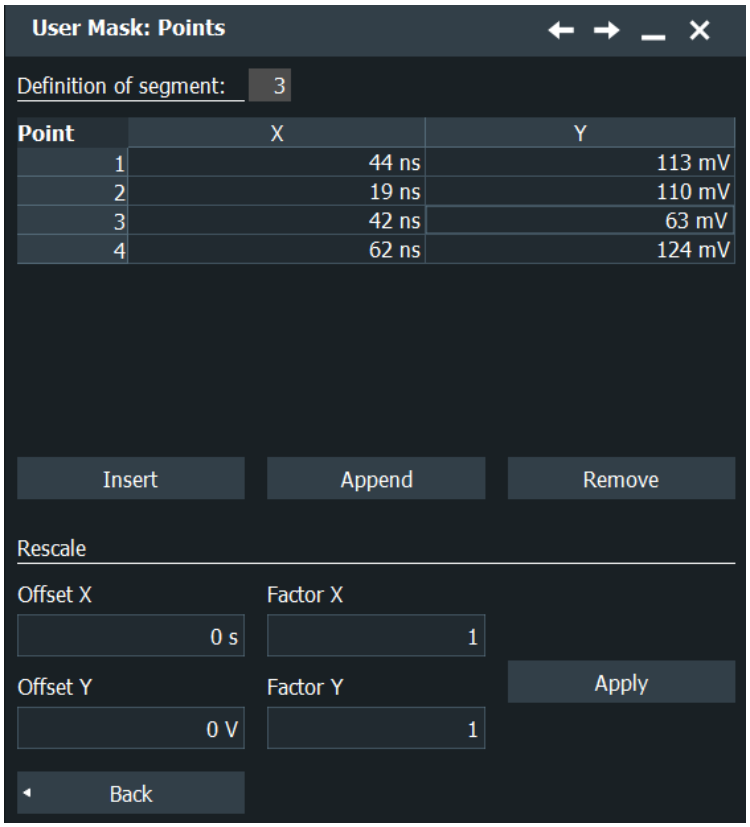
The settings mentioned here are described in detail in Chapter 10.2.2, "Mask definition", on page 427.

1. Open [Apps] > "Analysis" tab > "Mask" > "Define Mask".
2. At the top, select the mask test for which you want to change the mask definition.
3. To add or delete a mask segment, tap "Setup mask".
   Tap the segment's row in the "User Mask: Segments" table and tap the required button below:
   - "Insert": to add a new segment before the selected segment.
   - "Append": to add a new segment at the end of the list.
   - "Remove": to delete the selected mask segment from the list.




4. To add, delete, or move a point of a segment:
   a) Select the segment in the "User Mask: Segments" table.
   b) Tap "Configure".
   c) Select the point in the "Definition of segment" table.
   d) To add or delete the selected point, use the buttons below the table.
      - "Insert": to add a new point before the selected point.
      - "Append": to add a new point at the end of the list.
      - "Remove": to delete the selected point from the list.

e) To move the selected point, change the X- and Y-values.




**To rescale and move a mask segment**

The settings mentioned here are described in detail in Chapter 10.2.2, "Mask definition", on page 427.

1. Open [Apps] > "Analysis" tab > "Mask" > "Define Mask".
2. At the top, select the mask test for which you want to change the mask definition.
3. Tap "Setup mask" and select the required segment in the "User Mask: Segments" table.
4. Tap "Configure".
5. To stretch or compress the selected mask segment, enter the "Factor X" for horizontal scaling and the "Factor Y" for vertical scaling. The x-values and y-values of all points are multiplied with the corresponding factor. Factors >1 stretch the mask segment, while factors between 0 and 1 compress it. Negative values are possible and change the algebraic sign.
6. To move the selected mask segment, enter the "Offset X" for horizontal direction and the "Offset Y" for vertical direction. The specified offset is added to the corresponding values of all points.
7. Tap "Apply" to perform the scaling and/or move.

### 10.4.2 Setting up a mask test

In addition to the mask definition, the mask test contains further settings:

- The waveform to be tested
- The criteria for a failed test
- The actions to be taken if a test has failed or has been completed successfully

1. Open [Apps] > "Analysis" tab > "Mask" > "Define Mask" tab.
2. Select the "Target" to be tested.
3. Select the "Define Test" tab.
4. Set the conditions for a failed test:
   a) " Fail condition": select if sample hits or the number of acquisitions with sample hits are considered.
   b) "Violation tolerance": number of tolerable sample hits or acquisition hits.

   A test has failed if the number of sample hits or acquisition hits exceeds the limit of violation tolerance hits.
5. Tap "Action on mask".
6. For each action, select when the action will be executed:
   - "On violation" if the mask test has failed
   - "On successful completion"

### 10.4.3 Configuring the mask and hit display

The display of masks and mask violation is the same for all mask tests .

The settings mentioned here are described in detail in Chapter 10.2.3, "Define Test", on page 434.

1. Open [Apps] > "Analysis" tab > "Mask".
2. Select the "Display" tab.
3. Select "Show mask" to display the masks of all enabled mask tests on the screen.
4. Open "Menu" > "Settings" >"Appearance" dialog.
5. Select the "Mask" tab.
6. Define how the sample hits are displayed:
   a) Select "Show hits" to display the sample hits.
   b) Set the "Show for" time or "Show hits forever".
   c) Set the "Color" of the sample hits.
7. Define the color of the masks segments depending on the violation state:
   - Mask without violation
   - Mask with violation

- Mask with contact: this color shows that the edge of the mask segment was touched. In this case, the resolution is not sufficient to detect if the mask was really hit or not. Zoom into the concerned area to see the correct result.

### 10.4.4 Running a mask test

Before you can start a mask test, make sure that the mask setup is complete:

- The mask is defined, see Chapter 10.4.1.1, "Creating user masks", on page 441 and Chapter 10.4.1.2, "Modifying user masks", on page 443.
- The mask test is defined, see Chapter 10.4.2, "Setting up a mask test", on page 446
- The mask display is configured, see Chapter 10.4.3, "Configuring the mask and hit display", on page 446.

You can perform continuous testing or test a specified number of acquisitions.

1. Open [Apps] > "Analysis" tab > "Mask" > "Define Mask" tab.
2. Select "Enable".

   If the acquisition is running, the test starts immediately.
3. If the acquisition is not running, press [Run / Stop].

   The test starts and runs until you stop the acquisition or the stop action is executed if defined.

### 10.4.5 Saving and loading masks

Mask test definitions remain on the instrument until they are changed or deleted, or [Preset] is performed. If you want to keep a mask test, you can save and reload them.

**To save a mask**

1. Open [Apps] > "Analysis" tab > "Mask" > "Define Mask" tab.
2. To save the mask file in the current directory, change the file name if needed, and tap "Save".
   You can use the automatic file name generation, see Chapter 4.8.1, "Autonaming", on page 107.
3. To select the directory and enter the file name, tap "Save As".

**To load a mask**

1. Tap "Open".
2. Select the file from the file selection dialog box.

### 10.4.6 Mask testing on history acquisitions

In the same way as for running acquisitions, you can set up and perform the mask testing also on history waveforms.

The requirements for mask testing on history waveforms are also the same:

- The mask is defined, see Chapter 10.4.1.1, "Creating user masks", on page 441 and Chapter 10.4.1.2, "Modifying user masks", on page 443.
- The mask test is defined, see Chapter 10.4.2, "Setting up a mask test", on page 446
- The mask display is configured, see Chapter 10.4.3, "Configuring the mask and hit display", on page 446.

1. Perform and finish the acquisition.
2. Press [History].
3. In the quick-access "History" dialog box, tap "Play".

   The mask testing is performed on the complete history memory, starting with the oldest acquisition. The state of the mask test is set to "Finished" when "Nx Single count" acquisitions are tested.

For details on history, see Chapter 7.4.1, "About history", on page 298.

# 11 Search functions

Search functions allow you to detect and analyze specific events in the acquired data quickly and simply. You can search in various waveforms for several events at once. The search area can be limited by a gate.

The events that can be searched for are basically the same events you can trigger on. Thus, the search parameters are defined in the same way as the trigger conditions. The results are displayed in a result box and optionally shown in a zoom window.

## 11.1 Overview: search definition and results

### 11.1.1 Search definition

You can define up to 8 different searches and let them run simultaneously. For each search, you define the criteria, the parameters of each criterion, the gate, and the result display.

The instrument keeps the settings until the next preset. If you save a user-defined preset, the search settings are included in the preset.

Each search is configured in a separate tab and contains:

- *Search control*
  If you enable a search and run an acquisition, the search is performed continuously on the acquired data until acquisition is stopped.
  If acquisition is stopped and you enable a search, the data of the last acquisition is searched.
  Enabling the search zoom window disables the search, stops a running acquisition, and displays the search results of the last acquisition in the zoom window.
- *Source*
  Waveform that is searched for one or more events. You can search in analog and digital signals, math or reference waveforms, and tracks. Furthermore, search in decoded data of serial buses is possible.
  Search on differential signals and their input channels is not supported.
- *Search criteria and parameters*
  Various search criteria are available, depending on the source. Most parameters known from trigger event definition can also be configured as search conditions. Unlike triggering, you can configure several event types to be searched for simultaneously.
  If the source is an FFT spectrum, you can perform a frequency marker search by using the cursor measurement and defining the peak excursion. See Chapter 8.1.3.3, "Peak Search tab", on page 317.
- *Search gate*
  Searches can be performed on the entire waveform, or only on a defined area (gate). The gate can be coupled to an existing zoom.

Gating is not available for searches on digital signals and serial buses.

- *Result presentation*
  For each search, you define how the search results are displayed: in a result table and/or in a search zoom window.

Remote commands:

- `SEARch:ADD` on page 1566
- `SEARch:REMove` on page 1566

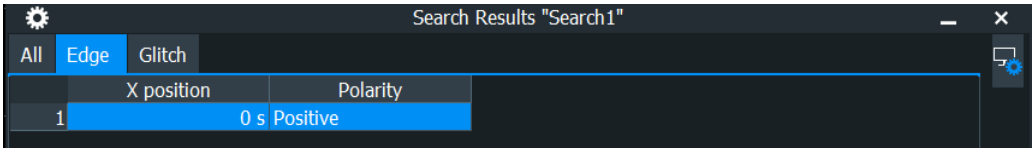
### 11.1.2 Search results

The results are displayed in a "Search Results" box and optionally in a zoom window.

**Search results box**

The results of each search are tabulated in a "Search Results" box.

If you search for several event types in parallel, the results are presented in several tabs - one for each search event and one for the combined results. Each tab contains a table with the position and, if available, further parameters for each result. The tables row can be sorted, and you can define a maximum number of table entries in the "Result Presentation" dialog box. As with all result boxes, you can minimize it, display it like a diagram, and define the default position.




If "Auto clear" is enabled in the "Display" dialog box, the instrument displays the search results of the last acquisition. If "Auto clear" is disabled, the first result of each acquisition is listed until the maximum number of entries in the table is reached.

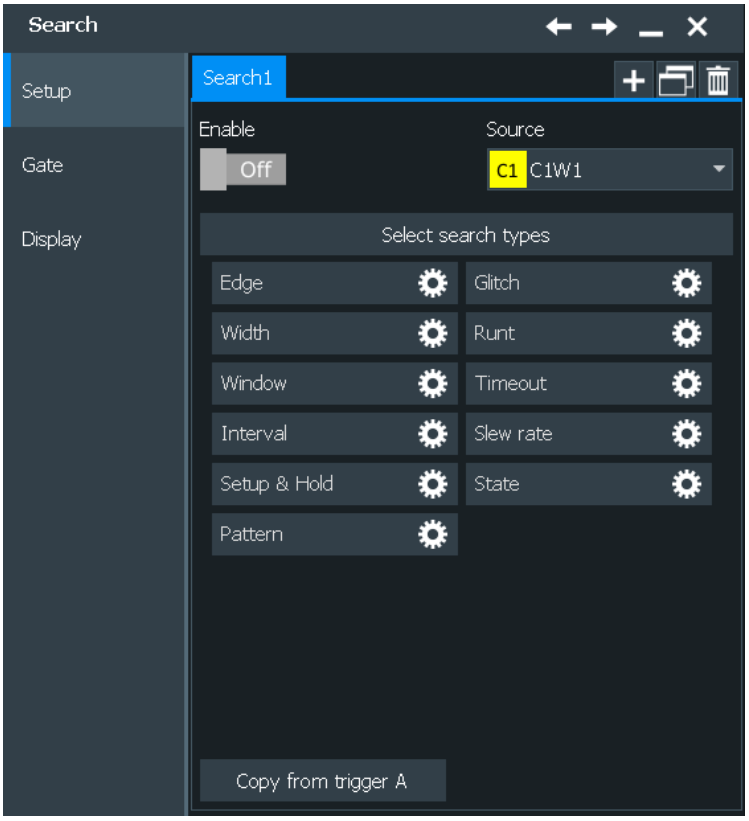
Remote commands for result query:

- `SEARch:RESult[:ALL]?` on page 1603

## 11.2 Search setup

Access: "Menu" > "Apps" > "Analysis" tab > "Search"

The search setup includes the source selection, the selection of search events (criteria), event-specific search conditions, and search control.




### 11.2.1 Search criteria

Access: "Menu" > "Apps" > "Analysis" tab > "Search" > "Setup" tab

**Enable**
If you enable a search and run an acquisition, the search is performed continuously on the acquired data until acquisition is stopped.

If acquisition is stopped and you enable a search, the data of the last acquisition is searched.

Remote command:
`SEARch:ONLine` on page 1567
`SEARch:ALL` on page 1567

**Source**
Defines the waveform to be searched. The source can be any analog and digital input signal, math or reference waveform, or track. While the instrument triggers only on real input signals, it can search also calculated and restored waveforms.

If the source is an FFT spectrum, you can perform a frequency marker search by using the cursor measurement and defining the peak excursion. See Chapter 8.1.3.3, "Peak Search tab", on page 317.

For some serial protocol options, search on a serial bus is available. For details, see the relevant chapters of the "Protocol Analysis" chapter.

Depending on the selected source, different search criteria are available.

Remote command:
SEARch:SOURce on page 1566

**Edge, Glitch, Width, Runt, Window, Timeout, Interval, Slew rate, Setup & Hold, State, Pattern**
Search criteria for analog and digital input signals, math and reference waveforms, and tracks. For searching on digital channels, only "Edge", "Width", "Timeout" and "Setup & Hold" criteria are available.

Tap the icon to include or exclude the search criteria in the next search. You can enable several event types for simultaneous search.

Remote command:
SEARch:TRIGger:EDGE[:STATe] on page 1568
SEARch:TRIGger:GLITch[:STATe] on page 1568
SEARch:TRIGger:WIDTh[:STATe] on page 1569
SEARch:TRIGger:RUNT[:STATe] on page 1568
SEARch:TRIGger:WINDow[:STATe] on page 1569
SEARch:TRIGger:TIMeout[:STATe] on page 1568
SEARch:TRIGger:INTerval[:STATe] on page 1568
SEARch:TRIGger:SLEWrate[:STATe] on page 1568
SEARch:TRIGger:SETHold[:STATe] on page 1568
SEARch:TRIGger:STATe[:STATe] on page 1568
SEARch:TRIGger:PATTern[:STATe] on page 1568

**Copy from trigger A'**
Copies the trigger type-specific settings from the A-trigger configuration to the search settings. The source itself is not copied.

Remote command:
SEARch:TRIGger:EDGE:ACOPy on page 1569
SEARch:TRIGger:GLITch:ACOPy on page 1569
SEARch:TRIGger:WINDow:ACOPy on page 1570
SEARch:TRIGger:WIDTh:ACOPy on page 1570
SEARch:TRIGger:RUNT:ACOPy on page 1569
SEARch:TRIGger:WINDow:ACOPy on page 1570
SEARch:TRIGger:TIMeout:ACOPy on page 1569
SEARch:TRIGger:INTerval:ACOPy on page 1569
SEARch:TRIGger:SLEWrate:ACOPy on page 1569
SEARch:TRIGger:SETHold:ACOPy on page 1569
SEARch:TRIGger:STATe:ACOPy on page 1569
SEARch:TRIGger:PATTern:ACOPy on page 1569

### 11.2.2 Search parameters

Most parameters available for trigger event definition can also be configured as search conditions. Each event type is defined in a separate subtab.

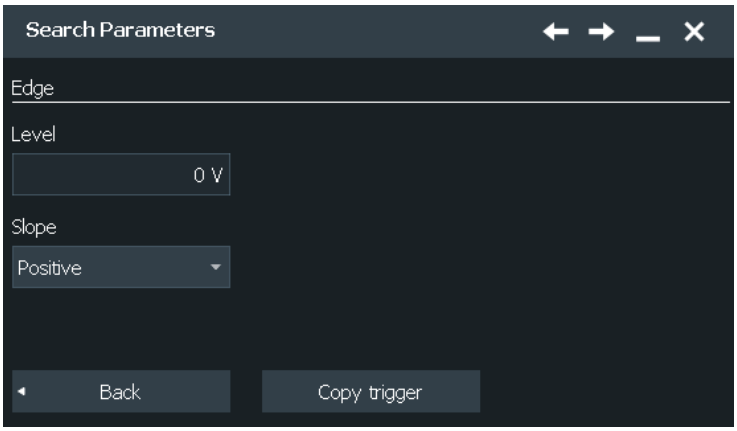
If the source is a spectrum, the instrument performs a frequency marker search.

For serial protocol options, search on a serial bus is available. These searches have protocol-specific search criteria. For details, see the relevant chapters of the "Protocol Analysis" chapter.


- Edge....453
- Glitch....454
- Width....454
- Runt....455
- Window....456
- Timeout....457
- Interval....458
- Slew rate....459
- Setup & Hold....460
- State....462
- Pattern....462


### 11.2.2.1 Edge

The edge search works the same way as the edge trigger.




**Slope**
Sets the edge type: rising edge ("Positive"), falling edge ("Negative"), or both.

Remote command:
`SEARch:TRIGger:EDGE:SLOPe` on page 1570

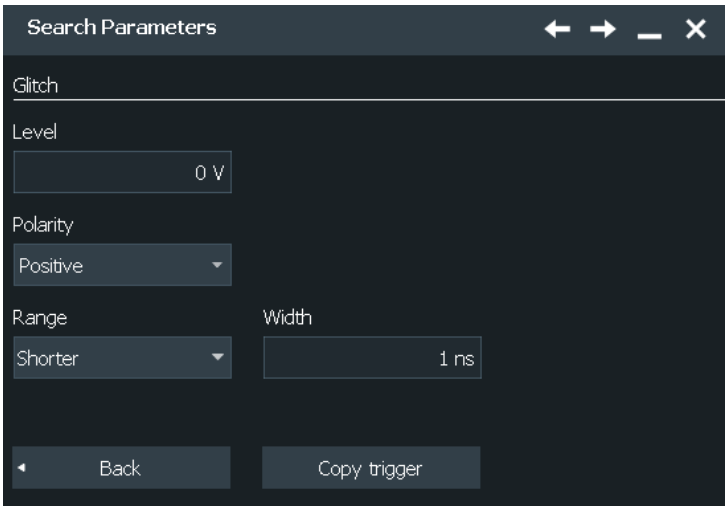
**Level**
Sets the voltage level for trigger search. The value is used for all search event types that require one trigger level. The search trigger level is search-specific, you can define different levels in different searches for the same event.

Remote command:
`SEARch:TRIGger:LEVel[:VALue]` on page 1569

### 11.2.2.2 Glitch

The glitch search works the same way as the glitch trigger. To apply the trigger settings to search, tap Copy from trigger A'.

The glitch search is not available if the search source is a digital channel.




**Polarity, Range, Width**
See trigger settings:
- "Range" on page 200
- "Width" on page 200
- "Polarity" on page 200

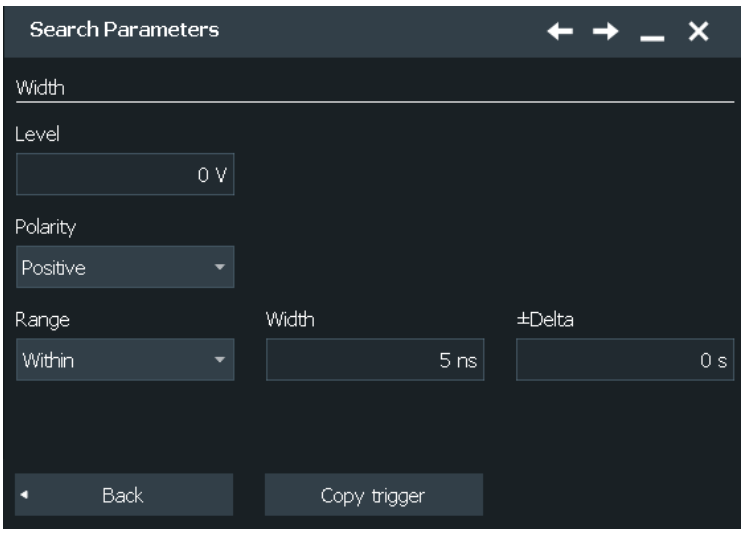
Remote command:
`SEARch:TRIGger:GLITch:POLarity` on page 1571
`SEARch:TRIGger:GLITch:RANGe` on page 1571
`SEARch:TRIGger:GLITch:WIDTh` on page 1571

**Level**
Sets the voltage level for trigger search. The value is used for all search event types that require one trigger level. The search trigger level is search-specific, you can define different levels in different searches for the same event.

Remote command:
`SEARch:TRIGger:LEVel[:VALue]` on page 1569

### 11.2.2.3 Width

The width search works the same way as the width trigger. To apply the trigger settings to search, tap Copy from trigger A'.




**Polarity, Range, Width, ±Delta**
See trigger settings:

- "Polarity" on page 201
  While the width trigger can only analyze positive *or* negative polarity, searching for a width is also possible for both polarities at the same time ("Either").
- "Range" on page 202
- "Width" on page 202
- "±Delta" on page 202

Remote command:
`SEARch:TRIGger:WIDTh:POLarity` on page 1580
`SEARch:TRIGger:WIDTh:RANGe` on page 1580
`SEARch:TRIGger:WIDTh:WIDTh` on page 1581
`SEARch:TRIGger:WIDTh:DELTa` on page 1580

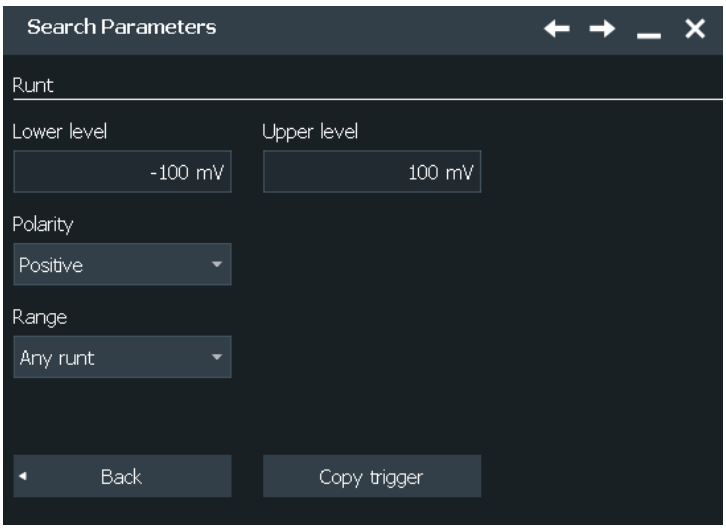
**Level**
Sets the voltage level for trigger search. The value is used for all search event types that require one trigger level. The search trigger level is search-specific, you can define different levels in different searches for the same event.

Remote command:
`SEARch:TRIGger:LEVel[:VALue]` on page 1569

### 11.2.2.4 Runt

The runt search settings are the same as the runt trigger settings. To apply the trigger settings to search, tap Copy from trigger A'.

The runt search is not available, if the search source is a digital channel.




**Polarity, Range, Runt width, ±Delta**
Time limit for the runt, see trigger settings:

- "Polarity" on page 200
- "Range" on page 203
- "Runt width" on page 204
- "±Delta" on page 204

Remote command:
`SEARch:TRIGger:RUNT:POLarity` on page 1574
`SEARch:TRIGger:RUNT:RANGe` on page 1574
`SEARch:TRIGger:RUNT:WIDTh` on page 1575
`SEARch:TRIGger:RUNT:DELTa` on page 1574

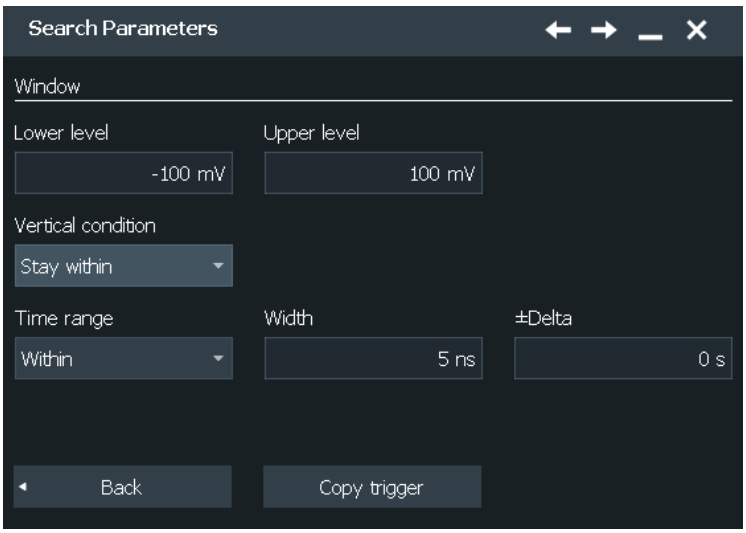
**Upper level, Lower level**
Set the upper and lower voltage thresholds. The amplitude of a runt crosses the first threshold twice in succession without crossing the second one.

Remote command:
`SEARch:TRIGger:LEVel:RUNT:UPPer` on page 1575
`SEARch:TRIGger:LEVel:RUNT:LOWer` on page 1575

### 11.2.2.5 Window

The window search settings are the same as the window trigger settings. This search type is not available if the search source is a digital channel. To apply the trigger settings to search, tap Copy from trigger A'.

The window search is not available if the search source is a digital channel.




**Vertical condition**
Defines the run of the signal relative to the window, see "Vertical condition" on page 205.

Remote command:
SEARch:TRIGger:WINDow:RANGe on page 1582

**Time condition, Width, ±Delta**
Set the time limit for the vertical condition, see:
- "Time condition" on page 205
- "Width" on page 206
- "±Delta" on page 206

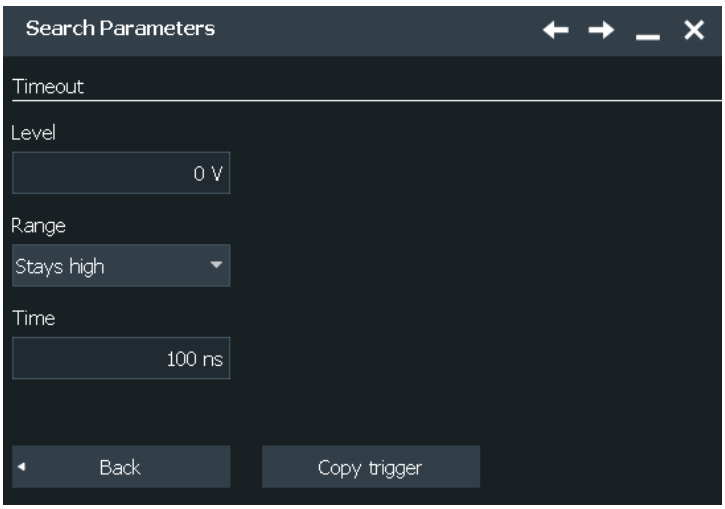
Remote command:
SEARch:TRIGger:WINDow:TIMerange on page 1583
SEARch:TRIGger:WINDow:WIDTh on page 1583
SEARch:TRIGger:WINDow:DELTa on page 1582

**Upper level, Lower level**
Set the upper and lower voltage thresholds. The amplitude of a runt crosses the first threshold twice in succession without crossing the second one.

Remote command:
SEARch:TRIGger:LEVel:WINDow:UPPer on page 1584
SEARch:TRIGger:LEVel:WINDow:LOWer on page 1584

### 11.2.2.6 Timeout

The timeout search settings are the same as the timeout trigger settings. To apply the trigger settings to search, tap Copy from trigger A'.




**Range, Time**
Set the timeout condition, see:

- "Range" on page 207
- "Time" on page 207

Remote command:
SEARch:TRIGger:TIMeout:RANGe on page 1579
SEARch:TRIGger:TIMeout:TIME on page 1579

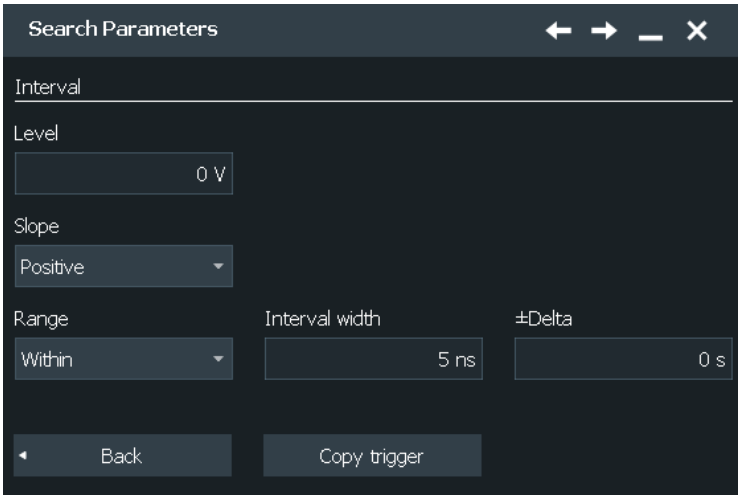
**Level**
Sets the voltage level for trigger search. The value is used for all search event types that require one trigger level. The search trigger level is search-specific, you can define different levels in different searches for the same event.

Remote command:
SEARch:TRIGger:LEVel[:VALue] on page 1569

### 11.2.2.7 Interval

The interval search settings are the same as the interval trigger settings. To apply the trigger settings to search, tap Copy from trigger A'.

The interval search is not available if the search source is a digital channel.




**Slope, Range, Interv. width, ±Delta**
Set the interval condition, see:

- "Slope" on page 208
  While the interval trigger can only analyze rising or falling edges, searching for a width is possible for both edges at the same time ("Either").
- "Range" on page 208
- "Interv. width" on page 208
- "±Delta" on page 208

Remote command:
`SEARch:TRIGger:INTerval:SLOPe` on page 1572
`SEARch:TRIGger:INTerval:RANGe` on page 1573
`SEARch:TRIGger:INTerval:WIDTh` on page 1573
`SEARch:TRIGger:INTerval:DELTa` on page 1572

**Level**
Sets the voltage level for trigger search. The value is used for all search event types that require one trigger level. The search trigger level is search-specific, you can define different levels in different searches for the same event.

Remote command:
`SEARch:TRIGger:LEVel[:VALue]` on page 1569

### 11.2.2.8 Slew rate

The slew rate search settings are the same as the slew rate trigger settings. To apply the trigger settings to search, tap Copy from trigger A'.

The slew rate search is not available if the search source is a digital channel.

**Polarity, Range, Slew rate, ±Delta**

- "Slope" on page 199
- "Range" on page 209
- "Slew rate" on page 210
- "±Delta" on page 210

Remote command:

`SEARch:TRIGger:SLEWrate:SLOPe` on page 1577
`SEARch:TRIGger:SLEWrate:RANGe` on page 1577
`SEARch:TRIGger:SLEWrate:TIME` on page 1577
`SEARch:TRIGger:SLEWrate:DELTa` on page 1576

**Upper level, Lower level**

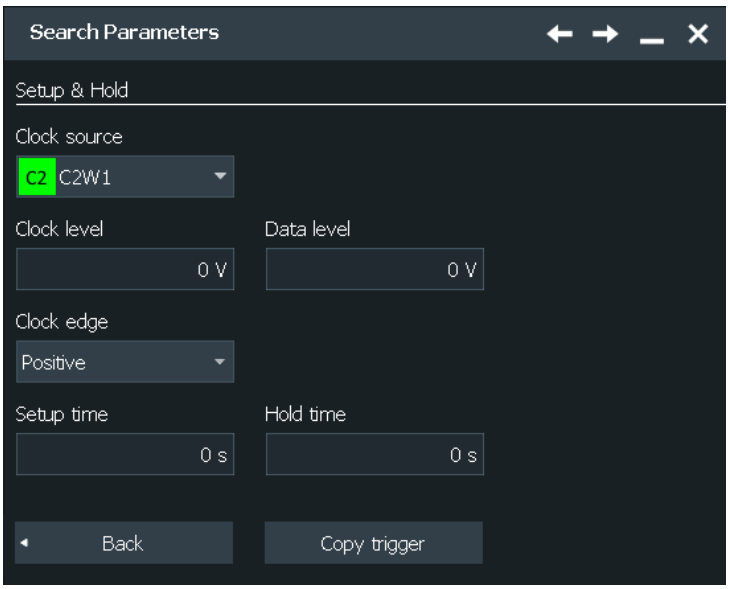
Set the upper and lower voltage thresholds. When the signal crosses a level, the slew rate measurement starts or stops depending on the selected slope.

Remote command:

`SEARch:TRIGger:LEVel:TRANsition:UPPer` on page 1578
`SEARch:TRIGger:LEVel:TRANsition:LOWer` on page 1578

### 11.2.2.9 Setup & Hold

The Setup & Hold search settings are the same as the Setup & Hold trigger settings. To apply the trigger settings to search, tap Copy from trigger A'.




**Clock source, Clock edge, Clock level**
Set the clock settings. Both "Clock level" and "Clock edge" define the starting point for calculation of the setup and hold time.

Remote command:
`SEARch:TRIGger:SETHold:CSOurce` on page 1585
`SEARch:TRIGger:SETHold:CEDGe` on page 1584
`SEARch:TRIGger:SETHold:CLEVel` on page 1585

**Data level**
Sets the voltage level for the data signal. At this level, the setup and hold time are measured.

Remote command:
`SEARch:TRIGger:LEVel[:VALue]` on page 1569

**Level**
Sets the voltage level for trigger search. The value is used for all search event types that require one trigger level. The search trigger level is search-specific, you can define different levels in different searches for the same event.

Remote command:
`SEARch:TRIGger:LEVel[:VALue]` on page 1569

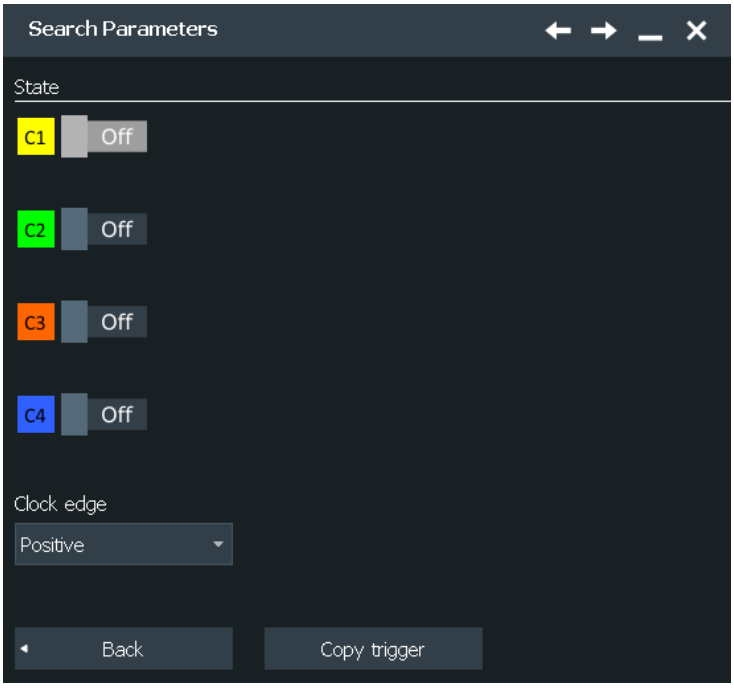
**Setup time, Hold time**
Sets the minimum time **before** (Setup) and **after** (Hold) the clock edge while the data signal must stay steady above or below the data level.

See also: "Setup time" on page 212 and "Hold time" on page 212.

Remote command:
`SEARch:TRIGger:SETHold:STIMe` on page 1586
`SEARch:TRIGger:SETHold:HTIMe` on page 1585

### 11.2.2.10 State

The state search is a qualified edge search. The state search is only available for analog channel sources (Ch1 to Ch4).




**Clock source, Clock edge**
Define the clock settings. The clock signal is the waveform to be searched.

Remote command:
`SEARch:TRIGger:STATe:CSOurce` on page 1591
`SEARch:TRIGger:STATe:CEDGe` on page 1592
`SEARch:TRIGger:STATe:CLEVel` on page 1592

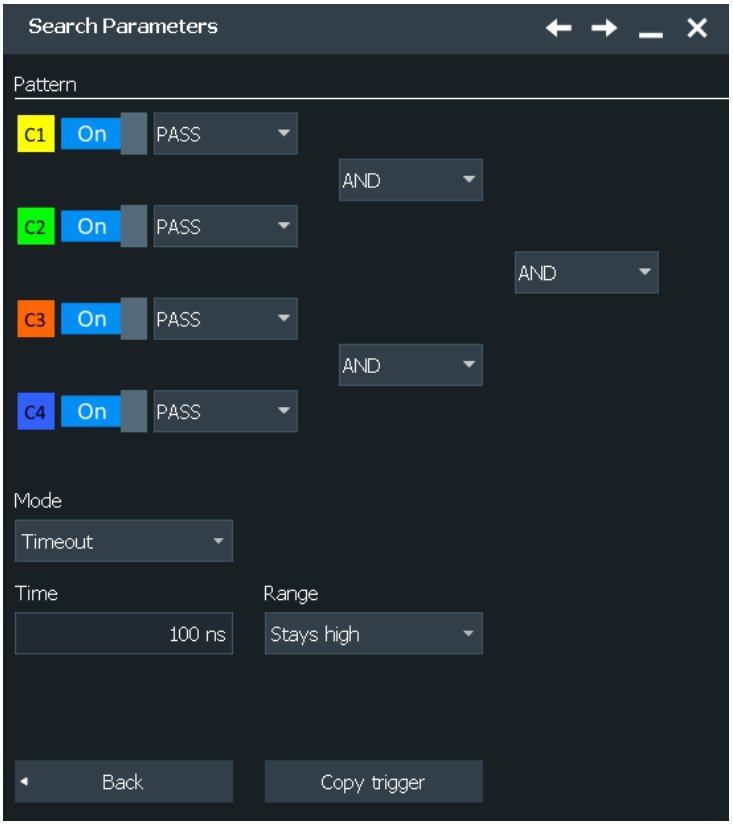
**State pattern**
State settings are the same as for the state trigger.

See also "Pattern" on page 214.

Remote command:
`SEARch:TRIGger:STATe:A[:ENABle]` on page 1592
`SEARch:TRIGger:STATe:A:LOGic` on page 1593
`SEARch:TRIGger:STATe:AB:LOGic` on page 1593
`SEARch:TRIGger:STATe:ABCD:LOGic` on page 1593

### 11.2.2.11 Pattern

The pattern search combines a logical combination of the input channels with a timing condition. The pattern search is only available for analog channel sources (Ch1 to Ch4).




**Pattern**
Pattern search settings are the same as for the pattern trigger.

See also "Pattern" on page 214.

Remote command:
`SEARch:TRIGger:PATTern:A[:ENABle]` on page 1587
`SEARch:TRIGger:PATTern:A:LOGic` on page 1587
`SEARch:TRIGger:PATTern:AB:LOGic` on page 1588
`SEARch:TRIGger:PATTern:ABCD:LOGic` on page 1588

**Timing condition: Mode, Range, Time, Width, ±Delta**
Additional time limitation to the pattern, see "Additional settings: Timing" on page 215.

Remote command:
`SEARch:TRIGger:PATTern:MODE` on page 1588
`SEARch:TRIGger:PATTern:TIMeout:MODE` on page 1589
`SEARch:TRIGger:PATTern:TIMeout[:TIME]` on page 1589
`SEARch:TRIGger:PATTern:WIDTh:RANGe` on page 1590
`SEARch:TRIGger:PATTern:WIDTh[:WIDTh]` on page 1590
`SEARch:TRIGger:PATTern:WIDTh:DELTa` on page 1591

### 11.2.3 Frequency marker search

When you start a search on a spectrum, a frequency marker search is performed to detect peaks in a spectrum. You can define which peaks the instrument finds by defining the noise reject settings.

**Threshold**
See "Threshold" on page 318.

**Peak excursion**
See "Peak excursion" on page 319.

### 11.2.4 Configuring the search setup

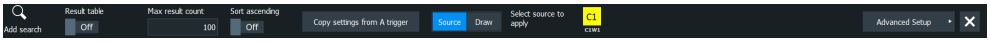
There are several ways to create a search:

- Creating a simple default search using the toolbar icon. This method is not available for search on serial buses.
- Setting up a search using the dialog box.

**To perform a simple search**

1. If more than one waveform is in the diagram, select the waveform to be searched for by tapping it in the diagram.
2. Select the "Search" icon on the toolbar.

   The overlay menu opens.

3. Tap the diagram with the waveform to be searched, or drag a rectangle on the diagram to define the search area.
4. Alternatively, use the settings provided in the overlay menu.

   The default edge search is configured as "Search<x>" and performed. The "Search Results" box is displayed.

**To create a user-defined search**

1. Open the search dialog: "Menu" > "Apps" > "Analysis" tab > "Search".
2. There are two ways to create a search:
   - If you want to create a new, unconfigured search, tap the "Add" icon.

   - If you want to create a new search based on an existing one, tap the "Copy" icon.

3. Enter a name for the search using the on-screen keyboard.

**To configure a user-defined search**

1. Open the search dialog: "Menu" > "Apps" > "Analysis" tab > "Search".
2. Select the "Setup" tab and the search that you want to configure.
3. Select the "Source" on which you want to perform the search.
4. Select the events that you want to include in the search.
5. Define the settings of the first search event.
   To use the same conditions as defined in the trigger configuration of the A-event, tap "Copy from trigger A'". The selected trigger settings are applied to the search settings.
6. Repeat the previous steps to define further events for the same search.
7. To perform the search only on a part of the waveform, configure the gate in the "Gate" tab as described in Chapter 11.3.2, "Defining the search gate", on page 467.

## 11.3 Search gate

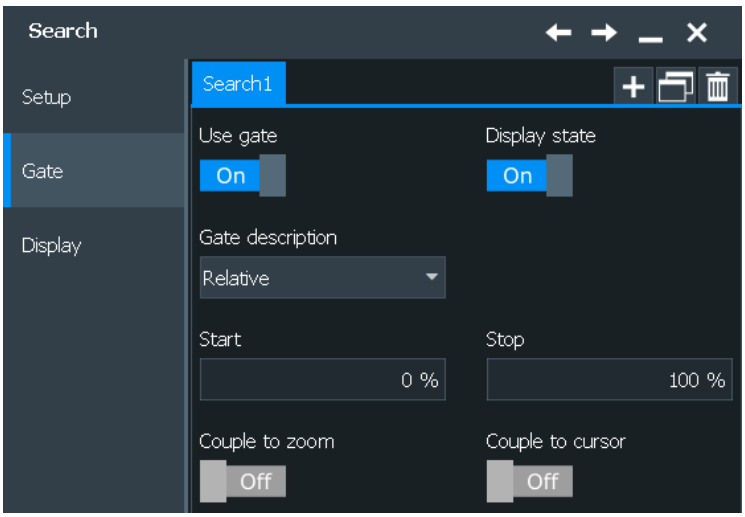
The gate defines the search area within the source waveform. You can use absolute or relative values to define the gate, or couple it to a previously defined zoom area.

### 11.3.1 Gate settings

Access: "Menu" > "Apps" > "Analysis" tab > "Search" > "Gate" tab

The search gate settings are identical to those for gate areas for measurements or FFT analysis.

Gating is not available if the search source is a digital channel or a serial bus.




**Use gate**
Enables the gate settings and shows the gate. Search is only performed on the defined gate area of the source waveform.

Remote command:
`SEARch:GATE[:STATe]` on page 1595

**Display state**
Displays the gate area in the source diagram.

Remote command:
`SEARch:GATE:SHOW` on page 1595

**Gate description**
Defines whether the gate settings are configured using absolute or relative values.

"Absolute" The gate is defined by absolute start and stop values.

"Relative" The gate's start and stop values are defined by a percentage of the value range.

Remote command:
`CALCulate:MATH<m>:FFT:GATE:MODE` on page 1533
`MEASurement<m>:GATE:MODE` on page 1503
`SEARch:GATE:MODE` on page 1595

**(Relative) Start**
Defines the starting value for the gate.

Remote command:
`CALCulate:MATH<m>:FFT:GATE:ABSolute:STARt` on page 1532
`CALCulate:MATH<m>:FFT:GATE:RELative:STARt` on page 1533
`MEASurement<m>:GATE:ABSolute:STARt` on page 1504
`MEASurement<m>:GATE:RELative:STARt` on page 1504
`SEARch:GATE:ABSolute:STARt` on page 1596
`SEARch:GATE:RELative:STARt` on page 1596

**(Relative) Stop**
Defines the end value for the gate.

Remote command:
CALCulate:MATH<m>:FFT:GATE:ABSolute:STOP on page 1532
CALCulate:MATH<m>:FFT:GATE:RELative:STOP on page 1533
MEASurement<m>:GATE:ABSolute:STOP on page 1504
MEASurement<m>:GATE:RELative:STOP on page 1504
SEARch:GATE:ABSolute:STOP on page 1596
SEARch:GATE:RELative:STOP on page 1597

**Zoom**
Zoom coupling is available if a zoom is defined. As long as "Zoom coupling" is enabled, the gate area is defined identically to the zoom area - if you change the zoom, the gate changes as well.

If several zoom diagrams are defined, select the zoom diagram to be used for gating. The "Start" and "Stop" values of the gate are adjusted accordingly.

Zoom coupling can be set for measurement gates, FFT gates, and search gates. The zoom must be defined on the diagram that contains the signal source of the measurement, FFT, or search.

Remote command:
MEASurement<m>:GATE:ZCOupling on page 1505
MEASurement<m>:GATE:ZDIagram on page 1506
CALCulate:MATH<m>:FFT:GATE:ZCOupling on page 1534
SEARch:GATE:ZCOupling on page 1597
SEARch:GATE:ZDIagram on page 1597

**Cursor**
If enabled, the gate area is defined by the cursor lines of an active cursor measurement.

If several cursor measurements are enabled, select the cursor set to be used for gating. The "Start" and "Stop" values of the gate are adjusted to the values of the cursor line positions. The measurement is limited to the part of the waveform between the cursor lines.

Remote command:
MEASurement<m>:GATE:CCOupling on page 1505
MEASurement<m>:GATE:CURSor on page 1504

### 11.3.2 Defining the search gate

If you create a search using the "Search" toolbar icon, you can directly define the gate by dragging a rectangle on the diagram. Otherwise, you define the gate in the "Gate" tab of the "Search" dialog box.

1. Open the "Menu" > "Apps" > "Analysis" tab > "Search" > "Gate" tab.
2. Select the search for which you want to define the gate.

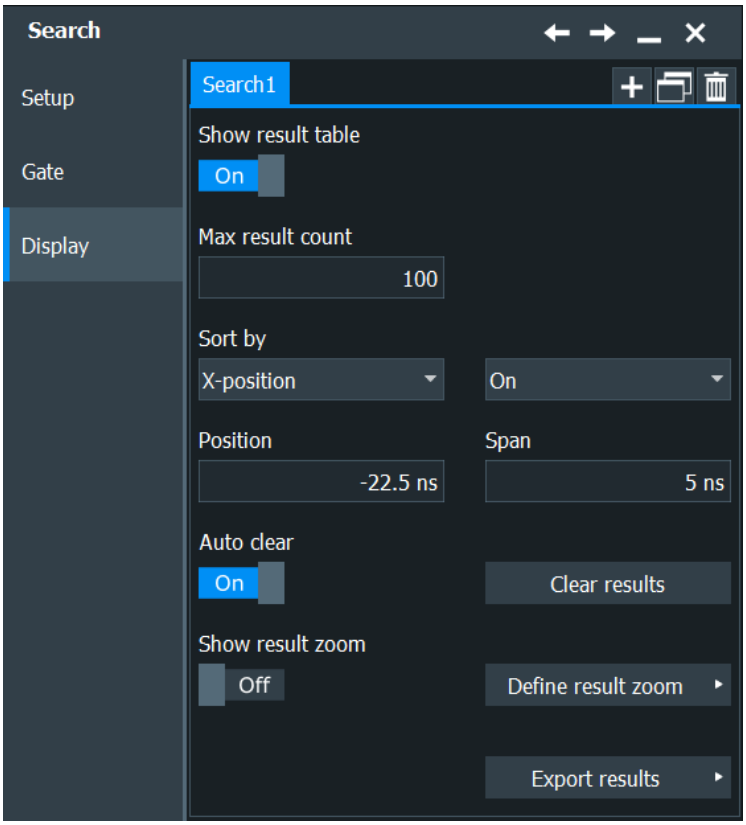
3. Use one of the following methods:
   a) Set the absolute or relative "Gate description" . Enter the start and stop values of the gate area.
   b) If a zoom area has already been defined for the waveform, couple the gate area to the zoom area by selecting the "Couple to zoom" option. If several zoom diagrams are defined, select the zoom diagram you want to use for gating.
4. Tap "Use gate" to enable the gate.
5. Optionally, tap "Display state" to display the gate area in the diagram.

## 11.4 Search display presentation

Search results are displayed in a table in the "Search Results" box. In addition, a zoom window for a selected search result can be displayed so that you can analyze the result in more detail.

### 11.4.1 Search display settings

Access: "Menu" > "Apps" > "Analysis" tab > "Search" > "Display"



The following settings configure the layout of the result table in the "Search Results" box and the size and position of the search zoom window. The result tables can be sorted by x-position or value. You can define a maximum number of table entries.

**Show result table**
Displays or hides the search result table.

Remote command:
`SEARch:RESult:SHOW` on page 1602

**Max result count**
Defines the maximum number of entries in the search result table.

Remote command:
`SEARch:RESult:LIMit` on page 1602

**Sort by**
Sorts the search results by x-value position or value of the result.

Remote command:
`SEARch:RESult:SORT[:MODE]` on page 1603

**Sort ascending**
By default, the results are listed in descending order, i.e. the largest value at the top. To change the sorting direction, enable "Sort ascending".

Remote command:
`SEARch:RESult:SORT:ASCending` on page 1603

**Clear results**
Clears the search results once and starts a new search.

Remote command:
`SEARch:CLEar` on page 1566

**Auto clear**
If "Auto clear" is enabled, the instrument displays the search results of the last acquisition.

If "Auto clear" is disabled, the first result of each acquisition is listed until the maximum number of entries in the table is reached.

**Show result zoom**
If enabled, a zoom window is displayed for the currently selected search result. The zoom area is indicated in the diagram that displays the source waveform of the search.

Remote command:
`SEARch:RESDiagram:SHOW` on page 1600

**Define result zoom**
Opens a dialog to define the reult zoom display settings.

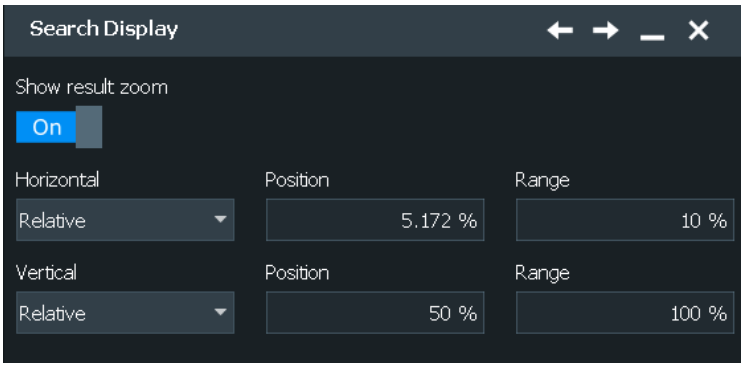
See Chapter 11.4.2, "Display zoom settings", on page 470.

**Export results**
Opens a dialog to define the export reults.

See Chapter 12.2.4, "Results", on page 499.

## 11.4.2 Display zoom settings

You can display a zoom window for the currently selected search result. The zoom area is indicated in the diagram that displays the source waveform of the search.




**Show result zoom**
If enabled, a zoom window is displayed for the currently selected search result. The zoom area is indicated in the diagram that displays the source waveform of the search.

Remote command:
`SEARch:RESDiagram:SHOW` on page 1600

**Y mode**
Defines whether absolute or relative values are used to specify the y-axis values.

Remote command:
`LAYout:ZOOM:VERTical:MODE` on page 1384
`SEARch:RESDiagram:VERT:MODE` on page 1601

**Y position**
Defines the y-value of the centerpoint of the zoom area.

Remote command:
`LAYout:ZOOM:VERTical:ABSolute:POSition` on page 1384
`LAYout:ZOOM:VERTical:RELative:POSition` on page 1386
`SEARch:RESDiagram:VERT:ABSolute:POSition` on page 1600
`SEARch:RESDiagram:VERT:RELative:POSition` on page 1601

**Y range**
Defines the height of the zoom area.

Remote command:
`LAYout:ZOOM:VERTical:RELative:SPAN` on page 1386
`LAYout:ZOOM:VERTical:ABSolute:SPAN` on page 1385

SEARch:RESDiagram:VERT:ABSolute:SPAN on page 1601
SEARch:RESDiagram:VERT:RELative:SPAN on page 1602

**X mode**
Defines whether absolute or relative values are used to specify the x-axis values.

Remote command:
LAYout:ZOOM:HORZ:MODE on page 1380
SEARch:RESDiagram:HORZ:MODE on page 1599

**X position**
Defines the x-value of the centerpoint of the zoom area.

Remote command:
LAYout:ZOOM:HORZ:ABSolute:POSition on page 1381
LAYout:ZOOM:HORZ:RELative:POSition on page 1382
SEARch:RESDiagram:HORZ:ABSolute:POSition on page 1598
SEARch:RESDiagram:HORZ:RELative:POSition on page 1599

**X range**
Defines the width of the zoom area.

Remote command:
LAYout:ZOOM:HORZ:ABSolute:SPAN on page 1381
LAYout:ZOOM:HORZ:RELative:SPAN on page 1383
SEARch:RESDiagram:HORZ:ABSolute:SPAN on page 1599
SEARch:RESDiagram:HORZ:RELative:SPAN on page 1600

### 11.4.3 Configuring the search results presentation

Initially, the "Search Results" box is displayed in front of the other diagrams or as result icon on the signal bar. This depends on the default setting in the "Diagram Layout" tab. Alternatively, you can display it in its own area on the screen, like any other diagram.

For details, see Chapter 3.3.7, "Displaying results", on page 61.

**To configure the result tables**

1. Open the search dialog: "Menu" > "Apps" > "Analysis" tab > "Search".
2. Select the tab for the search you want to configure.
3. Select the "Display" tab.
4. Select "Show result table" to display the "Search Results" box.
5. Select the sort mode of the result table.
6. By default, the results are listed in descending order, i.e. the largest value at the top. To change the sorting direction, enable "Sort ascending".
7. Define a maximum number of results to be displayed in the result table in the "Max. result count" field.

**To display search zoom windows**

1. In the "Search > Display" tab, enable "Show result zoom".

   This stops a running search and a running acquisition.
   The zoom area is indicated in the diagram that displays the source waveform of the search. The zoom window is displayed for the first result that was found.

2. If you need to adjust the search zoom area, you can drag the area or their edges on the screen. You can also enter the limits of the search zoom window in the "Search > Display > Define result zoom".
   Be aware, that the zoom window size is valid for all results of a search definition. If you change the settings drastically for one result, they may not be correct for the next search result you switch to.

   See also:
   - Chapter 7.1.3, "Zooming for details", on page 267

# 12 Data and file management

This chapter describes how to manage instrument settings, and measurement results like waveform data, numeric results and screenshots.

The "Save/Recall" dialog provides functions for saving and restoring data on the instrument. A naming pattern is available and can be adjusted to simplify a clear data storage.

The effect of the [Camera] key can be configured to save screenshots or reports.


- Save and recall user settings....473
- Save and recall waveform data and results....482
- Screenshots....507
- Reports....511
- File selection dialog....513


## 12.1 Save and recall user settings

To repeat measurements at different times or perform similar measurements with different test data, it is useful to save the used instrument settings and load them again later. Furthermore, it can be helpful to refer to the instrument settings of a particular measurement when analyzing the results. Therefore, you can easily save the instrument settings of a measurement. In addition to the measurement-related settings, user-specific display settings and active reference waveforms can also be saved and loaded.

Access: "Menu" key > "Save/Recall" key > "Save" tab > "User settings"

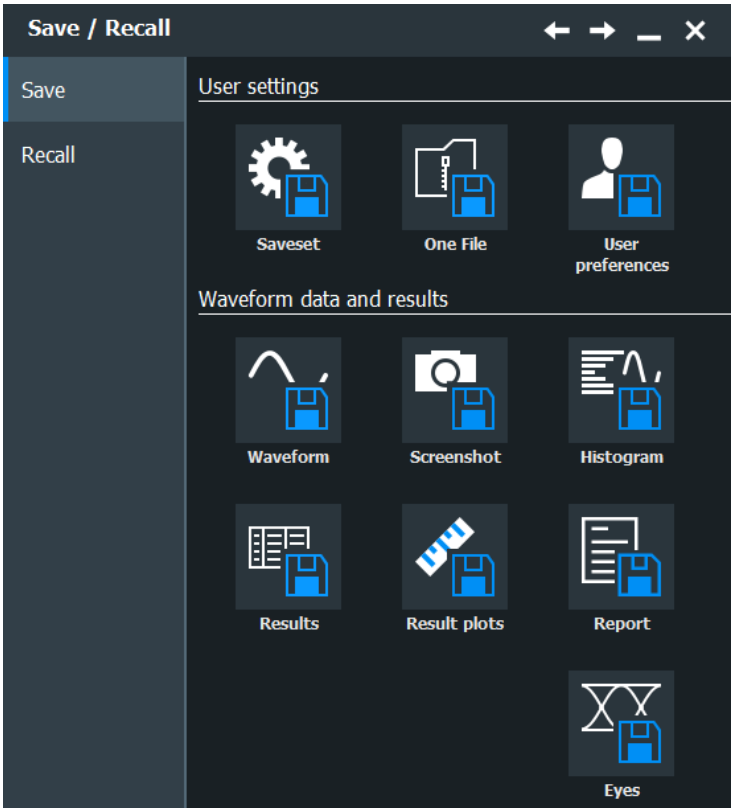



*Figure 12-1: Save tab*

Access to load instrument settings: "Menu" key > "Save/Recall" key > "Recall" tab > "User settings"

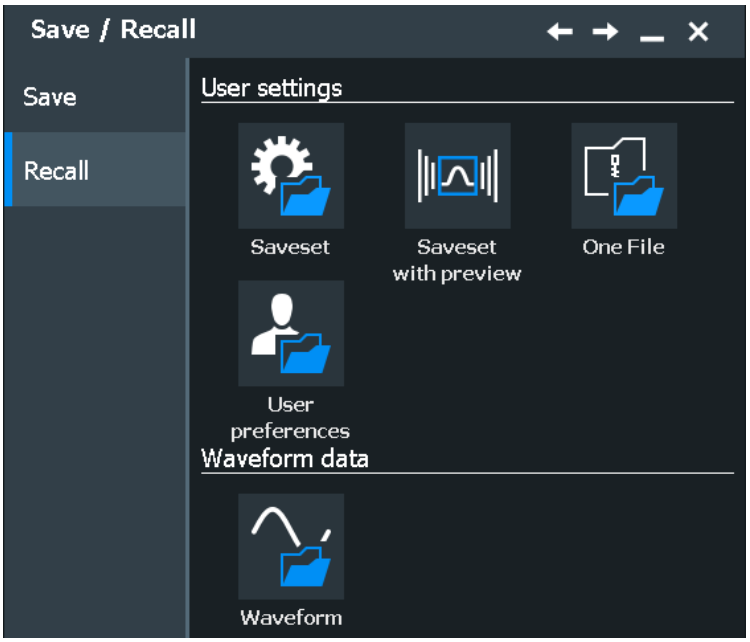



*Figure 12-2: Recall tab*

The R&S RTP provides the following types of saving and restoring settings:

- **Savesets** contain the complete instrument and measurement configuration except for user-specific display settings.
- **One File** contains the saveset and active reference waveform files in a ZIP file.
- **User preferences** contain the user specific display settings.
- **User-defined presets** contain the complete instrument setup including display settings, except for transparency and intensity. These settings can be restored by pressing the [Preset] key. See Chapter 4.6, "Preset setup", on page 101.

- Savesets ..... 475
- One File ..... 476
- User preferences ..... 478
- Saveset with preview function ..... 479
- Saving and loading settings ..... 480

## 12.1.1 Savesets

Savesets contain the complete instrument and measurement configuration including a screenshot of the current display, but except for user-specific display settings stored as user preferences. You can save an unlimited number of setting files.

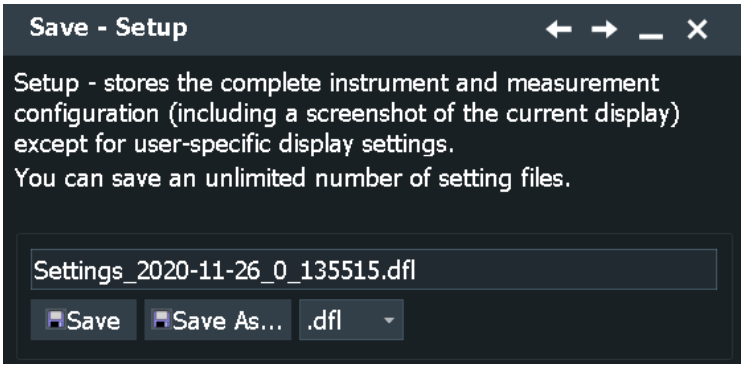
The waveform generator/pattern generator and the pulse source state is saved as "Off".

To load a saveset quickly, you can use the "Load saveset" function on the toolbar. A graphical preview helps you to find the required settings file.

If you need to store the instrument settings often, you can add the " Save settings" icon to the toolbar and use the icon to store the saveset file.

### 12.1.1.1 Save saveset settings

Access: "Menu" key > "Save/Recall" key > "Save" tab > "Savesets"



**Save to file**
Enter the file name to save the setting data to, and select the file format with the format button on the right. Double-tap the file name to open the file selection dialog box. See also: Chapter 12.5, "File selection dialog", on page 513.

By default, saveset file names have the prefix "Settings_".

"Save" Saves the data to the selected file.

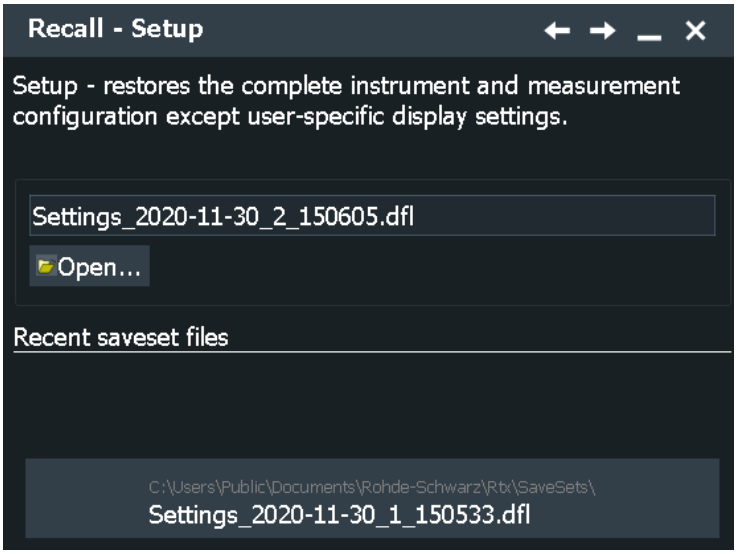
"Save As..." Opens the file selection dialog box and saves the data to the selected file.

".dfl/.xml" Selects the file format.

Remote command:
`MMEMory:SAV` on page 1610

#### 12.1.1.2 Load saveset settings

Access: "Menu" key > "Save/Recall" key > "Recall" tab > "Savesets"




**Load from file**
Enter the file name to load the setting data from, and select the file format with the format button on the right. Double-tap the file name to open the file selection dialog box. See also: Chapter 12.5, "File selection dialog", on page 513.

By default, saveset file names have the prefix "Settings_".

"Open" Opens a file selection dialog box and loads the selected file.

Remote command:
`MMEMory:RCL` on page 1611

### 12.1.2 One File

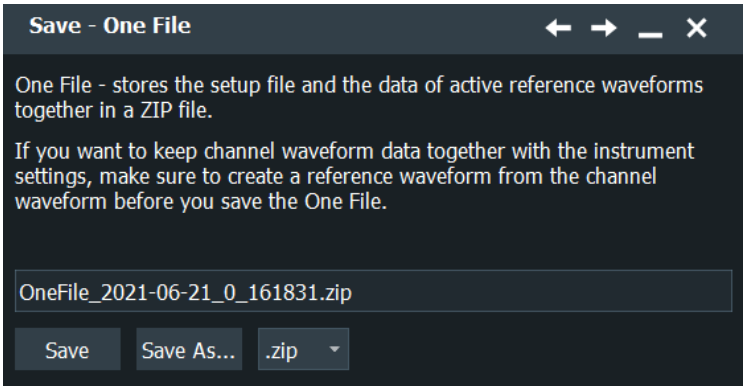
The One File function stores the savesets file and the data of active reference waveforms together in a ZIP file. If you want to keep channel waveform data together with

the instrument settings, make sure to create a reference waveform from the channel waveform before you save the One File.

The waveform generator/pattern generator and the pulse source state is saved as "Off".

#### 12.1.2.1 Save One File settings

Access: "Menu" key > "Save/Recall" key > "Save" tab > "One File"




**Save to file**
Enter the file name to save the One File data to, and select the file format with the format button on the right. Double-tap the file name to open the file selection dialog box.
See also: Chapter 12.5, "File selection dialog", on page 513.

By default, one file names have the prefix "OneFile_".

"Save" Saves the data to the selected file.

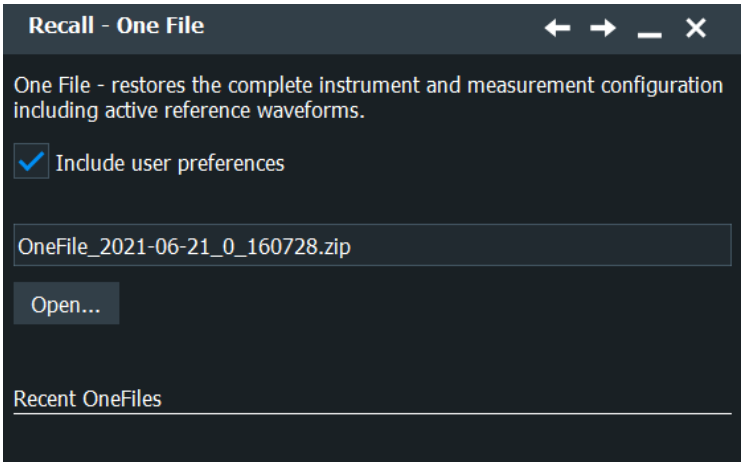
"Save As..." Opens the file selection dialog box and saves the data to the selected file.

".zip" Shows the file format.

Remote command:
`SAVeset:ONEFile:NAME` on page 1612
`SAVeset:ONEFile:SAVE` on page 1613

#### 12.1.2.2 Load One File settings

Access: "Menu" key > "Save/Recall" key > "Recall" tab > "One File"




**Load from file**
Select "Open" to open the file selection dialog box, and to select the required file. See also: Chapter 12.5, "File selection dialog", on page 513.

By default, one file names have the prefix "OneFile_".

Remote command:
`SAVeset:ONEFile:NAME` on page 1612
`SAVeset:ONEFile:OPEN` on page 1613

**Include user preferences**
If enabled, the user preferences settings are also restored together with the saveset and reference waveform. User preferences settings include the user-specific display settings like diagram layout, toolbar, intensity and transparency settings.

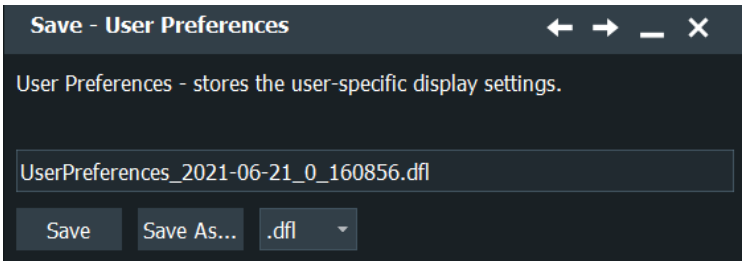
Remote command:
`SAVeset:ONEFile:USERpref` on page 1613

## 12.1.3 User preferences

User preferences contain user-specific display settings like diagram layout, toolbar, intensity and transparency settings.

### 12.1.3.1 Save user preferences settings

Access: "Menu" key > "Save/Recall" key > "Save" tab > "User Preferences"



**Save to file**
The file name to save the data to.

By default, user preference file names have the prefix "UserPreferences_".

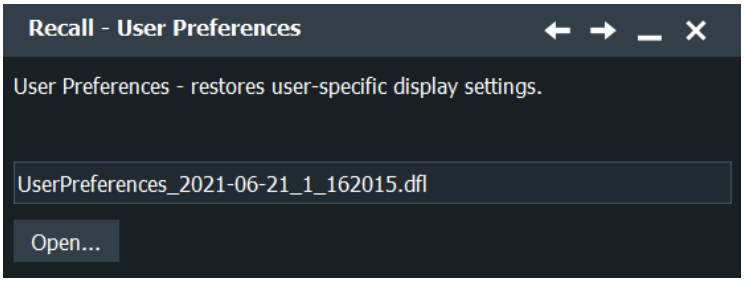
"Save" Saves the data to the selected file.

"Save As..." Opens the file selection dialog box and saves the data to the selected file.

".zip" Shows the file format.

#### 12.1.3.2 Load user preference settings

Access: "Menu" key > "Save/Recall" key > "Recall" tab > "User Preferences"




**Load from file**
Enter the file name to load the setting data from, and select the file format with the format button on the right. Double-tap the file name to open the file selection dialog box.
See also: Chapter 12.5, "File selection dialog", on page 513.

By default, user preference file names have the prefix "UserPreferences_".

"Open" Opens a file selection dialog box and loads the selected file.

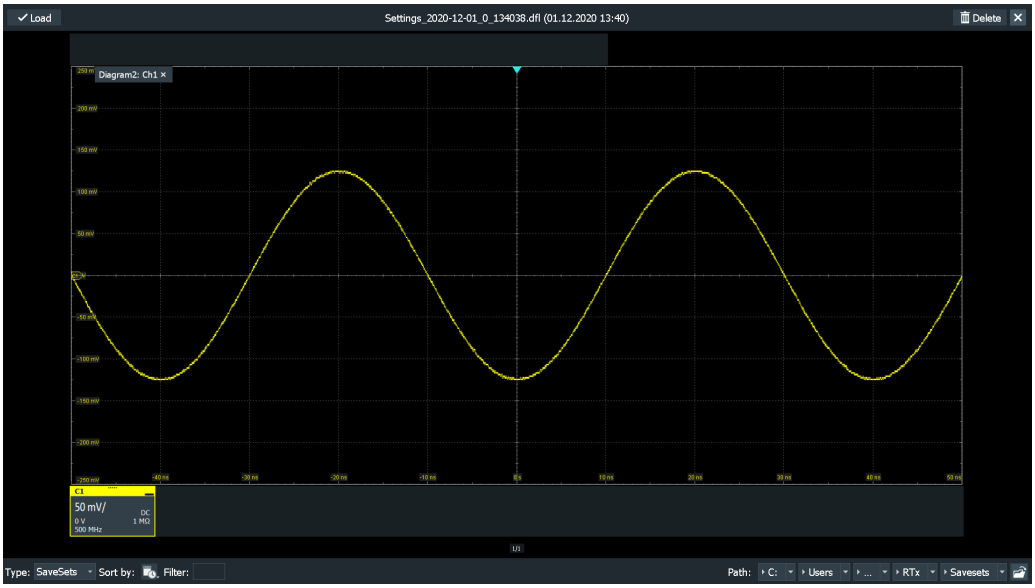
Remote command:
`MMEMory:RCL` on page 1611

### 12.1.4 Saveset with preview function

Access: "Menu" > "Save/Recall" > "Recall" tab > "Saveset with preview"

The graphical preview helps you to find the required settings file or a reference waveform.

The "Saveset with preview" function is also available on the toolbar, see Chapter 3.3.6.3, "Toolbar functions", on page 57.




1. On the toolbar, tap the "Saveset with preview " icon.

   A window opens and shows the screenshot of the first measurement configuration that is stored in the default directory.

2. Select if you want to display a "SaveSets" file or a reference waveform ("Ref Wfm").

3. Find the required file using one of the following methods:
   - Tap the "Next" icon on the right or the "Previous" icon on the left to scroll the savesets of the directory.
     The file name is shown on the top, and the screenshot helps to identify the settings.
   - If the saveset was stored in another directory, use the path buttons at the bottom, or tap "Open" to open the required directory.
   - Tap "Sort by" to sort the files according to the "Last modified" date or the "Filename".
   - Specify a "Filter" name to show only the files containing the filter string.

4. Tap "Load" in the upper left corner to recall the settings of the selected file.

### 12.1.5 Saving and loading settings

Settings can be stored in a file with user-defined name and location, or in a quick saveset. The settings in a saveset can be saved and retrieved quickly at the touch of a button, so savesets are ideal for frequently used measurements.

For details on save/recall instrument settings and associated remote commands, see Chapter 12.1.1, "Savesets", on page 475.

**To save settings to a saveset file**

Alternatively, you can add the " Save settings" icon to the toolbar and use the icon to store the saveset file to the folder and file specified in the "Settings" tab.

See also Chapter 3.3.6.2, "Configuring the toolbar", on page 56.

1. Tap "Menu" and select "Save/Recall".
2. In the "Save" tab, press the "Saveset" button .
3. Tap "Save" to save the settings to the specified file.
   Tap "Save As" to save the settings to a different file. Select the file and directory from the file selection dialog box.

   The current settings are saved to the selected file.

**To load settings from a saveset file**

Alternatively, you can use the "Recall" function on the toolbar.

1. Tap "Menu" and select "Save/Recall".
2. Select the "Recall" tab.
3. Press the "Saveset" button.
4. Tap "Load" to load the settings from the specified file.
   Tap "Open" to navigate to a different file. Select the file from the file selection dialog box and tap "Select".

   The saved settings are loaded to the R&S RTP.

**To save a One File**

1. If you want to save reference waveforms in the One File, create and display the reference waveforms.
2. Tap "Menu" and select "Save/Recall".
3. In the "Save" tab, tap the "One File".
4. Tap "Save" to save the settings to the specified file.
   Tap "Save As" to save the settings to a different file. Select the file and directory from the file selection dialog box.

   The current settings and the active reference waveforms are saved to the selected file.

**To load a One File**

1. Tap "Menu" and select "Save/Recall".
2. Press the "One File" button.
3. Tap "Load" to load the settings from the specified file.

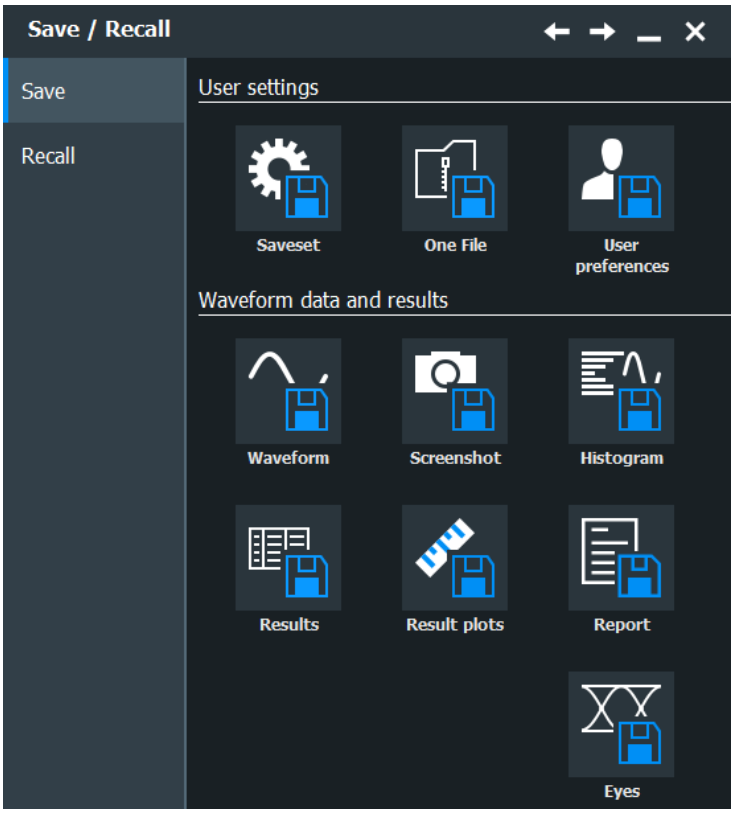
Tap "Open" to navigate to a different file. Select the file from the file selection dialog box and tap "Select".

The saved settings and reference waveforms are loaded to the R&S RTP.

## 12.2 Save and recall waveform data and results

Access: "Menu" key > "Save/Recall" > "Save" tab

You can export various data to file: waveform data, histograms, and measurement results.




For information on data export in I/Q mode (requires option), see Chapter 17.3, "I/Q data output", on page 1020.


- Waveform export files ..... 483
- Waveform settings ..... 490
- Histogram ..... 497
- Results ..... 499
- Result plots analysis ..... 500
- Eyes ..... 503
- Saving and loading waveform data ..... 505

### 12.2.1 Waveform export files

Waveforms can be stored in XML, CSV, or BIN format.

**Reloading waveforms: Restrictions**

To reload waveform data as a reference waveform, it must be stored in BIN format.

If multiple acquisitions of one waveform are exported (Data logging or Multiple waveforms), only the first acquisition can be reloaded.

If raw data was exported, after reload the waveform position is not the same as for the exported waveform.

If the signal is a spectrum, reloading is only possible for waveforms with "Magnitude unit" = Linear. Waveforms with logarithmic unit cannot be reloaded.

Data of all waveforms is saved in the following files:

- Header file: contains the header data, for example, time scale, vertical scale, vertical and horizontal positions, interpolation mode and much more. Header data is required to restore the waveform from data, or to analyze the data values of the data file.
- Waveform file: indicated by `*Wfm.*` in the filename. Contains the waveform data.

#### 12.2.1.1 Header files

The header files of XML and BIN waveform files are written in XML format. The header files of CSV waveform files are written in CSV format. You can open the header files and use their information for data analysis.

CSV header files only contain the property names and values, one property per row.

```
Resolution:1e-010:
RecordLength:1000:
```

XML header files contain more information than CSV header files. The additional information is required to reload the stored waveforms with their correct settings.

```
<Prop Avail="0" ValueKey="" Name="Resolution" Value="1e-010" UserValue="0"
Step="1e-011" Default="0" Min="0" Max="1e+026" StepDefault="1e-011"
StepFactor="10" Resolution="0" UnitId="55" UnitName="s" UnitPowerProduct=""
BitGroupSize="0" Format="0"></Prop>

<Prop Avail="0" ValueKey="" Name="RecordLength" Value="1000" UserValue="1000"
Step="1" Default="1000" Min="0" Max="4294967295" StepDefault="1" StepFactor="10"
Resolution="1" UnitId="93" UnitName="Sa" UnitPowerProduct="" BitGroupSize="0"
Format="0"></Prop>
```

Header files contain the following properties:

*Table 12-1: Header file properties*

| Value | Description |
|---|---|
| **General** | |
| FirmwareVersion | Firmware version that is installed on the R&S RTP (last entry in the header file) |
| Source | Name of the exported waveform |
| Resolution | Time between two samples<br>*Resolution = 1 / Sample Rate* |
| SignalResolution | Time between two samples in this waveform. The value can differ from Resolution if the source is, for example, a spectrum, a bus signal, a correlation or a measurement. The value is determined automatically considering the waveform parameters and their dependencies. If the signal is a spectrum, the value indicates the frequency range of FFT bins. |
| EnhancementMode | Method to increase the sample rate if the required sample rate is higher than the ADC sample rate. |
| InterpolationMode | Interpolation method. The value is relevant when the enhancement mode is interpolated time. |
| DecimationMode | Method to reduce the number of data samples to achieve the required sample rate |
| DecimationFactor | Factor to the number of data samples to achieve the required sample rate<br>*Decimation factor = ADC sample rate / Sample rate* |
| TraceArithmetics | Off, Envelope, or Average |
| InterleavedTraceCount | Number of y-values saved at each sampling time. The value is usually 1. The value is 2, if min and max values are saved for each sample, for example, for envelope waveforms. |
| SignalFormat | Format of the data values:<br>• FLOAT: floating point numbers, general export format<br>• INT (8 Bit): Integer 8 bit, used for "Raw (ADC sample)" data export.<br>• INT (16 Bit): Integer 16 bit, used for "Raw (ADC sample)" data export in high definition mode. |
| Timestamp | Absolute time of the waveform recording |
| ByteOrder | Endianness, only relevant for raw data export in high definition mode (SignalFormat = INT (16 Bit)).<br>• LSB first: little endian, least significant byte first<br>• MSB first: big endian, most significant byte first |
| NumericFormat | Number format of bus values and digital channel data (bit pattern format) |
| **Record length** | |
| RecordLength | Number of samples in a waveform record of one acquisition |
| HWRecordLength | Equivalent to the RecordLength |
| SignalRecordLength | Number of required samples in the waveform. The value can differ from RecordLength and HWRecordLength if the source is, for example, a spectrum, a bus signal, a correlation or a measurement. The value is determined automatically considering the waveform parameters and their dependencies. If the signal is a spectrum, the value indicates the number of FFT bins. |

| Value | Description |
|---|---|
| SignalHardwareRecordLength | Number of samples actually available in this waveform, including the number of required samples in the waveform and the additional samples needed for further computation |
| LeadingSettlingSamples | Relevant only for BIN files. In XML and CSV files, the value is 0.<br>Number of additional samples before the beginning of waveform samples. These additional samples are needed for further computation, for example, for filters. |
| **Horizontal system** | |
| TimeScale | Horizontal scale in seconds per division |
| HorizontalDivisionCount | Number of horizontal divisions |
| RescaleCenterTime | Horizontal position, the time distance between the reference point and the zero point of the diagram |
| RescaleCenterPoint | Position of the reference point in % of the screen |
| ReferencePoint | Position of the zero point in % of the screen |
| TriggerOffset | Time distance from the trigger point to the zero point of the diagram |
| XStart | Horizontal start value of the waveform (time or frequency) *) |
| XStop | Horizontal stop value of the waveform (time or frequency) |
| HardwareXStart | Actual horizontal start value of data, including the settling time for further computation *) |
| HardwareXStop | Actual horizontal stop value of data, including the settling time for further computation |
| | *) If the waveform is a spectrum, the XStart and HardwareXStart values may be slightly smaller than the specified start frequency, or even get negative. The spectrum is centered on the center frequency, and the frequency range covered by one spectral bin is given by the SignalResolution. Hence, the spectral bin in the center of the spectrum always covers the range [CenterFrequency; CenterFrequency + SignalResolution]. As a result, the range covered by the first spectral bin in the spectrum may reach further than the specified start frequency. It is ensured that the specified start frequency is included in the frequency range. |
| **Vertical system**<br>If multiple channels are exported, the values of channel1 are delivered, no matter if channel 1 is exported or not. | |
| VerticalScale | Vertical scale of the waveform in Volts per division, or other unit / division |
| VerticalDivisionCount | Number of vertical divisions |
| VerticalPosition | Vertical position of the waveform in divisions |
| VerticalOffset | Vertical offset of the waveform in Volts, or other unit |
| NofQuantisationLevels | Theoretical number of quantization levels in the signal. This value depends on the waveform format (8 bit, 16 bit, …). In case of a math waveform, it depends on the quantization levels of the operands and on the operator type. |
| BaseYStart | Vertical start value of the waveform |

| Value | Description |
|---|---|
| BaseYStop | Vertical stop value of the waveform |
| **Multi channel export**<br>The header files contain strings like this: `MultiChannelVerticalOffset: 4:1.63:1.96:0:0:1e-005:0:-1e+026:1e+026:1e-005:10:0:V:`. Only the first 5 values and the unit at the end of the string are relevant for data analysis. All other values are for internal use and not explained here.<br>Examples are in csv format. | |
| MultiChannelExport | Indication whether multiple channels are exported simultaneously: On \| Off |
| MultiChannelExportState | Number of channels and export status of the individual channels, for example, 4:On:Off:On:On...: channels 1, 3 and 4 are exported. |
| MultiChannelVerticalOffset | Number of channels and vertical offset of the individual channels, for example, 4:0:0:0.02:0...: channel 3 has an offset of 20 mV. |
| MultiChannelVerticalPosition | Number of channels and vertical position of the individual channels, for example, 4:0:0:0:2...: the position of channel 4 is 2 divisions. |
| MultiChannelVerticalScale | Number of channels and vertical scale of the individual channels, for example, 4:0.05:0:0.03:0.04...: scale of channel is 50 mV/div, channel 3 has 30 mV/div and channel 4 has 40 mV/div. |
| MultiChannelBaseYStart | Number of channels and minimum value of the vertical range for each individual channel, for example, 4:-0.25:0:-0.13:-0.28... |
| MultiChannelBaseYStop | Number of channels and maximum value of the vertical range for each individual channel, for example, 4:0.25:0:0.17:0.12...: The range of channel 1 is -250 mV to 250 mV. The range of channel 3 is -130 mV to 170 mV. The range of channel 4 is -280 mV to 120 mV. |
| History | |
| TimestampState | State of the timestamps export. If on, the timestamps of each history waveform are written to the waveform data file. |
| **Math waveform** | |
| BaseUnit | Base unit of a mathematic waveform, for example, linear unit |
| ViewUnit | User-selected unit of a mathematic waveform, for example, logarithmic unit for a spectrum. The value is only valid if the exported waveform is a math waveform. |
| ViewUnitRelative | Indication of a relative unit. It is true if the math waveform has the ViewUnit "dB", for example. The value is only valid if the exported waveform is a math waveform. |
| ViewReferenceLevel | Reference level for a relative unit. The value is only valid if the exported waveform is a math waveform, and the unit is relative. |
| **FFT** | |
| CenterFreq | Center frequency of the spectrum |
| FreqSpan | Frequency span of the spectrum |
| FrequencyStart | Start frequency of the spectrum |
| FrequencyStop | Stop frequency of the spectrum |

| Value | Description |
|---|---|
| WindowType | Window used for the spectrum computation |
| ResolutionBW | Resolution bandwidth of the spectrum |
| AdjustedResolutionBW | Actual resolution bandwidth of a spectrum waveform. The value is only valid if the exported waveform is a spectrum. |
| GateRBWCoupling | Indication whether the record length or the resolution bandwidth is a constant for the spectrum computation |
| **Parameters for power calculation** | |
| Impedance | Impedance used for power calculation |
| NoiseBandwidth | Noise bandwidth of a spectrum waveform, required for power calculation. The value is only valid if the exported waveform is a spectrum. |
| **Parameters for internal use** | |
| SourceType | Source qualifier |
| TraceType | Waveform qualifier |
| ValueType | |
| TOADone | |
| BaseUnitRelative | Base unit indication |
| UseInterSampleTriggerOffset | |
| ISO_TRG<br>SC_POST<br>SC_TRG | |

If "Include CDR info" is enabled, additional properties are appended to the header.

| Value | Description |
|---|---|
| AnalogSource | Analog source used for clock recovery |
| Algorithm | Software algorithm that is used for the hardware clock data recovery: constant frequency or feed forward |
| Bitrate | Bitrate |
| UnitIntervalOffset | Unit interval offset, number between 0 and 1 |
| PLLBandwidth | The bandwidth of the feed forward algorithm |
| PLLRelativeBandwidth | The relative bandwidth, that is the ratio of the nominal bit rate to the PLL bandwidth |

| Value | Description |
|---|---|
| SignalFormat | Format of the data values:<br>• FLOAT: floating point numbers, general export format<br>• INT (8 Bit): Integer 8 bit, used for "Raw (ADC sample)" data export.<br>• INT (16 Bit): Integer 16 bit, used for "Raw (ADC sample)" data export in high definition mode. |
| RecordLength | Number of samples in a waveform record of one acquisition |

#### 12.2.1.2 Waveform data files

The waveform data files - indicated by `*Wfm.*` in the filename - contain the actual waveform data. Usually only Y-values - mostly voltage values - are written subsequently. If the signal is a spectrum, the data of the last frame is written.

If the waveform consists of minimum and maximum values, two Y-values per sample are written, and the property `InterleavedTraceCount` in the header file is >1. This applies to envelope waveforms, for example.

The option "Interleaved X/Y" allows you to include horizontal values into the file.

If multi-channel export is enabled, the Y-values of the selected channels are written in interleaved order.

- One channel, single acquisition export
  - Normal waveform:
    $Y_0$; $Y_1$; $Y_2$; $Y_3$; ...
  - Envelope waveform:
    $Ymin_0$; $Ymax_0$; $Ymin_1$; $Ymax_1$; $Ymin_2$; $Ymax_2$; $Ymin_3$; $Ymax_3$; ...
  - Normal waveform, interleaved x/y data:
    $X_0$; $Y_0$; $X_1$; $Y_1$; $X_2$; $Y_2$; $X_3$; $Y_3$; ...
  - Envelope waveform, interleaved x/y data:
    $X_0$;$Ymin_0$; $Ymax_0$; $X_1$; $Ymin_1$; $Ymax_1$; $X_2$; $Ymin_2$; $Ymax_2$; $X_3$; $Ymin_3$; $Ymax_3$; ...
- Multi-channel, single acquisition export
  In the example, two channels are exported.
  - Normal waveforms:
    $YCh1_0$; $YCh2_0$; $YCh1_1$; $YCh2_1$; $YCh1_2$; $YCh2_2$; $YCh1_3$; $YCh2_3$; ...
  - Envelope waveforms, channel 1 and channel 2 are envelopes:
    $YCh1min_0$; $YCh1max_0$; $YCh2min_0$; $YCh2max_0$; $Ymin_1$; $Ymax_1$; $YCh2min_1$; $YCh2max_1$; $Ymin_2$; $Ymax_2$; $YCh2min_2$; $YCh2max_2$; $Ymin_3$; $Ymax_3$; $YCh2min_3$; $YCh2max_3$; ...
  - Normal waveforms, interleaved x/y data:
    $X_0$; $YCh1_0$; $YCh2_0$; $X_1$; $YCh1_1$; $YCh2_1$; $X_2$; $YCh1_2$; $YCh2_2$; $X_3$; $YCh1_3$; $YCh2_3$; ...
  - Envelope waveform and normal waveform, interleaved x/y data:
    $X_0$;$YCh1min_0$; $YCh1max_0$; $YCh2_0$; $X_1$; $YCh1min_1$; $YCh1max_1$; $YCh2_1$; $X_2$; $YCh1min_2$; $YCh1max_2$; $YCh2_2$; $X_3$; $YCh1min_3$; $YCh1max_3$; $YCh2_3$; ...

In XML and CSV waveform value files, the data of each sample is grouped. The example shows the values of two samples for two waveforms and interleaved x/y data. The first waveform is an envelope, the second one is a normal waveform.

In CSV files, the data values for a given sampling time are written in one row.

```
-1.96e-008    -0.0079051387    -0.0059288535    -0.1027668
-1.95e-008    -0.0098814229    -0.0079051387    -0.10474309
```

In XML format, an empty line marks the beginning of the next sample.

```
<Data>-1.96e-008</Data>
<Data>-0.0079051387 </Data>
<Data>-0.0059288535 </Data>
<Data>-0.1027668 </Data>

<Data>-1.95e-008</Data>
<Data>-0.0098814229 </Data>
<Data>-0.0079051387 </Data>
<Data>-0.1027668 </Data>
```

If multiple acquisitions (Data logging / Multiple waveforms) are exported, the first acquisition is written in the same way as with single acquisition export. The following acquisitions are appended in the same way. If the signal is a spectrum, the last frame of each acquisition is saved.

If the history is exported in one file with timestamps, the timestamp as double float precedes the data of each acquisition. The acquisition data ist written in the same way as with single acquisition, according to the selected settings.

```
<timestamp as double float of first acquisition>
<value 1>
<value 2>
<value ...>
<timestamp as double float of second acquisition>
<value 1>
<value 2>
<value ...>
```

In BIN files, the instrument writes some leading and trailing settling samples before and after the waveform data. They ensure that the analysis of the reloaded reference waveform returns the same results as analysis of the original waveform. The number of leading settling samples is provided in the header file.

#### 12.2.1.3 Number of samples in the waveform data file

In this section, a sample is defined as one or more values acquired at a given sampling time. The number of samples for one channel and acquisition is given in the header file by the property `SignalHardwareRecordLength`.

If the waveform has more than one Y-value per sample (e.g. envelope), the property `InterleavedTraceCount` is > 1, and the number of values in the file for this waveform is:

*No of values per waveform = InterleavedTraceCount * SignalHardwareRecordLength*

If multiple acquisitions are exported, the total number of values in the file is:

*No of values = InterleavedTraceCount * SignalHardwareRecordLength * No of exported acquisitions*

If "Interleaved x/y"is enabled, one horizontal value is added per sample. The total number of values in the file is:

*No of values = (1+ InterleavedTraceCount) * SignalHardwareRecordLength * No of exported acquisitions*

In BIN files, the value `SignalHardwareRecordLength` includes the number of required samples in the waveform and additional samples at the beginning (leading samples) and the end of the file (trailing samples). The number of additional samples is:

*No of additional samples = SignalHardwareRecordLength - SignalRecordLength*

The number of leading additional samples is given in the header file: `LeadingSettlingSamples`.

The number of trailing additional samples is:

*No of trailing additional samples = No of additional samples - LeadingSettlingSamples*
*= SignalHardwareRecordLength - SignalRecordLength - LeadingSettlingSamples*

MSO option R&S RTP-B1:

If the data of digital channels is stored in BIN format, 1 bit is written for each sample. 8 data samples are written in 1 byte (data word). Thus, the file size is

*File size = Number of samples / 8*

For example, 100 MSa are written into a 12.5 MByte BIN file. After reading the file, you have to extract the samples from the data words.

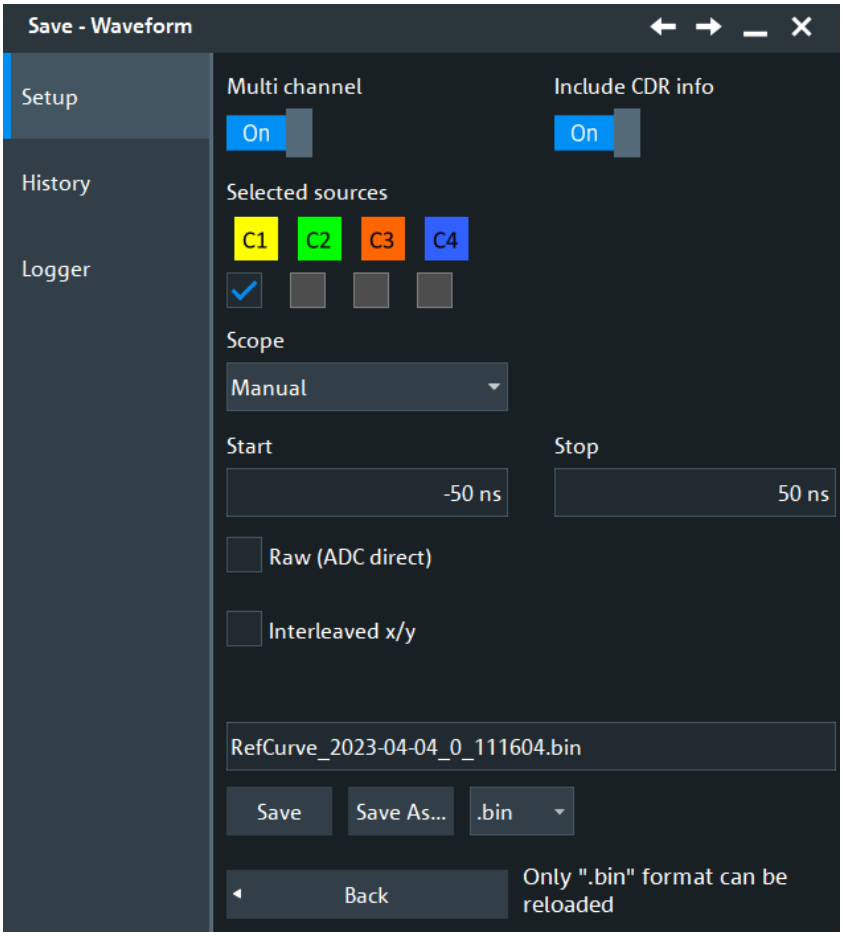
## 12.2.2 Waveform settings

In the "Save - Waveform" dialog, the storage settings for waveform data are defined.

See also: Chapter 12.2.7, "Saving and loading waveform data", on page 505.

### 12.2.2.1 Waveform setup settings

Access: "Menu" key >"Save/Recall" key > "Save" tab > "Waveform" > "Setup" tab.




**Multi channel**
Enables or disables the export of multiple input channels. If enabled, you can export the data of selected input channels (Selected sources) into one file.

If disabled, you can export one "Source" on page 492 waveform.

You can reload exported multiple channels if they are stored in BIN format. The import asks you to assign each stored waveform to a reference waveform.

Remote command:
`EXPort:WAVeform:MULTichannel` on page 1615

**Include CDR info**
Available, if all the following conditions are fulfilled:

- Type = "CDR"
- State = "On"
- Clock / CDR / Protocol = "CDR"

Includes the CDR information in a waveform file. If this function is enabled, an additional `RefCurve.CDR.*.bin/*.csv/*.xml` file is created. This file is available in `*.bin`, `*.csv` and `*.xml` format.

The file contains the CDR trigger timestamps corresponding to the waveform data.

The created file can be used as a reference source for math equations and for the advanced eye.

Remote command:
EXPort:WAVeform:ICDR on page 1622

**Source**
Selects the waveform to be exported if "Multichannel export" is disabled.

Active waveforms of input channels, math signals, reference waveforms and advanced jitter analysis are available for export.

If the MSO option R&S RTP-B1 is installed, you can save also digital channels and parallel buses.

If the selected source is defined as average or envelope waveform, make sure that all required acquisitions ("Average count") are done before you save the waveform. Otherwise, the exported waveform data are not correct.

Remote command:
EXPort:WAVeform:SOURce on page 1614

**Selected sources**
Select the channels to be included in data export if "Multi channel" is enabled. Waveform1 of up to four input channels can be saved into one file.

If at least one of the selected sources is defined as average or envelope waveform, make sure that all required acquisitions ("Average count") are done before you save the waveforms. Otherwise, the exported waveform data are not correct.

Remote command:
CHANnel<m>:EXPortstate on page 1616

**Scope**
Defines the part of the waveform record that has to be stored.

| | |
|---|---|
| "Full waveform" | Saves the complete waveform record. |
| "Zoom" | Saves the data included in the zoom area if at least one zoom is defined for the source waveform. The start and stop values of the area are shown. If several zooms are defined, select the "Zoom" to be used for export. |
| "Cursor" | Saves the data between the cursor lines if at least one cursor measurement is defined for the source waveform. The start and stop values of the area between the cursor lines are shown. If several cursor sets are defined, select the "Cursor set" to be used for export. |
| "Gate" | Saves the data included in the measurement gate if a gated measurement is defined for the source waveform. Select the "Measurement" for which the required gate is defined. The start and stop values of the gate are shown. |
| "Manual" | Saves the data between user-defined "Start" and "Stop" values. |

Remote command:
EXPort:WAVeform:SCOPe on page 1617
EXPort:WAVeform:STARt on page 1617

EXPort:WAVeform:STOP on page 1618
EXPort:WAVeform:ZOOM on page 1618
EXPort:WAVeform:CURSorset on page 1618
EXPort:WAVeform:MEAS on page 1619

**Raw (ADC direct)**
Enables the export of analog channel data in the raw sample format of the ADC. The data format is integer 8 bit (signed 8-bit binary format). This format reduces the file size (1 Byte/sample instead of 4 Bytes/sample in binary files) but decreases the precision of the values.

If the high definition mode is active, the data format is integer 16 bit, except for peak detect decimation (8 bit). See "Export" on page 139.

Only y-values are exported, the "Interleaved x/y" option is not available.

Currently, the setting is not available for the export of digital channel data and data of R&S RT-ZVC channels.

***Data conversion:***

To convert INT8 or INT16 data to physical quantities, e.g. voltages, use the following formulas:

*ConversionFactor = VerticalScale * VerticalDivisionCount / NofQuantisationLevels*

*PhysicalQuantity = (Value_ADC * ConversionFactor) + VerticalOffset - (VerticalScale * VerticalPosition)*

The raw values are written in the `*.Wfm.*` file, all other values can be found in the corresponding header file.

***Table 12-2: Example of raw data conversion***

| | INT8 | INT16, HD mode |
|---|---|---|
| VerticalScale | 110 mV/div | 50 mV/div |
| VerticalPosition | 1 div | 0 |
| VerticalOffset | 50 mV | 0 |
| NofQuantisationLevels | 253 | 253 * 256 |
| VerticalDivisionCount | 10 | 10 |
| Value_ADC | 13 | -61 |
| ConversionFactor | 0.11 * 10 / 253 = 0.004347826086957 | 0.05 * 10 / (253 * 256) = 0.0000771986 |
| Voltage | (13 * 0.004347826086957) + 0.05 - (1 * 0.11) = -0.003478260869559 V = 3.478260869559 mV | (-61 * 0.0000771986) + 0 = -4.7091146 mV |

When reloading an exported raw BIN waveform, the waveform position after reload is not the same as for the exported waveform.

Remote command:
EXPort:WAVeform:RAW on page 1621

**Interleaved x/y**
Includes horizontal values in the export data (time or frequency values, depending on the waveform). X and Y-values are written alternately to the file. If disabled, only Y-values - mostly voltage values - are written.

Interleaved x/y data cannot be exported as raw values, the "Raw (ADC direct)" option is not available.

Remote command:
`EXPort:WAVeform:INCXvalues` on page 1620

**Save to file**
Enter the filename to save the waveform to. Double-tap the filename to open the file selection dialog box.

By default, the filename has the prefix "RefCurves_". You can define a pattern for automatic naming in the "Autonaming" dialog, see Chapter 4.8.1, "Autonaming", on page 107.

"Save" Saves the waveform as a reference waveform in the selected file.

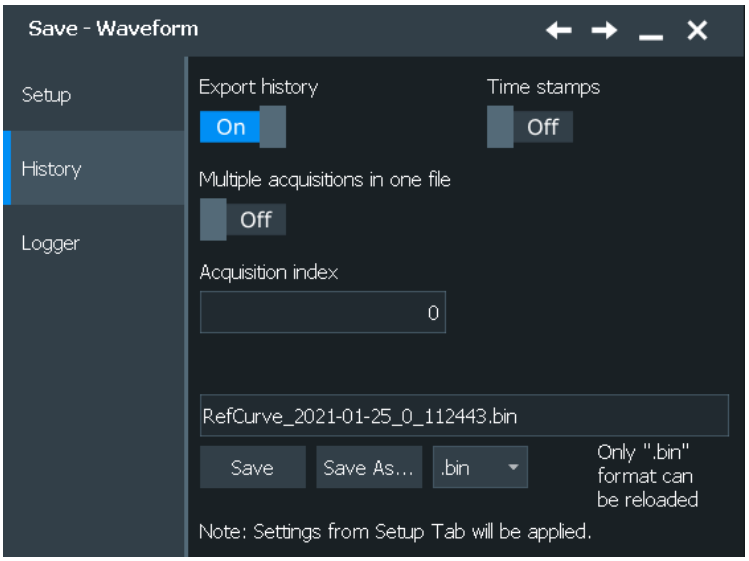
"Save As..." Opens the file selection dialog box and saves the waveform to the selected file. See also Chapter 12.5, "File selection dialog", on page 513

".bin/.xml/.csv" Selects the file format. Note that reference waveforms can be loaded from `.bin` files only.
See also: Chapter 12.2.1, "Waveform export files", on page 483.

Remote command:
`EXPort:WAVeform:NAME` on page 1616
`EXPort:WAVeform:SAVE` on page 1617

### 12.2.2.2 Waveform history settings

Access: "Menu" key >"Save/Recall" key > "Save" tab > "Waveform" > "History" tab.



In this dialog you can configure the export of the history waveform data to file. Select the channels and scope in the "Setup" tab.

**Export history**
Enables the history mode and the export of history waveforms to file.

Remote command:
`CHANnel<m>[:WAVeform<n>]:HISTory[:STATe]` on page 1405

**Time stamps**
If enabled, the relative timestamps of all history waveforms are written into the waveform data file at the beginning of each waveform record.

The following format is used:

```
<timestamp as double float of first acquisition>
<value 1>
<value 2>
<...>
<timestamp as double float of second acquisition>
<value 1>
<value 2>
<...>
```

You can also select the format of the time stamp in the "Acquire" > "History" dialog.

Remote command:
`EXPort:WAVeform:TIMestamps` on page 1620

**Multiple acquisitions in one file**
If enabled, allows you to save several or all history waveforms. Define the part of the history to be exported using "Start acquisition" and "Stop acquisition".

If multiple acquisitions of one waveform are exported into a BIN file, the first acquisition can be reloaded as reference waveform.

Remote command:
`CHANnel<m>[:WAVeform<n>]:HISTory:STARt` on page 1406
`CHANnel<m>[:WAVeform<n>]:HISTory:STOP` on page 1407

**Start export**
Starts the history replay and simultaneous saving.

Remote command:
`CHANnel<m>[:WAVeform<n>]:HISTory:PLAY` on page 1408

**Save to file**
Enter the filename to save the waveform to. Double-tap the filename to open the file selection dialog box.

By default, the filename has the prefix "RefCurves_". You can define a pattern for automatic naming in the "Autonaming" dialog, see Chapter 4.8.1, "Autonaming", on page 107.

"Save" Saves the waveform as a reference waveform in the selected file.

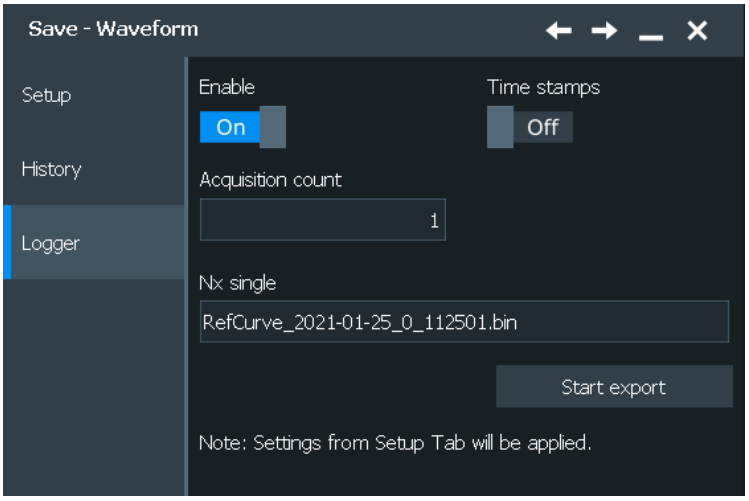
"Save As..." Opens the file selection dialog box and saves the waveform to the selected file. See also Chapter 12.5, "File selection dialog", on page 513

".bin/.xml/.csv" Selects the file format. Note that reference waveforms can be loaded from `.bin` files only.
See also: Chapter 12.2.1, "Waveform export files", on page 483.

Remote command:
`EXPort:WAVeform:NAME` on page 1616
`EXPort:WAVeform:SAVE` on page 1617

### 12.2.2.3 Waveform logger settings

Access: "Menu" key >"Save/Recall" key > "Save" tab > "Waveform" > "Logger" tab.




In this dialog, you can configure the export of the data of a running Nx Single acquisition. Select the channels and scope in the "Setup" tab.

**Enable**
Enables the export of all waveforms of an Nx Single acquisition into one file.

The waveform records are written in historical order one after the other, either the complete records or the sections as defined in Scope. Set the number of acquisitions to be acquired and stored with "Acquisition count". The maximum amount of data that can be written is shown in "Max. file size".

Enabling "Data logging" stops a running acquisition. To start the logging, tap "Start Export" on page 496 or press [Single].

Pressing "Run cont" disables data logging.

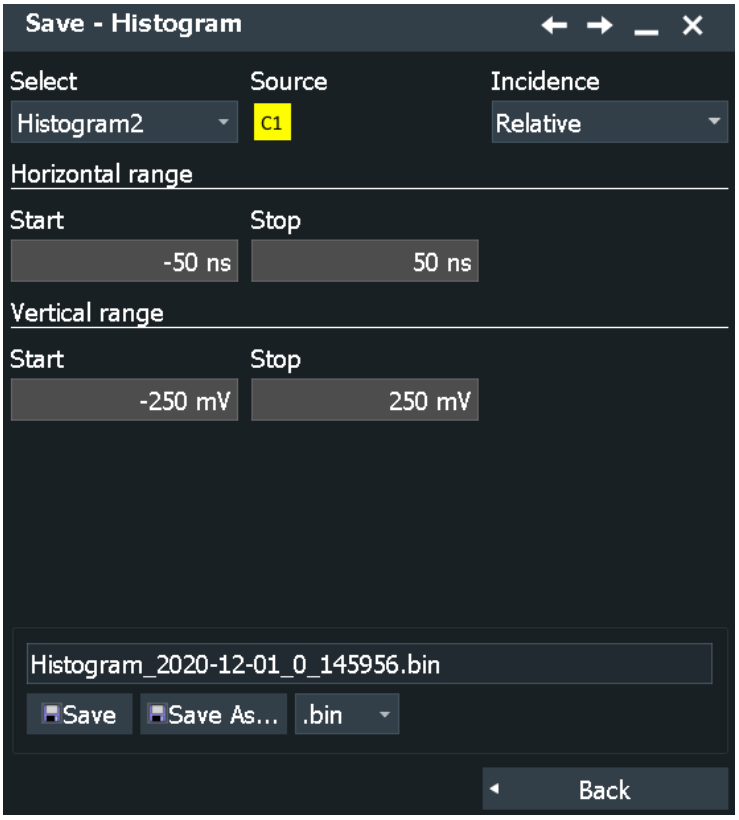
Remote command:
`EXPort:WAVeform:DLOGging` on page 1619

**Start Export**
Starts an Nx Single acquisition series and simultaneously saves the waveform data to a file.

Remote command:
RUNSingle on page 1235 (Nx Single acquisition)

### 12.2.3 Histogram

Access: "Menu" key >"Save/Recall" key > "Save" tab > "Histogram"




The waveform histogram export saves data in two files. The `*.Wfm.*` file contains 256 or 512 absolute or relative histogram values. The other file is the header file.

Contents of the header file:

- Source waveform of the histogram
- Histogram mode: vertical or horizontal
- Incidence of exported values: absolute or relative
- Histogram range: XStart, XStop, YStart, YStop
- Name of the exported histogram

***Data conversion:***

Using the header data, you can calculate the waveform value to which a histogram value belongs:

*YValue = (YStop - YStart) / HistogramValuesCount * HistogramValueNumber + YStart*

| | |
|---|---|
| YStart | -0.25 V |
| YStop | 0.25 V |
| HistogramValuesCount | 256 (total number of written rows in a CSV file) |
| HistogramValueNumber | 68 (number of the row in a CSV file) |
| Y-Value | (0.25 - (-0.25)) / 256 * 68 - 0.25 = -0.11719 V |

The histograms settings are defined in the "Histogram" dialog box ("Menu" > "Apps" > "Analysis" tab > "Histogram").

See also: Chapter 8.2.8.4, "Histogram setup", on page 373.

**Select**
Selects the histogram to be exported. All active waveform histograms are shown in the list.

Measurement histograms can also be exported, see Chapter 12.2.5, "Result plots analysis", on page 500.

Remote command:
`EXPort:HISTogram:SELect` on page 1622

**Source**
Displays the source of the selected histogram.

**Incidence**
Sets the mode of exported histogram data: relative or absolute count of values. If relative values are exported, the sum of all values is 1, and the count of each value is set in relation to the sum.

Remote command:
`EXPort:HISTogram:INCidence` on page 1622

**Horizontal range, Vertical range**
Displays the "Start" and "Stop" limits of the histogram area.

**Save to file**
Enter the filename to save the waveform histogram to. Double-tap the filename to open the file selection dialog box.

By default, the filename has the prefix "Histogram_". You can define a pattern for automatic naming in the "Autonaming" tab. The default directory is:

`C:\Users\Public\Documents\Rohde-Schwarz\RTP\Histograms`

"Save" Saves the histogram data in the selected file.

"Save As..." Opens the file selection dialog box and saves the histogram data to the selected file. See also Chapter 12.5, "File selection dialog", on page 513

".bin/.xml/.csv" Selects the file format.

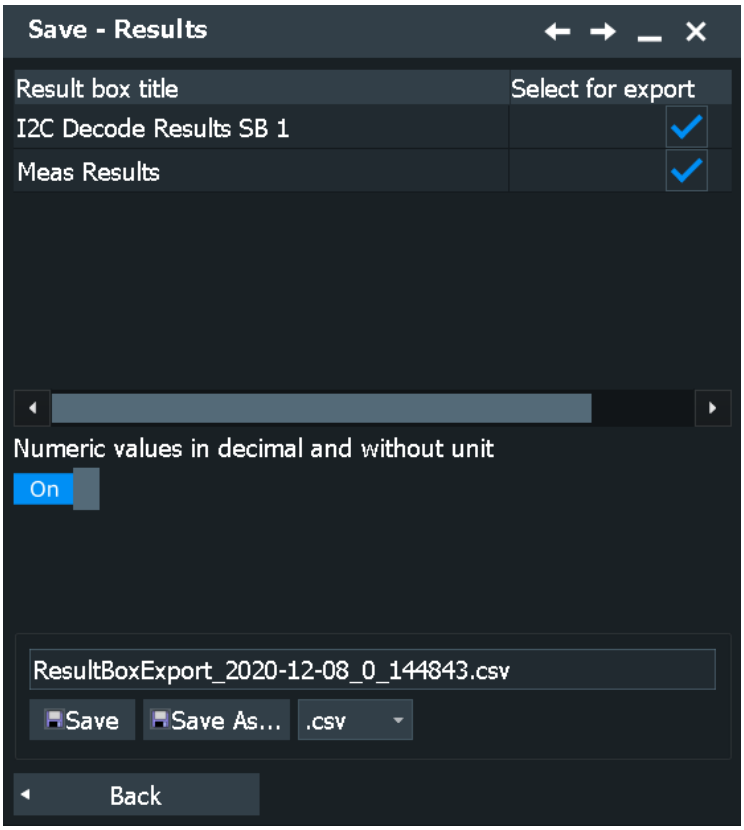
Remote command:
EXPort:HISTogram:NAME on page 1622
EXPort:HISTogram:SAVE on page 1623
EXPort:HISTogram:DATA? on page 1623

### 12.2.4 Results

In this dialog, you can select the result boxes to be saved, and define the storage settings.

Access: "Menu" key >"Save/Recall" key > "Save" tab > "Results"

Furthermore, you can export the decode result tables of serial protocol analysis.




**Result selection**
The table lists all result boxes and decode tables that are currently open, including minimized boxes and docked boxes. Select the results that you want to save to file. All results are written into one file.

**Note:** If the result box is minimized, only the columns shown on the result icon are saved (2 columns). Statistical results are not shown on the minimized results icon, and they are not saved.

**Numeric values in decimal and without unit**
By default, numeric result values are written with their unit to the file. If the option is enabled, the values are saved with more decimal places.

**Export results**
Enter the filename to save the results to. Double-tap the filename to open the file selection dialog box.

By default, the filename has the prefix "ResultBoxExport_".

You can define a pattern for automatic naming in the "Menu" > "Settings" > "Save/Recall" > "Autonaming".

"Save" Saves the selected results to the indicated file.

"Save As..." Opens the file selection dialog box and saves the selected results to the selected file. See also Chapter 12.5, "File selection dialog", on page 513

".csv/.html" Selects the file format.

- CSV: the values are saved in a file
  You can select the value delimiter and the list separator symbol in the "CSV Export" dialog, see Chapter 4.8.2, "CSV export", on page 110.
  Tip for using MS Excel: It is recommended that you use the semicolon as csv decimal symbol. When you open the file with MS Excel, use "File > Open" and follow the wizard to set the separators correctly, or set the separator settings with "Tools > Options > International".
- HTML: Results are saved as web page for display in a browser.

### 12.2.5 Result plots analysis

Access: "Menu" key >"Save/Recall" key > "Save" tab > "Result Plots"

You can export the data of long-term measurements, the measurement histogram and track data to file.

The measurement export saves results in two files. The `*.Wfm.*` file contains data values, and the other file is the header file.

The header file contains:

- Source waveform of the measurement
- Measurement scale
- Export type = histogram, long term or track
- Exported measurement
- Histogram range: XStart, XStop, YStart, YStop
  The range is only relevant for export type = histogram. The measurement axis is the X-axis, which can be a horizontal or vertical axis depending on the histogram mode.

**Long-term measurements:** The `*.Wfm.*` file contains one value or value set for each long-term measurement point. The maximum number of points is defined in the "Horizontal scaling" dialog box.

- If statistics are disabled, the current result of the main measurement is written - one double value per long-term point.
- If statistics are enabled, seven values for each long-term point are saved:
  - Upper peak
  - Lower peak
  - RMS
  - Standard deviation
  - Average
  - Event count per point: number of measurement results that creates one long-term point
  - Waveform count per point: number of waveforms included in one long-term point.

**Measurement histogram:** The `*.Wfm.*` file contains 1000 absolute or relative histogram values.

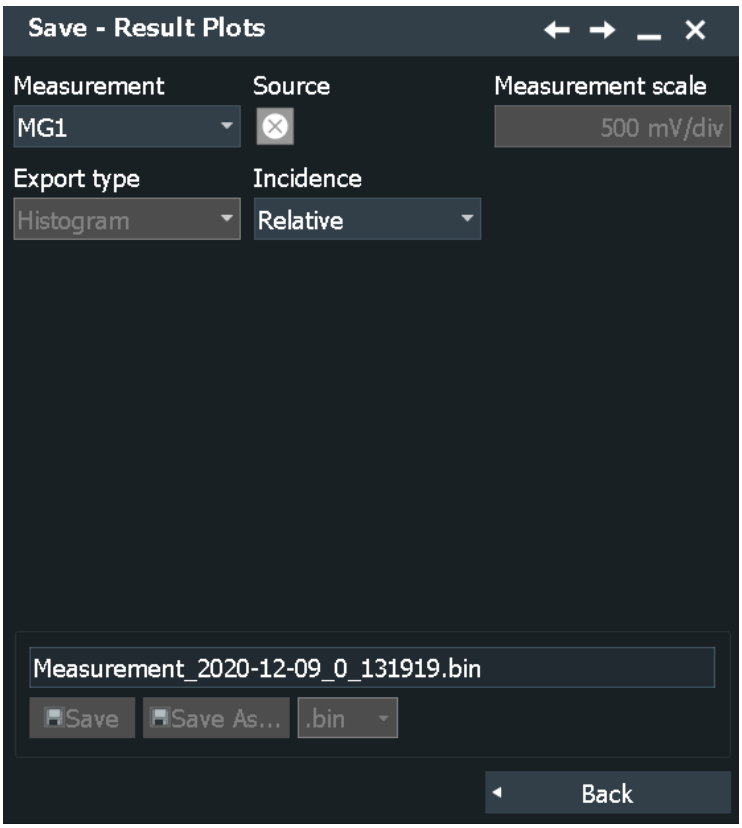
***Data conversion of measurement histogram data:***

Using the header data, you can calculate the measurement value to which a histogram value belongs:

*MeasValue = (XStop - XStart) / 1000 * HistogramValueNumber + XStart*

Example: The 273rd histogram value is 0.491749. That means, the relative frequency of the measurement value 0.1246 V is 0.491749.

| | |
|---|---|
| XStart | 0.07 V |
| XStop | 0.27 V |
| HistogramValueNumber | 273 (number of the row in a CSV file) |
| MeasValue | (0.27 - 0.07) / 1000 * 273 + 0.07 = 0.1246 V |




**Measurement**
Selects the measurement to be exported.

Remote command:
`EXPort:MEASurement:SELect` on page 1625

**Source, Measurement scale**
Displays the measurement settings source and scale.

**Export type**
You can export the result data of the long-term measurement, the measurement histogram or the track.

To export the data, the required type must be enabled in the ""Menu" > "Measure" > Plot" tab: "Long term > Enable", "Histogram > Enable" or "Track > Enable".

Remote command:
`EXPort:MEASurement:TYPE` on page 1625

**Incidence**
Sets the mode of exported histogram data: relative or absolute count of values. If relative values are exported, the sum of all values is 1, and the count of each value is set in relation to the sum.

Remote command:
`EXPort:HISTogram:INCidence` on page 1622

**Save to file**
Enter the filename to save the measurement data to. Double-tap the filename to open the file selection dialog box.

By default, the filename has the prefix "Measurement_".

You can define a pattern for automatic naming in the "Autonaming" tab, see Chapter 4.8.1, "Autonaming", on page 107.

The default directory is:

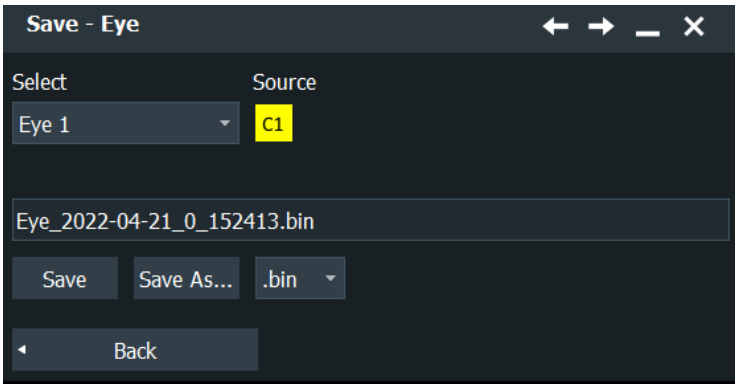
`C:\Users\Public\Documents\Rohde-Schwarz\RTP\Measurements`

| | |
|---|---|
| "Save" | Saves the measurement data in the selected file. |
| "Save As..." | Opens the file selection dialog box and saves the measurement data to the selected file. See also Chapter 12.5, "File selection dialog", on page 513. |
| ".bin/.xml/.csv" | Selects the file format. |

Remote command:
`EXPort:HISTogram:NAME` on page 1622
`EXPort:MEASurement:SAVE` on page 1626
`EXPort:MEASurement:DATA?` on page 1626

## 12.2.6 Eyes

Access: "Menu" key >"Save/Recall" key > "Save" tab > "Eyes".




In this dialog, you can select an eye diagram that you want to export.

Eye export saves the accumulated values at each pixel of an eye diagram.

The export supports *.csv and *bin files. Each export generates a header file and a data file.

**Content and format of the CSV files**

The header file contains the following values:

- Name of the exported eye (e.g. Eye 1)
- Corresponding source (e.g. C1W1)
- Timestamp

- Image height
- Image width
- Pixel format
- XStart
- XStop
- YStart
- YStop

The data file contains the comma-separated list of the pixel values. The number of lines and columns corresponds to the number of lines and rows of the eye diagram image.

**Content and format of the BIN files**

The header file contains the following values:

- Name of the exported eye (e.g. Eye 1)
- Corresponding source (e.g. C1W1)
- Timestamp
- Image height
- Image width
- Pixel format
- XStart
- XStop
- YStart
- YStop

The data file contains the following information in the given order.

- Number of bytes per value - uint32
- Number of samples – uint32
- Column wise sorted pixel values – ‘number of samples’ * ‘number of bytes per value’

**Select/ Source**
Selects the eye measurement to be exported and displays its source.

Remote command:
`EXPort:EYE:SELect` on page 1624
`EXPort:EYE:SOURce?` on page 1624

**Save to file**
Enter the filename to save the eye data to. Double-tap the filename to open the file selection dialog box.

By default, the filename has the prefix "Eye_".

You can define a pattern for automatic naming in the "Autonaming" tab, see Chapter 4.8.1, "Autonaming", on page 107.

The default directory is:

C:\Users\Public\Documents\Rohde-Schwarz\RTP\Eyes

"Save" Saves the measurement data in the selected file.

"Save As..." Opens the file selection dialog box and saves the measurement data to the selected file. See also Chapter 12.5, "File selection dialog", on page 513

".bin/.csv" Selects the file format.

Remote command:
EXPort:EYE:NAME on page 1624
EXPort:EYE:SAVE on page 1624
EXPort:EYE:DATA? on page 1623

### 12.2.7 Saving and loading waveform data

You can save the data of a channel, math or reference waveform to an `.xml`, `.csv`, or `.bin` file. The data export of several channels into one file is also possible. Files in `.bin` format can be reloaded to the R&S RTP as reference waveforms.

Instead of a complete waveform, you can also save a part of it, limited by a previously defined zoom, cursor lines, measurement gate or user-defined time values.

To save waveform data quickly, you can add the "Save Waveform" icon to the toolbar and use it for saving. The icon does not work for saving actions that are started with "Start export" (data logging and multiple history waveforms).

It is also possible to save history data to file. Furthermore, you can save a "live record" of a running Nx Single acquisition to one data file.

For details on waveform save/recall settings, see Chapter 12.2.2, "Waveform settings", on page 490.

The following procedures are described:

- "To save a waveform or a part of a waveform to a file" on page 505
- "To save a waveform using the toolbar icon" on page 506
- "To export waveform data of a running acquisition" on page 506
- "To save the history data" on page 304
- "To load a reference waveform" on page 273
"To save a reference waveform" on page 272

**To save a waveform or a part of a waveform to a file**

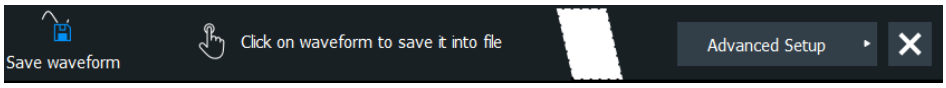
1. Open "Menu" and select "Save/Recall" > "Save".
2. Tap the "Waveform" button.

   The "Save - Waveform" dialog opens.
3. Select the waveforms to be saved:
   - To save one waveform, tap the "Source" icon. Select the waveform.
   - To save data of several channels, enable "Multi channel" and select the channels.

4. In the "Scope" list, select the part of the waveform record to be saved. Zoom, cursor and gate segments require the same setup for the selected waveform before saving. For "Manual", enter the "Start" and "Stop" time of the section.
5. Check the filename under "Save to file" and change it if needed. Usually, autonaming is used.
6. Check the file format and the "Export format" settings and change them if needed.
7. Tap "Save" to save the waveform data to the specified file.
   Tap "Save As" to save the waveform data to a different file or file type. Select the file from the file selection dialog box.

**To save a waveform using the toolbar icon**

1. Add the "Save Waveform" icon to the toolbar, see Chapter 3.3.6.2, "Configuring the toolbar", on page 56.
2. Set the scope, export format, and other parameters in "Menu" > "Save/Recall" > "Waveforms".
3. If necessary, adjust the autonaming pattern and the storage path in "Menu" > "Settings" > "Save/Recall" > "Autonaming".
4. To save the waveform data:
   a) Tap the "Save Waveform" icon on the toolbar. The "Save Waveform" overlay menu opens.




   b) Tap the waveform that you want to save.
   If you tap the diagram background, the data of the focused waveform is saved.

**To export waveform data of a running acquisition**

1. Select the waveform that you want to save and the scope as described in "To save a waveform or a part of a waveform to a file" on page 505, step 1 to 5.
2. If you want to save only a section of each waveform, set the "Scope".
3. Open the "Logger" tab.
4. Tap "Enable".
5. Enter the number of acquisitions to be acquired and saved in "Acquisition count".
6. Check the filename under "Nx single" and change it, if needed. Usually, autonaming is used.
7. Tap "Start export" to start the acquisition and to save the acquired waveform data to the specified file.

**To load waveform data as a reference waveform**

To reload waveform data from a previous measurement, the waveform must have been stored as a reference waveform in a BIN file before.

The procedure is described in Chapter 7.2.1, "Working with reference waveforms", on page 271.

## 12.3 Screenshots

To store the graphical results of the measurement, you can save a screenshot of the graphic area. To document current settings, the open dialog box can be included in the screenshot.

The "Image" toolbar icon saves the current display to a file according to the settings in "Menu" > "Save/Recall" > "Save" tab > "Screenshot". See also "Image" on page 58.

You can configure the [Camera] key to save screenshots by a single keypress. See also Chapter 4.5.1, "Hardkeys: function assignment", on page 97.

If a USB flash drive is connected to the instrument, the default path of the user data directory is set to the USB flash drive. Thus, you save data to USB flash drive automatically, and you can change the directory in the file explorer at any time.

**Screenshots on a computer using the Web interface**

If the R&S RTP is connected to a LAN, you can create and save screenshots of the instrument's display on a computer. See Chapter 23.3.2, "Web browser", on page 1142.

**Meta information in screenshots**

The meta data of the screenshot also contains instrument information. In PNG and JPEG files, meta information is saved as EXIF information and can be read, for example, using the ExifTool.

**Example:**

Reading meta information using the ExifTool.

Command: `# exif C:\Screenshot_2020-07-14_0_110551.png`

Result:

```
ExifTool Version Number         : 10.20
File Name                       : Screenshot_2016-07-14_0_110551.png
Directory                       : C:/
File Size                       : 37 kB
File Modification Date/Time     : 2020:07:14 11:05:51+02:00
File Access Date/Time           : 2020:07:14 11:05:51+02:00
File Creation Date/Time         : 2020:07:14 11:05:51+02:00
...
Instrument Firmware Version     : 3.30.0.46
Instrument Material Number      : 1329.7002k44
Instrument Serial Number        : 123456
Image Size                      : 1280x800
Megapixels                      : 1.0
```

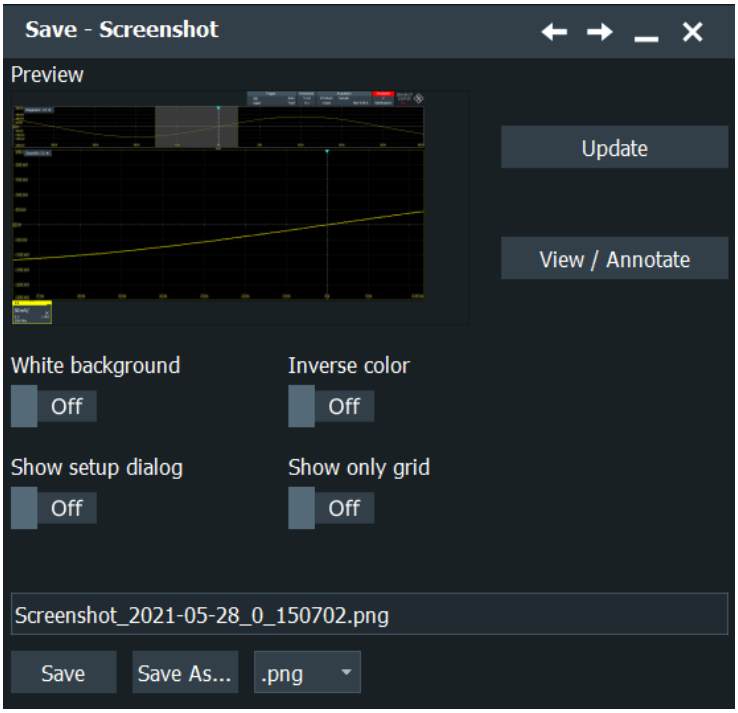
### 12.3.1 Screenshot settings

Access: "Menu" > "Save/Recall" > "Save" tab > "Screenshot"

In the "Screenshot" dialog box, you configure the image to be saved or included in a report. You select the storage location for screenshot files. The image is created when you open the dialog box, and can be updated at any time.

You can also edit the image before saving, and include an open dialog box or the sidebar in the image.

You can save the image in the "Screenshot" dialog box. To save screenshots quickly, use the "Image" toolbar icon, or configure and use the [Camera] key.




Preview ..... 509
Update ..... 509
White background ..... 509
Inverse color ..... 510
Show setup dialog ..... 510
Show only grid ..... 510
View/ Annotate ..... 510
Save to file ..... 510

**Preview**
Shows a preview of the screenshot. The image is created when the dialog box opens.

**Update**
Updates the preview of the screenshot with the current display view, e.g. after changes to the settings have been made, or an additional channel has been activated.

**White background**
Inverts the background color. So you can picture waveforms with normal waveform colors on white background.

If both "White background" and Inverse color are enabled, the instrument inverts the background twice, and it appears black.

| "White background" | "Inverse color" | Background | Waveform and results |
|---|---|---|---|
| On | Off | White | Screen colors |
| Off | On | White | Inverted colors |

| "White background" | "Inverse color" | Background | Waveform and results |
|---|---|---|---|
| On | On | Black | Inverted colors |
| Off | Off | Black | Screen colors |

Remote command:
HCOPy:WBKG on page 1629

**Inverse color**
Inverts the colors of the output, i.e. a dark waveform is shown on a white background.

See also: "White background" on page 509.

Remote command:
HCOPy:DEVice<m>:INVerse on page 1628

**Show setup dialog**
If you want to save dialog boxes in screenshots, enable "Show setup dialog". The currently open dialog box is included in the screenshot.

Remote command:
HCOPy:SSD on page 1629

**Show only grid**
If enabled, the screenshot shows only the grid. The signal bar is not included.

Remote command:
HCOPy:ISBA on page 1630

**View/ Annotate**
Opens the screenshot in the Paint application. Edit the image as necessary. You can store the file using "Save as". Alternatively, save the file and close the Paint application to return to the "Screenshot" dialog, then save the edited image. The changes are not shown in the preview.

**Save to file**
Defines the filename to which screenshot is saved. By default, the filename has the prefix "Screenshot_". Double-tap the filename filed to change the name.

If a USB flash drive is connected to the instrument, the default path of the user data directory is set to the USB flash drive. Thus, you save data to USB flash drive automatically, and you can change the directory in the file explorer at any time.

"Save" Saves the current screenshot to the specified file.

"Save As..." Opens the file selection dialog box. Here you can adjust the target directory and the file name and save the current screenshot to the file. The symbols of important target folders are listed on the left of the file explorer.

"Delete" Opens the file selection dialog box and deletes the selected file.

Remote command:
HCOPy:DEVice<m>:LANGuage on page 1628
HCOPy:DESTination<1..2> on page 1627

MMEMory:NAME on page 1627
HCOPy:IMMediate<m>[:DUM] on page 1630
HCOPy:IMMediate<m>:NEXT on page 1630
MMEMory:DELete on page 1609

### 12.3.2 Configuring and saving screenshots

You can edit the image, invert all colors, and set the background color. A preview of the current image is shown for reference.

1. Open the "Menu" > "Save/Recall" > "Save" tab > "Screenshot".
2. To enhance the images for later print on white paper, enable "White background" or "Inverse color". If you print this image later on a monochrome printer, you get a grayscaled picture. The contrast of the resulting gray lines depends on waveform colors and the used printer.
3. Select the file format: png, jpg, or another one.
4. To change the directory, tap "Save As" and configure the path.

   The symbols of often used target folders are listed on the left of the file explorer. By default, screenshots are saved in the `C:\Users\Public\Public Documents\Rohde-Schwarz\RTx\ScreenShots` directory.
5. Tap "Save".

   The file is saved and the dialog box closes.
6. Check if the screenshot is saved to the desired directory.
7. To save further screenshots, use one of the following ways:
   - Configure the [Camera] key. Press the key to save a screenshot.
     See also Chapter 4.5.1, "Hardkeys: function assignment", on page 97.
   - Add the "Image" icon to the toolbar. Tap the icon to save an image.
     See also Chapter 3.3.6.2, "Configuring the toolbar", on page 56.
   - Tap "Save" in the "Screenshot" dialog box to save the image to the specified file.
   - To save the image with a dedicated filename or to another directory, tap "Save As" in the "Screenshot" dialog box. Select the path, enter a filename, and tap "Save".

## 12.4 Reports

Reports document the current measurement and test results. The report contains general information, current vertical and horizontal settings, trigger settings, active chan-

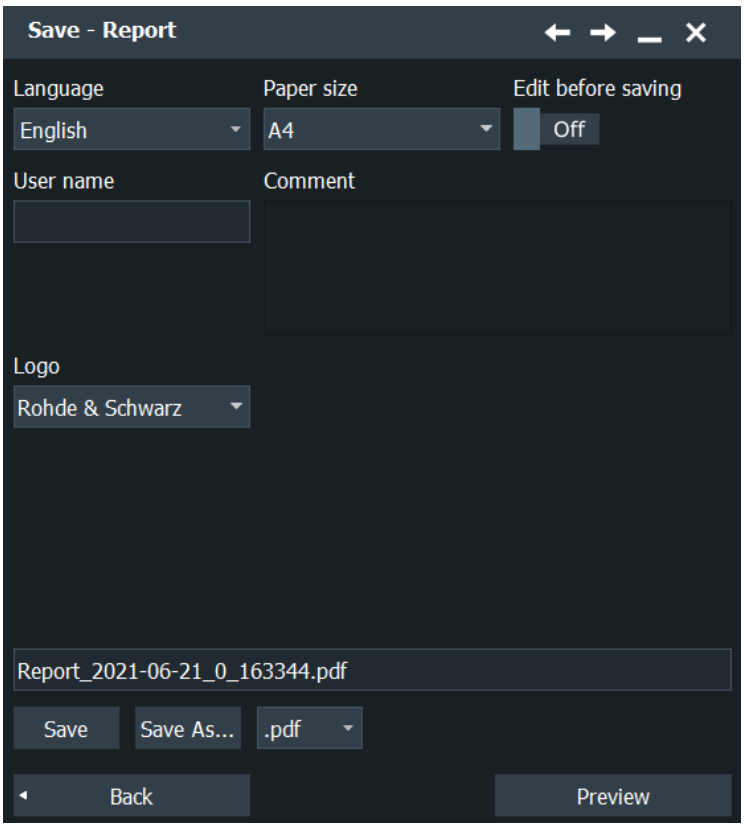
nels and all current results except for zoom and search results. A screenshot is also included.

You can create the report manually, or automatically on defined events:

- Press the [Camera] key.
- Tap the "Create report" toolbar icon.
  Before, add the icon to the toolbar, see also Chapter 3.3.6.2, "Configuring the toolbar", on page 56.
- Action on micro button, available on active Rohde & Schwarz probes
- Action on trigger
- Event action at mask testing
- Event action at limit checks

### 12.4.1 Report settings

Access: "Menu" menu > "Save/Recall" menu > "Save" tab > "Report"




**Language**
Selects the language to be used in the report. Available languages are listed in the data sheet.

Remote command:
`REPort:LANGuage` on page 1631

**Paper size**
Selects the paper size: A4 or US Letter.

Remote command:
REPort:PAPersize on page 1631

**Edit before saving**
Enables you to edit the report info when you save reports using the [Camera] key. When you press the key, a dialog box opens where you can change the user name and the comment.

**User name / Comment**
Enter information that appears in the general information section at the beginning of the report.

Remote command:
REPort:USER on page 1632
REPort:COMMent on page 1632

**Logo**
By default, the Rohde & Schwarz logo is shown in the header of the report pages. You can switch the logo off, or select your logo to be shown. A preview of the selected log file is shown.

Remote command:
REPort:LOGType on page 1631
REPort:LOGO on page 1632

**Save to file**
Select the file format and define the filename of the report file. By default, the filename has the prefix "Report_".

Double-tap the filename to open the file selection dialog box.

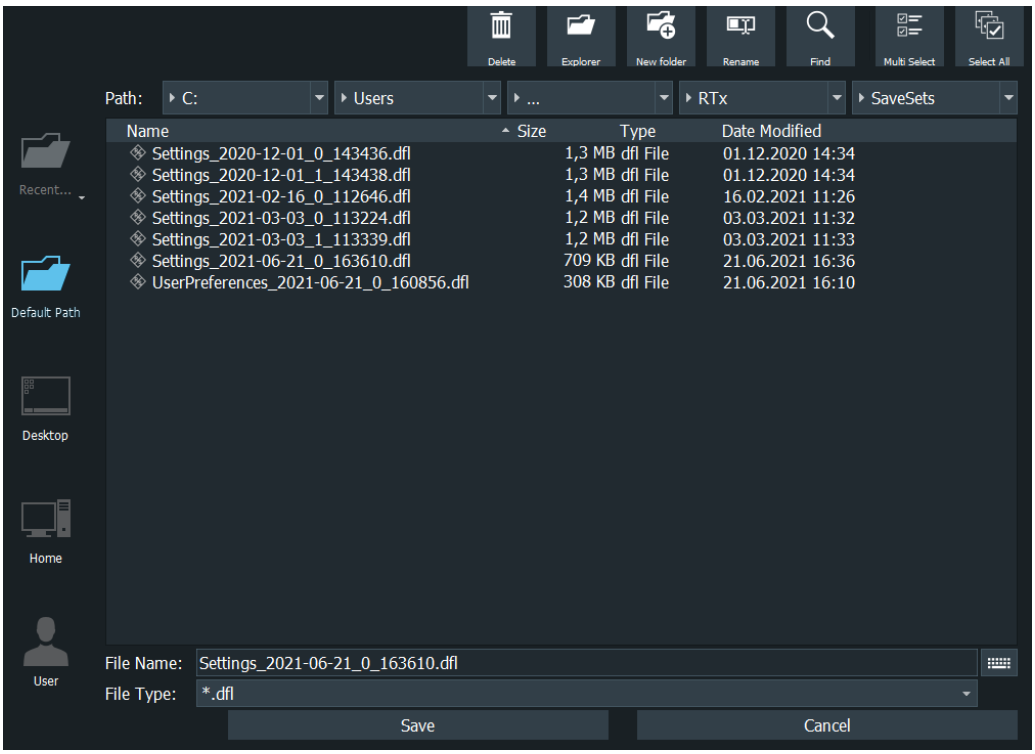
| | |
|---|---|
| "pdf/doc/html" | Selects the report format. |
| "Save" | Saves the current report to the specified file. |
| "Save As..." | Opens the file selection dialog box and saves the report to the selected file. |

Remote command:
REPort:FILE:NAME on page 1632
REPort:FILE:SAVE on page 1633

**Preview**
Opens the current report in PDF format.

## 12.5 File selection dialog

The file selection dialog provides a file explorer from which you can select a file to load or to save data to. You can also manage your files in this dialog.




**Path**
Tap the path elements to change the current folder. The default folder is defined in Default path for all file operations.

You can save the data in a local folder on the instrument, to an external storage device (usually a USB flash drive), or to a folder on a connected network drive. The path list provides all available drives and folders.

On the left, shortcut icons provide access to often used folders.

**Toolbar**
The toolbar on the top provides various functions for file management.

**File Name**
The file name to be loaded or stored to. Double-tap the file name, or tap the keyboard icon to enter the file name using the online keyboard.

The default file name for new files is defined in the "Autonaming" tab, see Chapter 4.8.1, "Autonaming", on page 107.

**File Type**
The file extension of the file to be loaded or stored to.

**Save, Select**
Selects the specified file for the open or save operation and closes the dialog box.

**Cancel**
Closes the dialog box without selecting a file.

# 13 Protocol analysis

Using the serial protocol options for the R&S RTP, you can analyze various serial protocols.


- Basics of protocol analysis....515
- I²C (option R&S RTP-K1)....525
- SPI bus (option R&S RTP-K1)....539
- UART/RS-232/RS-422/RS-485 (option R&S RTP-K2)....550
- CAN and CAN FD (option R&S RTP-K3/K9)....559
- LIN (option R&S RTP-K3)....585
- MIL-1553 (option R&S RTP-K6)....599
- ARINC 429 (option R&S RTP-K7)....616
- Ethernet 10BASE-T and 100BASE-TX (option R&S RTP-K8)....629
- Ethernet 100BASE-T1 (option R&S RTP-57)....644
- Ethernet 1000BASE-T1 (option R&S RTP-K58)....663
- RFFE (option R&S RTP-K40)....681
- D-PHY (option R&S RTP-K42)....702
- M-PHY and USB SSIC (option R&S RTP-K44 and K64)....718
- Custom: Manchester / NRZ (option R&S RTP-K50)....748
- 8b/10b (option R&S RTP-K52)....781
- MDIO (option R&S RTP-K55)....794
- USB (option R&S RTP-K60)....807
- USB 3.1 (option R&S RTP-K61/K62)....834
- USBPD (option R&S RTP-K63)....856
- SpaceWire (option R&S RTP-K65)....869
- PCIe (option R&S RTP-K72/K73)....881
- DDR (option R&S RTP-K91/K93 )....916


## 13.1 Basics of protocol analysis

Before you can analyze a serial signal, the bus has to be configured according to the protocol and specifics of the signal. The configuration contains:

- Assignment of the data and clock lines to the input channels
- Logical thresholds
- Protocol-specific settings

Serial data can be analyzed in several ways:

- Triggering: You can trigger on various events that are typical for the selected protocol type, for example, on start and stop of messages, or on specified data patterns in the message.
  Triggering on a trigger event sequence is not supported, and holdoff settings are not available.
- Protocol decoding: The digitized signal data is displayed on the screen together with the decoded content of the messages in readable form, and the decode results are listed in a table.

- Search on decoded signal data: For most serial protocols, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, because several event types can be combined. Thus, you get the results for the complete acquisition cycle.

Restrictions:

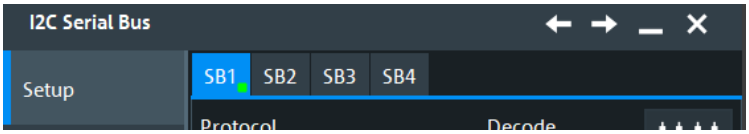
- Protocol analysis does not work if the "Decimation" is set to "High res".
- All serial protocols are disabled in roll mode.

### 13.1.1 Setup - general settings

For all protocols, configuration starts with the selection of the serial bus and the protocol.

***Table 13-1: Configuration settings are protocol-specific. They are described in the related chapters:***

| | |
|---|---|
| ARINC 429 | Chapter 13.8.2, "ARINC 429 configuration", on page 617 |
| CAN, CAN FD | Chapter 13.5.2, "CAN and CAN FD configuration", on page 559 |
| Custom: Manchester / NRZ | Chapter 13.15.2, "Custom: Manchester / NRZ configuration", on page 750 |
| Ethernet 10BASE-T and 100BASE-TX | Chapter 13.9.2, "Ethernet configuration", on page 629 |
| Ethernet 100BASE-T1 | Chapter 13.10.2, "100BASE-T1 configuration", on page 646 |
| Ethernet 1000BASE-T1 | Chapter 13.11.2, "1000BASE-T1 configuration", on page 663 |
| I²C | Chapter 13.2.2, "I²C configuration", on page 527 |
| LIN | Chapter 13.6.2, "LIN configuration", on page 587 |
| MDIO | Chapter 13.17.2, "MDIO configuration", on page 796 |
| MIL-1553 | Chapter 13.7.2, "MIL-STD-1553 configuration", on page 602 |
| MIPI D-PHY | Chapter 13.13.2, "D-PHY configuration", on page 703 |
| MIPI M-PHY | Chapter 13.14, "M-PHY and USB SSIC (option R&S RTP-K44 and K64)", on page 718 |
| MIPI RFFE | Chapter 13.12.2, "RFFE configuration", on page 683 |
| SpaceWire | Chapter 13.21.2, "SpaceWire configuration", on page 871 |
| SPI | Chapter 13.3.2, "SPI configuration", on page 540 |
| UART | Chapter 13.4.2.1, "UART configuration settings", on page 551 |
| USB | Chapter 13.18.2, "USB 2.0 configuration", on page 813 |
| USB 3.1 | Chapter 13.19.2, "USB 3.1 configuration", on page 843 |
| USB PD | Chapter 13.20.2, "USBPD configuration", on page 858 |
| USB SSIC | Chapter 13.14.2, "M-PHY configuration", on page 720 |
| 8b/10b | Chapter 13.16.2, "8b/10b configuration", on page 782 |
| DDR3/DDR4 | Chapter 13.23.1, "DDR configuration", on page 916 |




Make sure that the tab of the correct serial bus is selected.

**Protocol**
Defines the protocol type of the selected serial bus.

Remote command:
`BUS<m>:TYPE` on page 1637

**Decode**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
`BUS<m>[:STATe]` on page 1637

### 13.1.2 Autoset for protocols

If you want to perform a quick protocol measurement, you can use the "Autoset" function. It allows you to find the correct instrument settings for the desired protocol. It displays at least one decoded frame after the execution.

**Performing a protocol autoset**

1. Press [Bus] on the front panel.
2. Select the bus you want to configure.
3. Select the "Setup" tab.
4. Tap the "Protocol" you want to configure.
5. Assign the sources for the input channels.
6. Press the "Autoset" button.

"Autoset" performs the following steps:

- Executes "Autoset" for the horizontal and vertical scale.
- Adjusts the horizontal scale to display at least one frame or packet .
- Executes "Find thresholds" to determine the thresholds.
- If necessary, performs bit rate estimation.
- Sets default protocol trigger (frame start) .
- Turns on the protocol decoder.

Remote command:

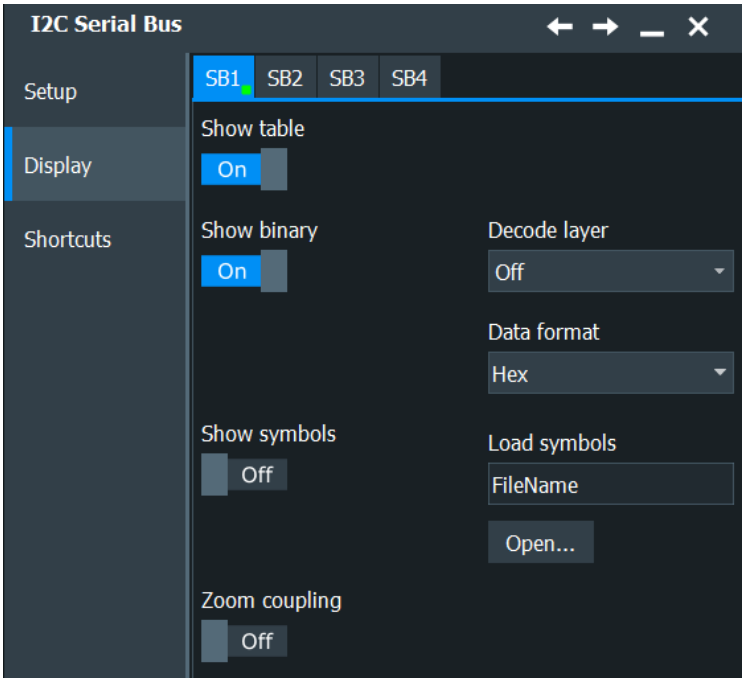
BUS<m>:FAUToset on page 1638

There are some limitations for the autoset:

- It is not available, if the digital MSO channels are selected as a source for the waveform.
- Signals with longer idle phases might not find horizontal adjustment.
- On duplex signals, different forward/reverse amplitudes can fail to determine thresholds.

### 13.1.3 Display

For all protocols, you can select to display the decoded signal as a table and to show the binary signal on the screen.




For some protocols, the result table provides a button to show the details of the selected frame.

#### Label Lists

For all protocols using ID or address identification, it is possible to create label lists containing addresses or IDs, a symbolic name for each node (symbolic label), and some protocol-specific information.

You can load label lists, and activate its usage for decoding. As a result, an additional "Label" column appears in the "Decode results" table, containing the symbolic label. The frame captions of the decoded signal show the symbolic label instead of the ID or address values. Hence it is easy to identify the messages of the different bus nodes.

You can also use the label list to trigger on an identifier or address. Instead of entering the value, you select the name, which is defined in the label list.

**Show table**
Opens a table with decoded data of the serial signal. The function requires the option for the analyzed protocol.

Decode results are protocol-specific. They are described in the "Decode Results" chapter of each protocol.

Remote command:
`BUS<m>:RESult` on page 1638

**Show binary**
For each configured line, the binary signal is also displayed.

**Data format**
Sets the data format for the values displayed in the decode table and in the combs of the decoded signal.

Remote command:
`BUS<m>:FORMat` on page 1639

**Show symbols**
Activates the symbols list to be used for decoding. The "Label" appear in the "Decode results" table and in the frame captions of the decoded signal.

Remote command:
`BUS<m>:SYMBols` on page 1639

**Load symbols**
Selects and loads a label list file. Available file formats are PTT, CSV, DBC (CAN only), and XML (FIBEX files, FlexRay only).

Label lists are protocol-specific. Their contents are described in the corresponding protocol chapters.

Remote command:
`BUS<m>:NEWList` on page 1639

**Zoom coupling**
If enabled, the decode zoom and result table are synchronized. If you select a row in the result table, this result is shown in the decode zoom.

With "Zoom window", you can also select the zoom used for coupling.

For an introduction to zoom settings and setup, see:
- Chapter 7.1.2, "Zoom settings", on page 263
- Chapter 7.1.3, "Zooming for details", on page 267

Remote command:
`BUS<m>:ZCOupling` on page 1640

### 13.1.4 Content and format of the PTT file

Label lists are stored as PTT (protocol translation table) files. The PTT file format is an extension of the CSV format (comma-separated values). You can edit it with standard editors, for example, with MS Excel or a text editor.

The PTT file has three types of lines:

- Comment lines begin with a hash character #. A hash character at any other position in the line is treated like a standard character.
- Command lines begin with a commercial at character @. An @ character at any other position in the line is treated like a standard character.
- Standard lines are the lines that not qualify as comment or command lines. They build the core of the label list.

#### Command lines

Command lines define the version of the PTT file and the protocol name:

- @FILE_VERSION: must appear exactly once in the file
- @PROTOCOL_NAME: must appear at least once in the file. Thus, one file can contain several label lists for different protocols.

```
# --- Start of PTT file
@FILE_VERSION  = 1.0
@PROTOCOL_NAME = i2c
[... Label list for I2C]
@PROTOCOL_NAME = can
[... Label list for CAN]
# --- End of PTT file
```

#### Standard lines

Standard lines define the contents of the label list. The rules for standard lines follow the csv convention, they are:

- Values are separated by commas
- Space characters following a delimiter are ignored
- Values with a special character (comma, newline, or double quote) must be enclosed in double quotes
- Text in double quotes must be escaped by double quote characters

The format of the numeric value is indicated by a suffix. The following formats are supported:

| Format | Suffix | Example |
|---|---|---|
| Decimal | <empty><br>d | `106, DeviceName`<br>`106d, DeviceName` |
| Hexadecimal | h | `6Ah, DeviceName`<br>or prefix: `0x6A, DeviceName` |

| Format | Suffix | Example |
|---|---|---|
| Octal | o | `152o, DeviceName` |
| Binary | b | `01101010b, DeviceName` |

The maximum supported word size for (unsigned) integers is 64 bits.

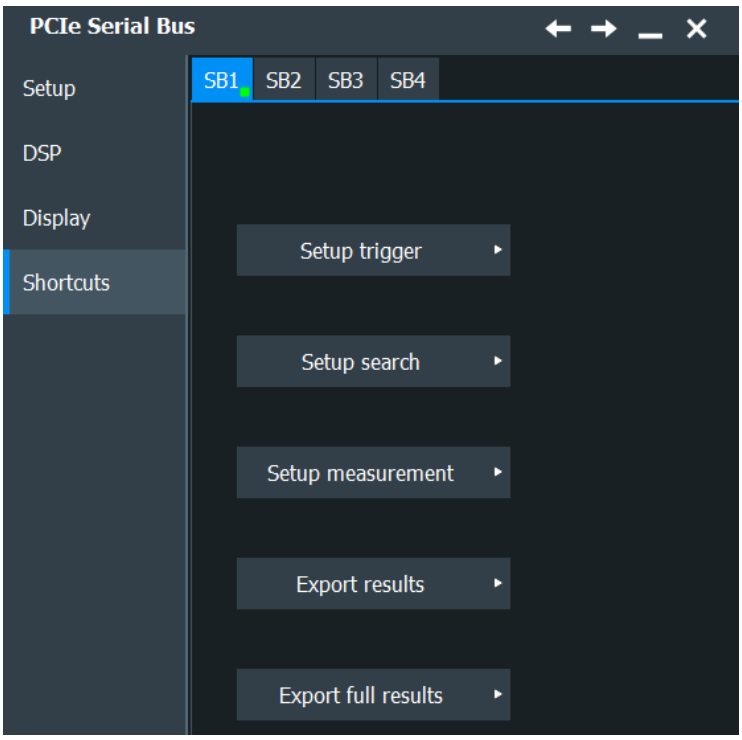
```
# --- Start of PTT file
@FILE_VERSION  = 1.0
@PROTOCOL_NAME = i2c
#    Following two lines are equal:
7,01h,Temperature
7,01h, Temperature
#    A comma must be enclosed in double quotes:
7,01h,"Temperature, Pressure, and Volume"
#    A double quote must also be enclosed in double quotes:
7,7Fh,"Highspeed ""Master"" 01"
#    Following lines yield the same result:
7d,0x11,Pressure
7h,11h,Pressure
0x7,17d,Pressure
7,17,Pressure
```

### 13.1.5 Shortcuts

The "Shortcuts" tab gives quick access to other dialogs with protocol relevant settings. The availability depends on the functions supported by the protocol.

If you access a dialog from the "Shortcuts" tab, some settings are already predefined.

- "Trigger" dialog: you can trigger on various events that are typical for the selected protocol type, for example, on start and stop of messages, or on specified data patterns in the message.
- "Search" dialog: here you can configure protocol-specific search settings.
- In the "Measurements" dialog, you can perform additional measurements for some protocols. See Chapter 8.2.11, "Protocol measurements (option R&S RTP-K35)", on page 378.
- "Export results": in the "Save Results" dialog, you can select the decode results you want to export, the file format, and the delimiter.
  For details, see Chapter 12.2.4, "Results", on page 499.
- "Export full results": in this dialog, you can select the decode results you want to export and the protocol-specific details to be included.

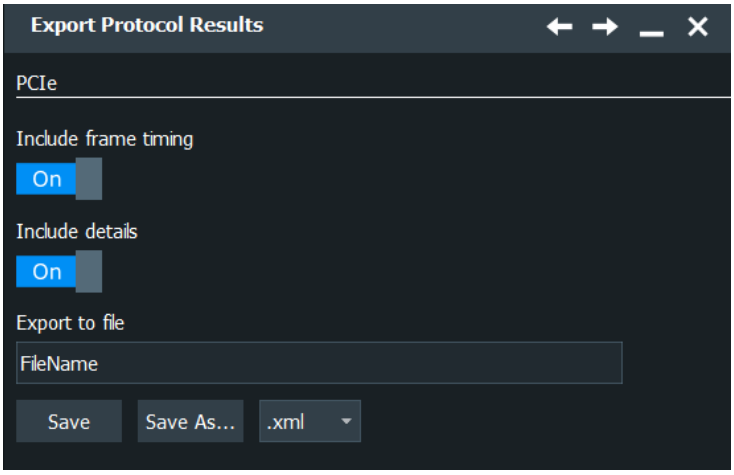



### 13.1.6 Export protocol results

In the "Export Protocol Results" dialog, you can export the results and all details of the selected protocol.

This is availble for the following protocols:

- ARINC 429
- CAN, CAN FD
- USB 3.1
- Ethernet 10BASE-T and 100BASE-TX
- Ethernet 100BASE-T1
- Ethernet 1000BASE-T1
- LIN
- MIL-1553
- MIPI D-PHY
- SPI
- UART




**Include frame timing**
Includes the frame timing.

Remote command:
`BUS<m>:EXPResult:TIME` on page 1641

**Include details**
Includes the detailed results for all frames.

Remote command:
`BUS<m>:EXPResult:DETail` on page 1640

**Export to file**
Enter the filename to save the results to. Double-tap the filename to open the file selection dialog box.

"Save" Saves the selected results to the indicated file.

"Save As..." Opens the file selection dialog box and saves the selected results to the selected file. See also Chapter 12.5, "File selection dialog", on page 513

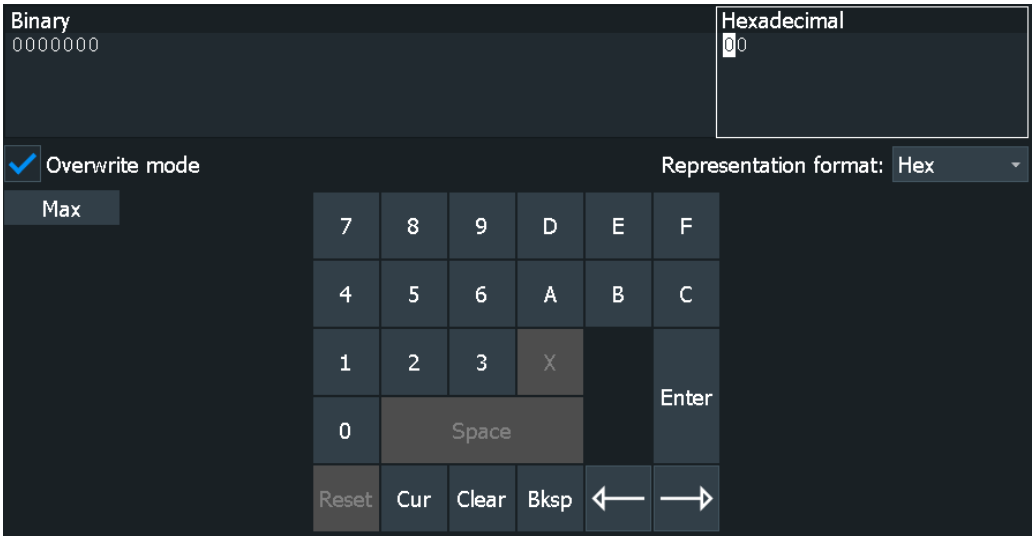
File format Selects the file format.

- ".csv": the values are saved in a file.
  You can select the value delimiter and the list separator symbol in the "CSV Export" dialog, see Chapter 4.8.2, "CSV export", on page 110.
  Tip for using MS Excel: It is recommended that you use the semicolon as csv decimal symbol. When you open the file with MS Excel, use "File > Open" and follow the wizard to set the separators correctly, or set the separator settings with "Tools > Options > International".
- ".html": the results are saved as web page for display in a browser.
- ".xml": the results are saved in an xml compatible file format.
- ".py": the values are saved in a python compatible file format.

Remote command:
`BUS<m>:EXPResult:SAVE` on page 1640

### 13.1.7 Bit pattern editor

If you want to enter a specified address or data pattern, the bit pattern editor helps you to enter the pattern in various formats - decimal, hexadecimal, octal, binary and ASCII.




The editor displays the pattern in two columns. The left column always shows binary data. For the right column, you can select the format, the default depends on the data specifics. You can edit data in the left or right column. The keypad adapts itself to the column format, only keys appropriate to the format are enabled.

The data is grouped and converted in bit groups. The size of a bit group depends on the address or data specifics and is set by the instrument. Groups are automatically separated by blanks. The maximum size of a bit group is 64 bit, the most common group size is 1 byte.

"Overwrite mode": If disabled, the data behind the new digit is shifted to the right. Bit groups are rearranged automatically.

Format-specific information:

- Unsigned: Decimal data format without sign. It is available for I²C, SPI, UART, CAN, LIN and FlexRay protocols. If you enter a decimal number that is too large for the defined bit group, the number is truncated and a message appears. X (do not care) in the decimal column sets all binary digits of the bit group to X.
- Signed: Signed decimal format, available for audio protocols. The first bit represents the sign. You can use the 2's complement or 1's complement format.
- Binary: 0, 1 and X (do not care) is allowed.
- Hex: most common format in the right column.
- Octal: Each digit represents 3 bit.
- ASCII: In the ASCII column, "X" is the character X. The binary X (do not care) is not allowed. If an X is included in the binary value in the left column, the ASCII column displays "§" to indicate that the value is not defined.

Where applicable, frequently used values are provided in a "Predefined values" list below the pattern table, for example, reserved end words of data packets in the UART protocol.

## 13.2 I²C (option R&S RTP-K1)

The Inter-Integrated Circuit is a simple, low-bandwidth, low-speed protocol used for communication between on-board devices.


- The I²C protocol ..... 525
- I²C configuration ..... 527
- I²C trigger ..... 530
- I²C decode results ..... 535
- Search on decoded I²C data ..... 537


### 13.2.1 The I²C protocol

This chapter provides an overview of protocol characteristics, data format, address types and trigger possibilities. For detailed information, read the "I2C-bus specification and user manual" available on the NXP manuals webpage at http://www.nxp.com/.

#### I²C characteristics

Main characteristics of I²C are:

- Two-wire design: serial clock (SCL) and serial data (SDA) lines
- Master-slave communication: the master generates the clock and addresses the slaves. Slaves receive the address and the clock. Both master and slaves can transmit and receive data.
- Addressing scheme: each slave device is addressable by a unique address. Multiple slave devices can be linked together and can be addressed by the same master.
- Read/write bit: specifies if the master reads (=1) or writes (=0) the data.
- Acknowledge: takes place after every byte. The receiver of the address or data sends the acknowledge bit to the transmitter.

The R&S RTP supports all operating speed modes: high-speed, fast mode plus, fast mode, and standard mode.

#### Data transfer

The format of a simple I²C message (frame) with 7-bit addressing consists of the following parts:

- Start condition: a falling slope on SDA while SCL is high
- 7-bit address of the slave device that either is written to or read from
- R/W bit: specifies if the data is written to or read from the slave

- ACKnowledge bits: is issued by the receiver of the previous byte if the transfer was successful
  Exception: At read access, the master terminates the data transmission with a NACK bit after the last byte.
- Data: several data bytes with an ACK bit after every byte
- Stop condition: a rising slope on SDA while SCL is high

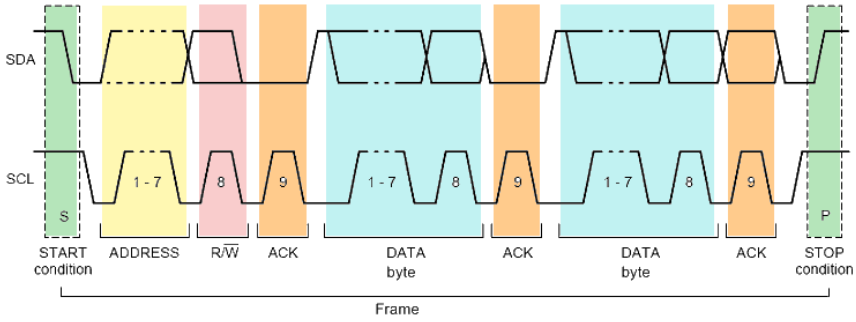



*Figure 13-1: I2C writes access with 7-bit address*

### Address types: 7-bit and 10-bit

Slave addresses can be 7 bits or 10 bits long. A 7-bit address requires 1 byte, 7 bits for the address followed by the R/W bit.

A 10-bit address for write access requires 2 bytes: the first byte starts with the reserved sequence 11110, followed by the two MSB of the address and the write bit. The second byte contains the remaining 8 LSB of the address. The slave acknowledges each address byte.

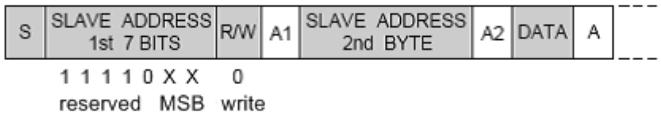



*Figure 13-2: 10-bit address, write access*

A 10-bit address for read access requires 3 bytes. The first 2 bytes are identical to the write access address. The third byte repeats the address bits of the first byte and sets the read bit.

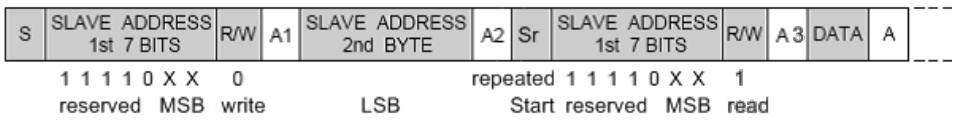



*Figure 13-3: 10-bit address, read access*

### Trigger

The R&S RTP can trigger on various parts of I²C messages. The data and clock lines must be connected to the input channels, triggering on math and reference waveforms is not possible.

You can trigger on:

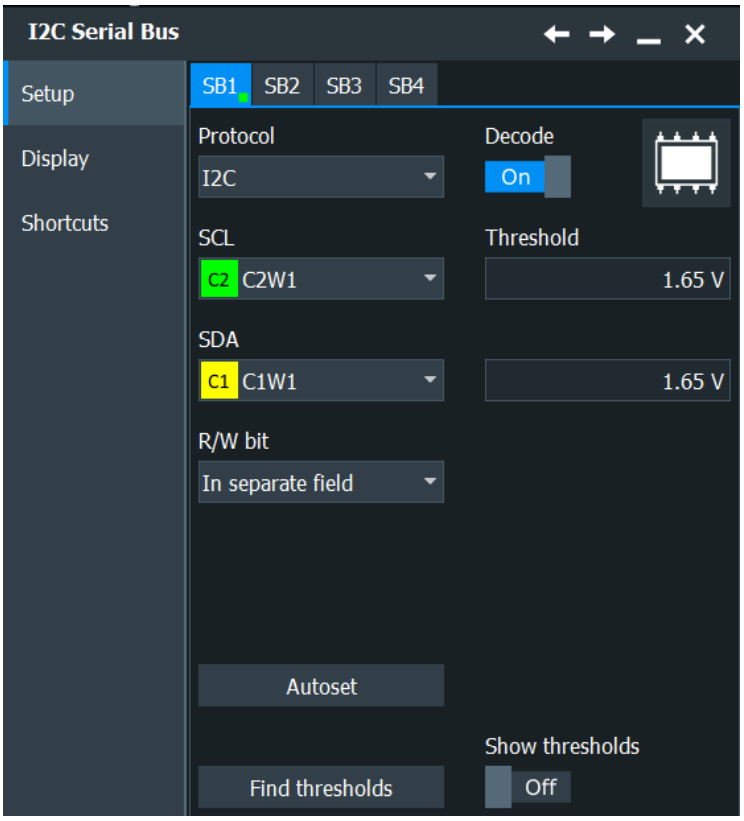
- Start or stop condition
- Repeated start condition
- Transfer direction (read or write)
- Bytes with missing acknowledge bit
- Specific slave address or address range
- Specific data pattern in the message

## 13.2.2 I²C configuration

### 13.2.2.1 I²C configuration settings

Access: [Bus] > "I2C" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**SDA, SCL**
Set the waveforms of the data line (SDA) and clock line (SCL).

Usually, the source is one of the analog channels. Reference and math waveforms are available as source if the trigger source is one of the analog channels but not the serial bus.

Do not combine a reference waveform with channel or math waveform because the time correlation of these waveforms might differ.

Alternatively, digital channels can be used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.

For triggering on a serial bus, analog or digital channel sources are required.

Remote command:
`BUS<m>:I2C:SDA:SOURce` on page 1644
`BUS<m>:I2C:SCL:SOURce` on page 1643

**R/W bit**
Defines if the R/W bit is considered separately or as part of the address. The setting affects the Adress Type of the trigger conditions.

Remote command:
`BUS<m>:I2C:RWBit` on page 1645

**Threshold**
Sets the threshold value for digitization of signals for each line. If the signal value on the line is higher than the threshold, the signal state is high (1 or true for the Boolean logic). Otherwise, the signal state is considered low (0 or false) if the signal value is below the threshold.

There are several ways to set the threshold:
- "Threshold"
  Enter the value directly in the field.
- "Autoset"
  Starts software algorithms for determining the signal threshold levels and bitrate.
- "Find thresholds"
  Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

**Note:** If the sources are digital channels, the same threshold values are used for the parallel and the serial buses. You can set the thresholds either in the parallel bus configuration or in the serial bus configuration.

Remote command:
`BUS<m>:I2C:SCL:THReshold` on page 1644
`BUS<m>:I2C:SDA:THReshold` on page 1645
`BUS<m>:I2C:TECHnology` on page 1645
`BUS<m>:SETReflevels` on page 1638
`BUS<m>:FAUToset` on page 1638

**Show thresholds**
If enabled, the threshold levels are displayed in the diagram.

Remote command:
BUS<m>:THReshold on page 1638

#### 13.2.2.2 Display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

**I2C Specific Label Settings**

Label lists are protocol-specific. Label lists for I²C are available in CSV and PTT format.

An I²C label file contains three values for each address:

- Address type, 7-bit or 10-bit long
- Address value
- Symbolic label: name of the address, specifying its function in the bus network.

**Example: I²C PTT file**

```
# ------------------------------------------------------------------------
@FILE_VERSION  = 1.00
@PROTOCOL_NAME = i2c
# ------------------------------------------------------------------------
# Labels for I2C protocol
#   Column order: Identifier type, Identifier value, Label
# ------------------------------------------------------------------------
7,0x1E,Voltage
7,38h,Pressure
7,2Ah,Temperature
7,16h,Speed
7,118,Acceleration
7,07h,HighSpeed_Master_0x3
7,51h,EEPROM
10,3A2h,DeviceSetup
10,1A3h,GatewayStatus
10,06Eh,LeftSensor
# ------------------------------------------------------------------------
```

#### 13.2.2.3 Configuring I²C protocol

The configuration of the I²C is simple - assign the two lines to input channels, and set the thresholds.

For details on configuration settings, see Chapter 13.2.2, "I²C configuration", on page 527.

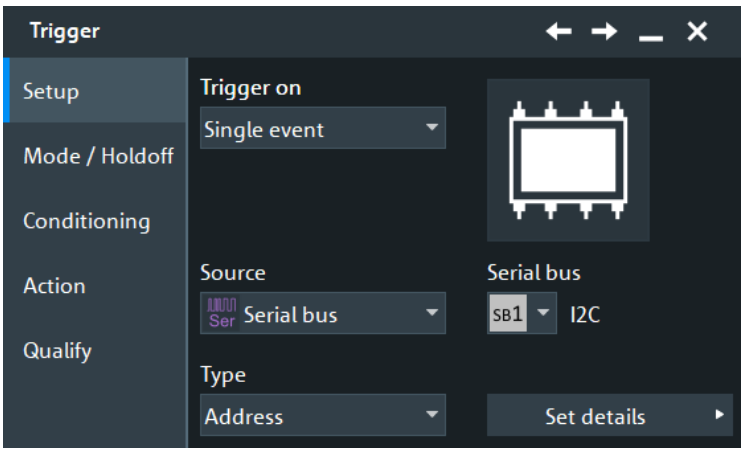
1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".

3. Select the "Setup" tab.
4. Tap the "Protocol" button and select the protocol: "I2C".
5. Tap the "SDA" button, and select the waveform of the data line.
6. Tap the "SCL" button, and select the waveform of the clock line.
7. Set the logical thresholds.
8. Enable "Decode".

## 13.2.3 I²C trigger

### 13.2.3.1 I²C trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"




Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Serial bus**
Selects the serial bus to be triggered on. Make sure to select the correct bus before you enter the settings.

To trigger on a serial bus, the signals sources must be channel signals. If the data or clock source is a math or reference waveform, you cannot trigger on that bus.

Remote command:
TRIGger<m>:SOURce:SBSelect on page 1642

**Trigger type**
Selects the trigger type for I²C analysis.

Some trigger types have additional settings that can be defined. In this case, the "Define trigger details" button appears next to the "Type" function. Open it for a detailed definition of the trigger conditions.

Remote command:
TRIGger<m>:I2C:TYPE on page 1647

**Start ← Trigger type**
Sets the trigger to the start of the message. The start condition is a falling edge on SDA while SCL is high. The trigger instant is the falling edge of the SDA line.

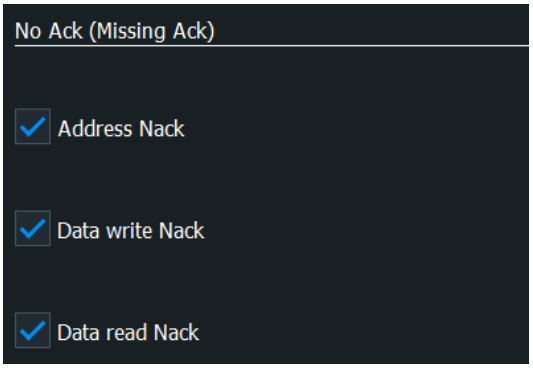
You can change the SDA and SCL lines here if necessary.

**Repeated start ← Trigger type**
Sets the trigger to a repeated start - when the start condition occurs without previous stop condition. Repeated start conditions occur when a primary exchanges multiple messages with a secondary without releasing the bus.

**Stop ← Trigger type**
Sets the trigger to the end of the message. The stop condition is a rising slope on SDA while SCL is high.

**No Ack (Missing Ack) ← Trigger type**
Missing acknowledge bit: the instrument triggers if the data line remains HIGH during the clock pulse following a transmitted byte.

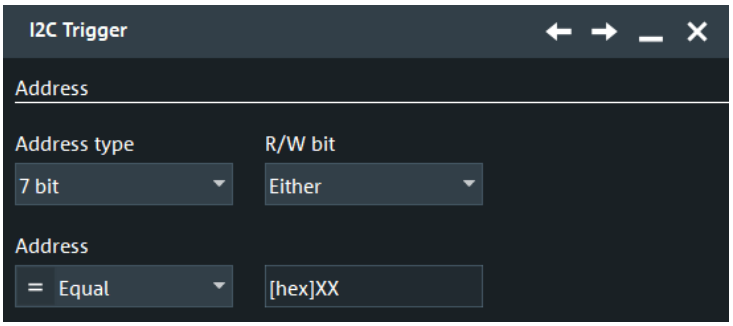
You can also localize specific missing acknowledge bits by setting the No Ack conditions.




**Address ← Trigger type**
Sets the trigger to one specific address condition or a combination of address conditions. The trigger time is the falling clock edge of the acknowledge bit after the address. You can specify:

- Adress Type
- "Address" on page 533

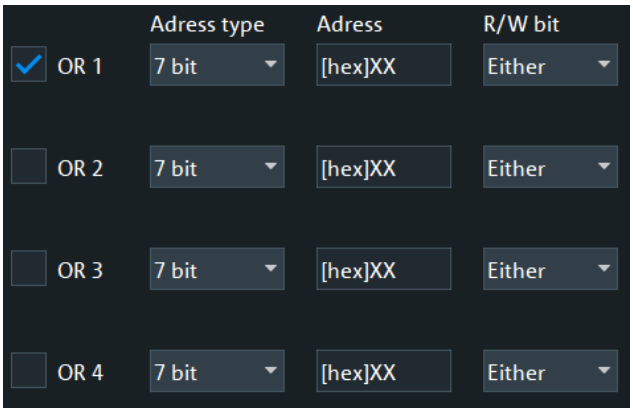
- R/W bit




**Address OR ← Trigger type**
Triggers on one to four address conditions. Each enabled condition requires an exact address. You can specify:

For each condition to be used, select "Monitor"

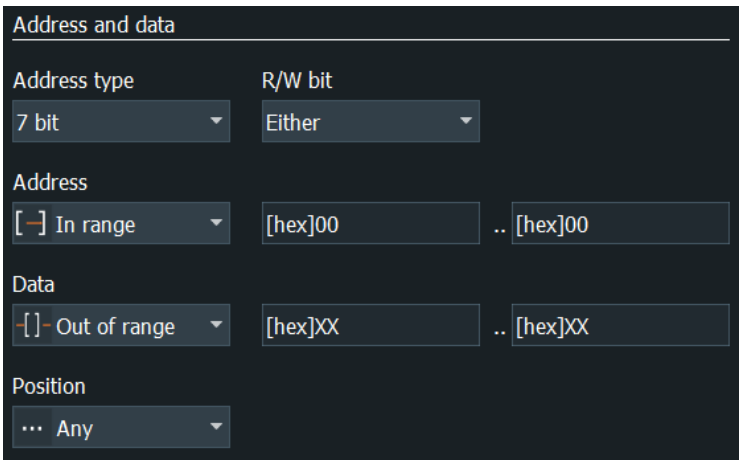
- "OR x" on page 534
- Adress Type
- "Address" on page 533
- R/W bit




**Address and data ← Trigger type**
Sets the trigger to a combination of address and data condition. You can specify:

- Adress Type
- R/W bit
- "Address" on page 533
- "Position" on page 534
- "Data" on page 534

The address conditions are the same as for the "Address" trigger type and "Data" on page 534.




**No Ack conditions**
Selects which missing acknowledge bits is detected if the trigger type is set to "Missing Ack".

"Address Nack"
No secondary recognizes the address.

"Data write Nack"
The addressed secondary does not accept the data.

"Data read Nack"
Marks the end of the read process when the primary reads data from the secondary. This Nack is sent according to the protocol definition, it is not an error.

Remote command:
`TRIGger<m>:I2C:ADNack` on page 1647
`TRIGger<m>:I2C:DWNack` on page 1648
`TRIGger<m>:I2C:DRNack` on page 1648

**Adress Type**
Sets the address length to be triggered on: 7 bit, 7+1 bit, or 10 bit. Available settings depend on the R/W bit setting of the bus configuration.

For "7 bit" and "10 bit", enter the address bits in the Address field, and use the "R/W bit" on page 534 field to select the transfer direction.

For "7+1 bit", enter the seven address bits and also the R/W bit in the "Address" field.

If the trigger type is "Address + data", you can set the address type "Any" to trigger on data only, regardless of the address.

Remote command:
`TRIGger<m>:I2C:AMODe` on page 1648
`TRIGger<m>:I2C:ADOR<n>:ADRType` on page 1650

**Address**
The trigger address setup consists of a comparison condition and one or two address patterns.

Defines the bit pattern of the secondary device address. The length of the entry is adjusted to the selected address type.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"Min" Specifies the value or sets the start value of a range.

"Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger<m>:I2C:ADDRess` on page 1649
`TRIGger<m>:I2C:ADDTo` on page 1649
`TRIGger<m>:I2C:ADOR<n>[:VALue]` on page 1650

**R/W bit**
Toggles the trigger condition between read and write access of the primary. Select "Either" if the transfer direction is not relevant for the trigger condition.

Remote command:
`TRIGger<m>:I2C:ACCess` on page 1647
`TRIGger<m>:I2C:ADOR<n>:RWBit` on page 1650

**OR x**
Enables each selected condition, for "Trigger Type" = "Address OR".

Remote command:
`TRIGger<m>:I2C:ADOR<n>:ENABle` on page 1649

**Position**
Sets the number of data bytes to be skipped after the address. You can define an exact position, or a position range. The index 0 is associated with the first data byte.

"Condition" Sets the comparison condition to a specific value or a range.

"Index min" Specifies the value or sets the start value of a range.

" Index max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger<m>:I2C:DPOPerator` on page 1651
`TRIGger<m>:I2C:DPOSition` on page 1651
`TRIGger<m>:I2C:DPTO` on page 1651

**Data**
Specifies the trigger conditions for the data bit pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"Value min " Specifies the value or sets the start value of a range.
Enter the bytes in msb first bit order. The maximum pattern length is 64 bit. Waveform data is compared with the pattern byte-by-byte.

"Value max " Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger<m>:I2C:DCONdition` on page 1651
`TRIGger<m>:I2C:DMIN` on page 1652
`TRIGger<m>:I2C:DMAX` on page 1652

#### 13.2.3.2 Triggering on I²C signals

Prerequisites: An I²C bus is configured, see Chapter 13.2.2.3, "Configuring I²C protocol", on page 529.

1. Press the [Bus] key and select the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Select the serial bus that is set to I²C.
5. Select the "Trigger type".
6. For more complex trigger types, enter the address and/or data conditions: address, acknowledge bits, R/W bit, and data pattern.
   For details, see Chapter 13.2.3, "I²C trigger", on page 530

### 13.2.4 I²C decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Serial Bus" dialog > "Setup" tab, enable "Decode".
2. In the "Serial Bus" dialog > "Display" tab, select additional result display settings: "Show table" and "Show binary".
3. If required, enable "Zoom coupling"

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode Results Table

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

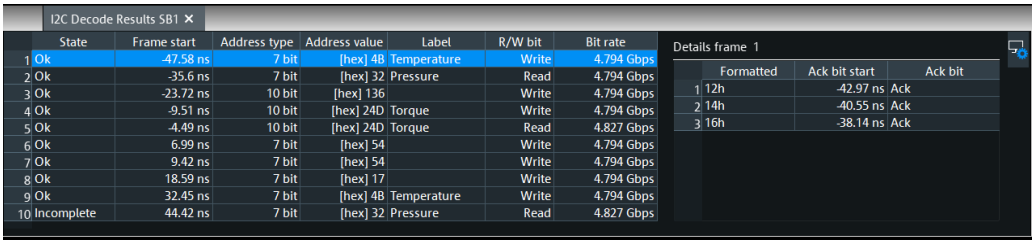



***Table 13-2: Content of the Details frame table***

| Column | Description |
|---|---|
| State | Overall state of the frame.<br>"Insuffcient waveform length" indicates that the frame is not completely contained in the acquisition. Change the horizontal scale, or move the reference point to the left to get a longer acquisition. |
| Frame start | Time of frame start |
| Address type | Address length, 7 bit or 10 bit |
| Address value (hex) | Hexadecimal value of the address |
| Label | Symbolic label name defined in the label list |
| R/W bit | Value of the R/W bit |
| Bit rate | Value of the bit rate |

***Table 13-3: Content of the Decode results table***

| Column | Description |
|---|---|
| Formatted | Value of all data bytes of the frame |
| Ack bit start | Start time of the acknowledge bit |
| Ack bit | Value of the address acknowledge bit |

### Tools

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

### Remote commands

Remote commands are described in Chapter 24.17.3.3, "Decode results", on page 1652.

### 13.2.5 Search on decoded I²C data

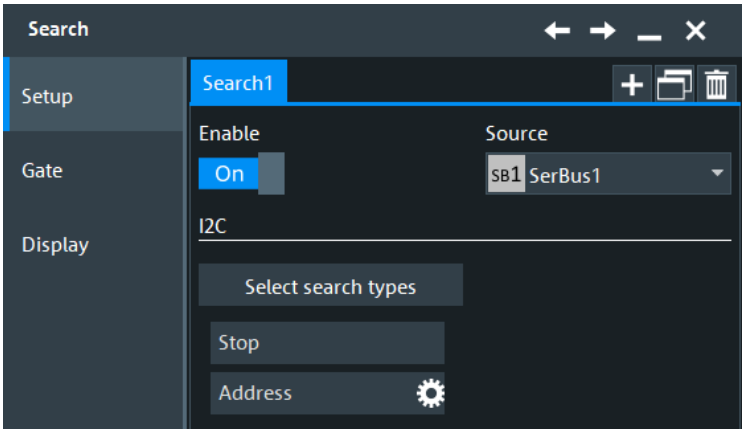
Using the search functionality, you can find various events in the decoded data, the same events which you also can trigger on. Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search "Source" to the serial bus that is configured for the protocol to be analyzed.

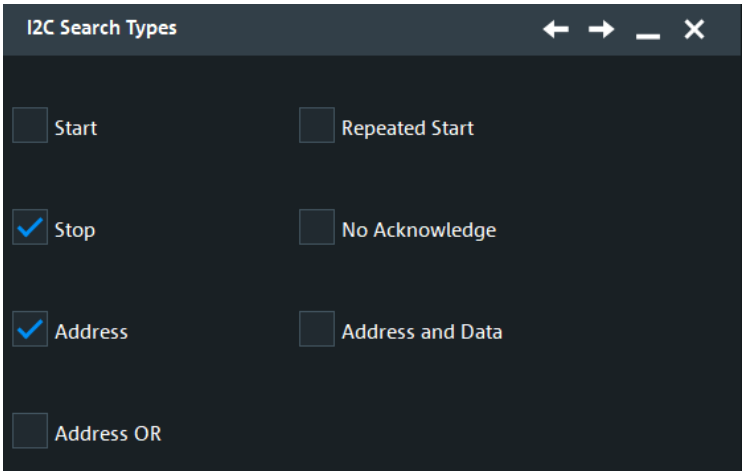
See also Chapter 11, "Search functions", on page 449.

#### 13.2.5.1 I²C search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"




**Select search types**
Enable the events to be searched for. Unlike triggering, where you can trigger only on one defined event, you can search for various different events in one search.




"Start" Searches for the start of the message. The start condition is a falling edge on SDA while SCL is high. The event is the falling edge of the SDA line.
You can change the SDA and SCL lines here if necessary.

| | |
|---|---|
| "Repeated start" | Searches for a start condition without previous stop condition. Repeated start conditions occur when a master exchanges multiple messages with a slave without releasing the bus. |
| "Stop" | Searches for the end of the message. The stop condition is a rising slope on SDA while SCL is high. |
| "No ACK (Missing ACK)" | Searches for a missing acknowledge bit: an event is found if the data line remains HIGH during the clock pulse following a transmitted byte. |
| "Address" | Searches for one specific address condition or a combination of address conditions. The event is the falling edge of the acknowledge bit after the address. |
| "Address OR" | Searches for one to four address conditions. |
| "Address and data" | Searches for a combination of address and data conditions. |

Remote command:
`SEARch:TRIGger:I2C:SCONdition` on page 1661
`SEARch:TRIGger:I2C:RCONdition` on page 1662
`SEARch:TRIGger:I2C:STCNdition` on page 1662
`SEARch:TRIGger:I2C:NACKnowledge` on page 1662
`SEARch:TRIGger:I2C:SADDress` on page 1662
`SEARch:TRIGger:I2C:ADOR` on page 1663
`SEARch:TRIGger:I2C:ADData` on page 1663

**Address setup: Condition, Type, Addr. min, Addr. max, R/W bit**
The address setup consists of the condition, type, R/W bit and one or two address patterns.

The address setup settings are the same as in the I2C trigger setup, see "Address" on page 533.

Remote command:
`SEARch:TRIGger:I2C:ACONdition` on page 1664
`SEARch:TRIGger:I2C:ADDRess` on page 1665
`SEARch:TRIGger:I2C:ADDTo` on page 1665
`SEARch:TRIGger:I2C:AMODe` on page 1664
`SEARch:TRIGger:I2C:ACCess` on page 1665

**Data setup: Condition, Position, Index min, Index max, Value min, Value max**
The data setup consists of the condition, position, and one or two index/value patterns.

The data setup settings are the same as in the I2C trigger setup, see "Data" on page 534.

Remote command:
`SEARch:TRIGger:I2C:DPOPerator` on page 1667
`SEARch:TRIGger:I2C:DPOSition` on page 1668
`SEARch:TRIGger:I2C:DPTO` on page 1668
`SEARch:TRIGger:I2C:DCONdition` on page 1668
`SEARch:TRIGger:I2C:DMIN` on page 1669
`SEARch:TRIGger:I2C:DMAX` on page 1669

**Address OR setup: , OR x Address type, Address, R/W bit**
The address OR setup consists of the monitor, address type, address and an R/W bit.

The address OR setup settings are the same as in the I2C trigger setup, see Chapter 13.2.3.1, "I²C trigger settings", on page 530.

Remote command:
`SEARch:TRIGger:I2C:ADDO<m>:ENABle` on page 1666
`SEARch:TRIGger:I2C:ADDO<m>:ADRType` on page 1666
`SEARch:TRIGger:I2C:ADDO<m>[:VALue]` on page 1666
`SEARch:TRIGger:I2C:ADDO<m>:RWBit` on page 1667

**No ACK setup: Addr/W/R NACK**
The no ACK setup consists of the Addr/W/R NACK.

The no ACK setup settings are the same as in the I2C trigger setup, see "No Ack conditions" on page 533.

Remote command:
`SEARch:TRIGger:I2C:DRNack` on page 1670
`SEARch:TRIGger:I2C:DWNack` on page 1670
`SEARch:TRIGger:I2C:NACKnowledge` on page 1662

#### 13.2.5.2 I²C search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

The remote commands for the search reults are described in Chapter 24.17.3.5, "I²C search results", on page 1670.

## 13.3 SPI bus (option R&S RTP-K1)

- The SPI protocol ..... 539
- SPI configuration ..... 540
- SPI trigger ..... 545
- SPI decode results ..... 547
- Search on decoded SPI data ..... 548

### 13.3.1 The SPI protocol

A 4-channel instrument is required for full support of the SPI protocol, or the MSO option R&S RTP-B1.

The Serial Peripheral Interface SPI is used for communication with slow peripheral devices, in particular, for transmission of data streams.

Main characteristics of SPI are:

- Master-slave communication
- No device addressing; The slave is accessed by a chip select, or slave select line.
- No acknowledgement mechanism to confirm receipt of data
- Duplex capability

Most SPI buses have four lines, two data and two control lines:

- Clock line to all slaves (SCLK)
- Slave Select or Chip Select line (SS or CS)
- Master data output, slave data input (MOSI or SDI)
- Master data input, slave data output (MISO or SDO)

When the master generates a clock and selects a slave device, data may be transferred in either or both directions simultaneously.

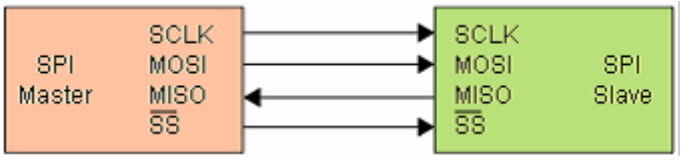



*Figure 13-4: Simple configuration of SPI bus*

The data bits of a message are grouped by following criteria:

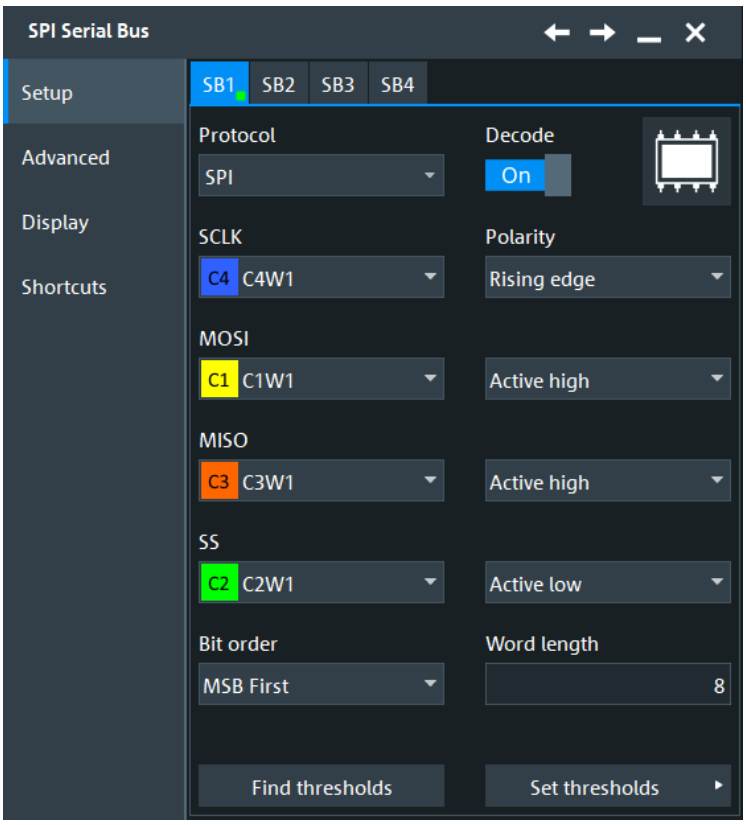
- A word contains a number of successive bits. The word length is defined in the protocol configuration.
- A frame contains a number of successive words, at least one word.

## 13.3.2 SPI configuration

### 13.3.2.1 SPI configuration settings

Access: [Bus] > "SPI" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**SCLK**
Defines the settings for the clock line.

**SCLK source ← SCLK**
Sets the input channel of the clock line. Waveform 1 of channel signals, math waveforms, and reference waveforms can be used for decoding.

Alternatively, digital channels can be used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.

For triggering on a serial bus, analog or digital channel sources are required.

Remote command:
`BUS<m>:SPI:SCLK:SOURce` on page 1678

**Polarity ← SCLK**
Two settings define the clock mode: the clock polarity and the clock phase. Together, they determine the edges of the clock signal on which the data are driven and sampled. A master/slave pair must use the same parameter pair values to communicate.

The clock polarity is "Idle low" (idle = 0) or "Idle high" (idle = 1).

The clock phase defines the slope. It selects if data is stored with the rising or falling slope of the clock. The slope marks the begin of a new bit.

**SS, MISO, MOSI**
Configures the Slave Select, MISO and MOSI lines.

**Source ← SS, MISO, MOSI**
Sets the input channel of the selected line.

Usually, the source is one of the analog channels. Reference and math waveforms are available as source if the trigger source is one of the analog channels but not the serial bus.

Do not combine a reference waveform with channel or math waveform because the time correlation of these waveforms might differ.

Alternatively, digital channels can be used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.

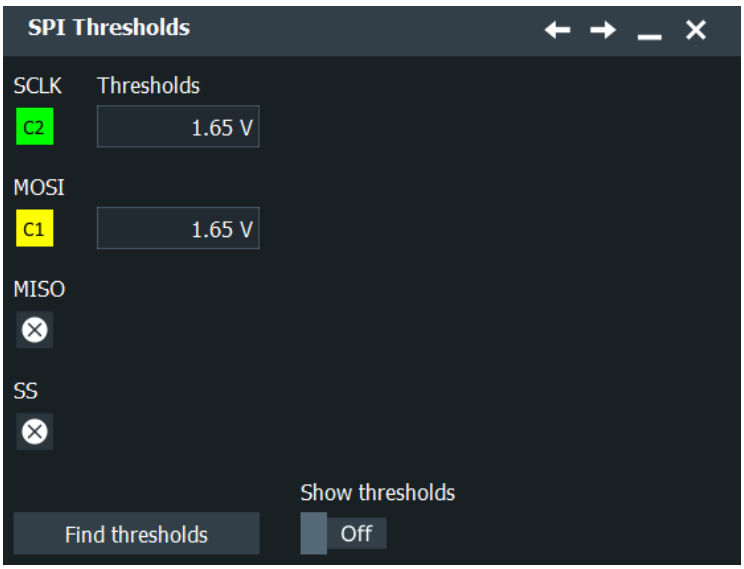
For triggering on a serial bus, analog or digital channel sources are required.

Remote command:
`BUS<m>:SPI:SSELect:SOURce` on page 1679
`BUS<m>:SPI:MISO:SOURce` on page 1679
`BUS<m>:SPI:MOSI:SOURce` on page 1680

**Polarity ← SS, MISO, MOSI**
Selects whether transmitted data or the slave select signal is high active (high = 1) or low active (low = 1).

Remote command:
`BUS<m>:SPI:SSELect:POLarity` on page 1679
`BUS<m>:SPI:MISO:POLarity` on page 1680
`BUS<m>:SPI:MOSI:POLarity` on page 1681

**Set Thresholds**
Opens a dialog, where you can set the tresholds for digitization of signals for each line. If the signal value on the line is higher than the threshold, the signal state is high. Otherwise, the signal state is considered low if the signal value is below the threshold. The interpretation of HIGH and LOW is defined by the Polarity.



There are several ways to set the threshold:

- "Threshold"
  Enter the value directly in the field.
- "Autoset"
  Starts software algorithms for determining the signal threshold levels and bitrate.
- "Find thresholds"
  Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

**Note:** If the sources are digital channels, the same threshold values are used for the parallel and the serial buses. You can set the thresholds either in the parallel bus configuration or in the serial bus configuration.

Remote command:
`BUS<m>:SPI:SCLK:THReshold` on page 1681
`BUS<m>:SPI:MISO:THReshold` on page 1681
`BUS<m>:SPI:MOSI:THReshold` on page 1681
`BUS<m>:SPI:SSELect:THReshold` on page 1681
`BUS<m>:SETReflevels` on page 1638
`BUS<m>:FAUToset` on page 1638

**Bit order**
Defines if the data of the messages starts with msb (most significant bit) or lsb (least significant bit). The display of the decoded signal considers this setting, results are displayed in the specified order.

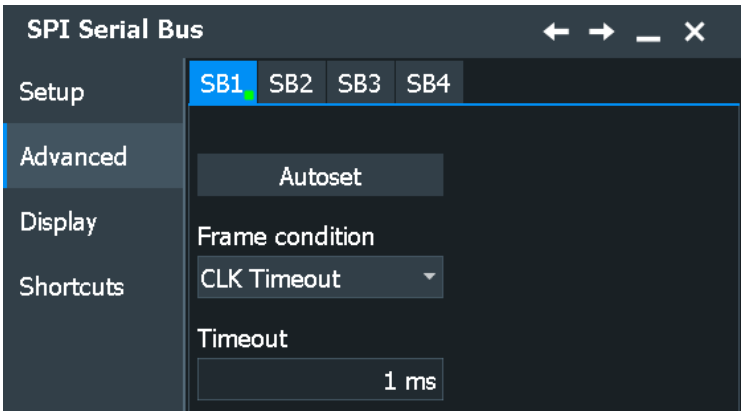
Remote command:
`BUS<m>:SPI:BORDer` on page 1678

**Word length**
Sets the number of bits in a word. The maximum length is 32 bit.

Remote command:
`BUS<m>:SPI:WSIZe` on page 1678

### 13.3.2.2 Advanced settings

Access: [Bus] > "Advanced" tab.



**Autoset**
Starts software algorithms for determining the signal threshold levels and bitrate.

Remote command:
`BUS<m>:FAUToset` on page 1638

**Frame condition**
Defines the start of a frame. A frame contains a number of successive words, at least one word.

"SS" Start and end of the frame is defined by the active state of the slave select signal.

"CLK timeout" Defines a timeout on the clock line SCLK as limiter between two frames. The timeout condition is used for SPI connections without an SS line. Enter the minimum clock idle time in the field.

Remote command:
`BUS<m>:SPI:FRCondition` on page 1681

**Timeout**
Sets the minimum clock idle time if a timeout on the clock line SCLK is used as limiter between two frames.

See also: "Frame condition" on page 544.

Remote command:
`BUS<m>:SPI:TIMeout` on page 1682

### 13.3.2.3 Configuring SPI signals

For configuration, assign the lines to the input channels, and define the active states and the logical thresholds.

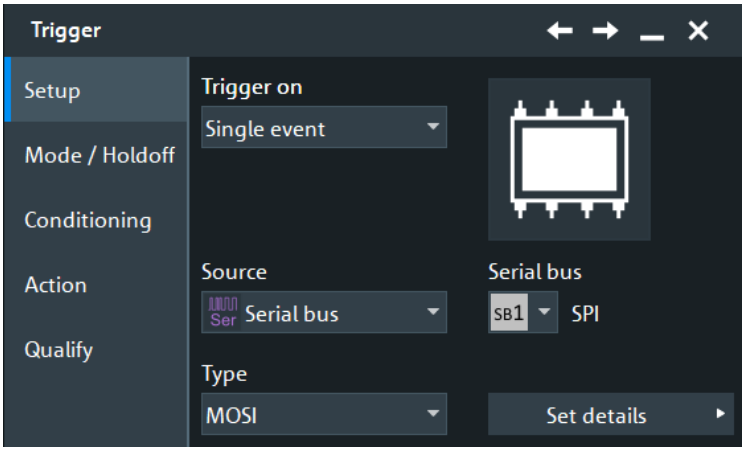
For details on configuration settings, see Chapter 13.3.2, "SPI configuration", on page 540.

1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. Tap the "Protocol" button and select the protocol: "SPI".
4. Tap the "SCLK Source" button, and select the waveform of the clock line.
5. Set the polarity (clock mode) for SCLK.
6. For each of the available SS, MISO and MOSI lines, assign the waveform. Define the polarity (active state) of the line.
7. Set the logical thresholds.
8. Set the "Bit order", "Word length", and "Frame condition" according to your signal.
9. Enable "Decode".

## 13.3.3 SPI trigger

### 13.3.3.1 SPI trigger

Access: [Bus] > "Shortcuts" tab > "Setup trigger"




Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

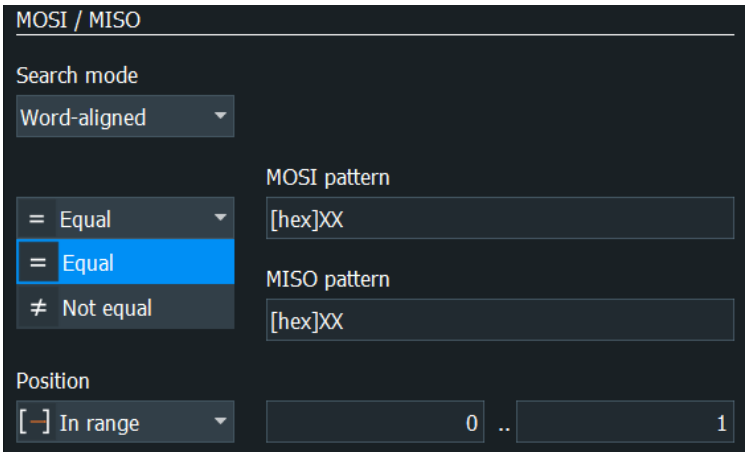
**Trigger type**
Selects the trigger type for SPI analysis.

| | |
|---|---|
| "Frame start (SS)" | Trigger on the start of the frame when the slave select signal SS changes to the active state. This trigger type is available if Frame condition is set to "SS". |
| "Frame start (Timeout)" | Triggers on the start of the frame when the clock idle time exceeds the "Timeout" time. This trigger type is available if Frame condition is set to "CLK timeout". |
| "MOSI" | Sets the trigger to a specified data pattern expected on the MOSI line.<br>See: "MOSI and MISO data conditions" on page 546. |
| "MISO" | Sets the trigger to a specified data pattern expected on the MISO line.<br>See: "MOSI and MISO data conditions" on page 546. |

"MOSI/MISO" Sets the trigger to specified data patterns expected on the MOSI and MISO lines on the same time.

Remote command:
TRIGger<m>:SPI:TYPE on page 1682

**MOSI and MISO data conditions**
The trigger on MOSI and MISO patterns is defined in the same way:




**Condition ← MOSI and MISO data conditions**
Selects the operator for the "Data" pattern: "Equal" or "Not equal".

Remote command:
TRIGger<m>:SPI:FCONdition on page 1684

**MOSI pattern, MISO pattern ← MOSI and MISO data conditions**
Specify the data patterns to be found on the MOSI and/or MISO line.

If the trigger type is "MOSI" or "MISO" (one pattern is defined), the maximum pattern length is 256 bit. If the trigger type is "MOSI/MISO", two patterns must be found at the same time. Thus, both patterns must have the same length, and the maximum pattern length of each pattern is 128 bit.

Enter the words in msb first bit order. The starting point of the pattern is defined by Index min, Index max and Search mode.

The bit pattern editor helps you to enter the pattern, see Chapter 13.1.7, "Bit pattern editor", on page 524.

Remote command:
TRIGger<m>:SPI:MOSipattern on page 1685
TRIGger<m>:SPI:MISopattern on page 1684

**Position ← MOSI and MISO data conditions**
Operator for the data position. You can define an exact position, a position range, or let the position undefined ("Any").

Remote command:
TRIGger<m>:SPI:DPOPerator on page 1683

**Index min, Index max ← MOSI and MISO data conditions**
The effect of data positioning depends on the Search mode. It sets the number of bits or words before the first word of interest. These offset bits/words are skipped. If the position operator defines a range, the first and the last bit/word of interest are defined. The index 0 is associated with the first data bit or word.

Remote command:
`TRIGger<m>:SPI:DPOSition` on page 1684
`TRIGger<m>:SPI:DPTO` on page 1684

**Search mode ← MOSI and MISO data conditions**
Defines how the specified data pattern is searched:

"Word-aligned" The pattern is matched only at word boundaries.

"Bit-aligned" Bit-by-bit: the pattern can start at any position in the message.

Remote command:
`TRIGger<m>:SPI:PALignment` on page 1683

#### 13.3.3.2 Triggering on SPI

Prerequisites: A bus is configured for the SPI signal to be analyzed.

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".
3. Select the serial bus that is set to SPI.
4. Select the "Trigger type".
5. For more complex trigger types, enter the data pattern conditions
   For details, see Chapter 13.3.3, "SPI trigger", on page 545

### 13.3.4 SPI decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".

2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode Results Table

The "Decode results" table contains information about all decoded frames.

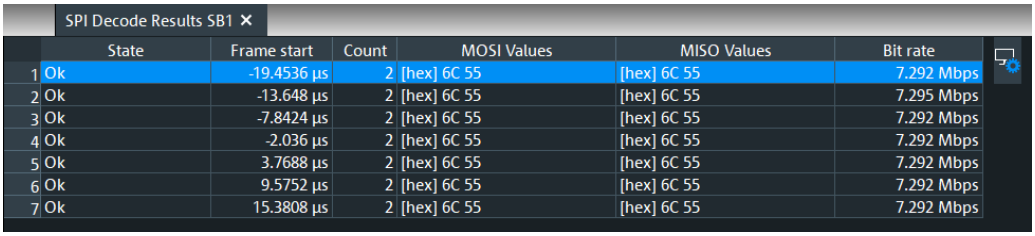



***Table 13-4: Content of the "Decode results" table***

| Column | Description |
|---|---|
| State | Overall state of the frame |
| Frame start | Times of frame start and frame end |
| Count | Number of words in the frame |
| MOSI values | Value of the MOSI data words. The data format is selected below the table. |
| MISO values | Value of the MISO data words. The data format is selected below the table. |
| Bit rate | Value of the bit rate |

### Tools

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

### Remote commands

Remote commands to retrieve decode results are described in Chapter 24.17.4.3, "SPI decode results", on page 1685.

## 13.3.5 Search on decoded SPI data

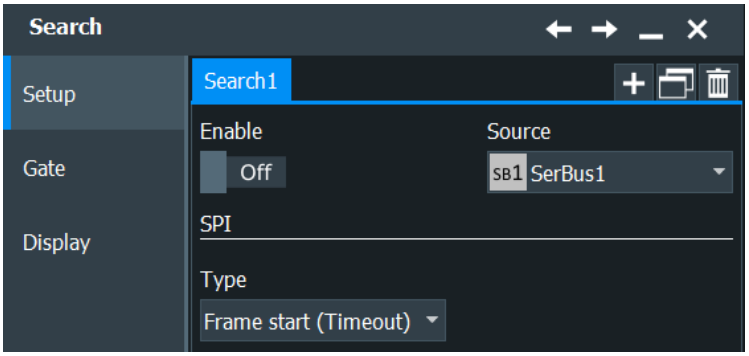
Using the search functionality, you can find various events in the decoded data, the same events which you also can trigger on. Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search "Source" to the serial bus that is configured for the protocol to be analyzed.

See also Chapter 11, "Search functions", on page 449.

### 13.3.5.1 SPI search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"



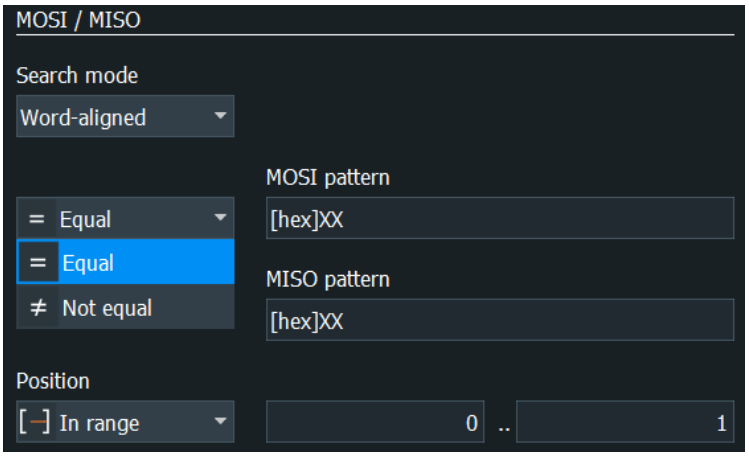

**Type**
Sets the event to be searched for.

| | |
|---|---|
| "Frame start (SS)" | Searches for the start of the frame when slave select signal SS changes to the active state. This type is available if the slave select line is configured in the bus setup, and the frame condition is "SS". |
| "Frame start (Timeout)" | Searches for the start of the frame when the clock idle time exceeds the timeout. This trigger type is available if frame condition is set to "CLK timeout". |
| "MOSI" | Searches for a specified data pattern expected on the MOSI line. |
| "MISO" | Searches for a specified data pattern expected on the MISO line. |
| "MOSI / MISO" | Searches for specified data patterns expected on the MOSI and MISO lines. |

Remote command:
`SEARch:TRIGger:SPI:TYPE` on page 1691

**MOSI and MISO data search**
The MOSI and MISO setup consists of the condition, position, MOSI pattern, MISO pattern, search mode (word-aligned, bit-aligned) and one or two index patterns.

The MOSI and MISO setup settings are the same as in the SPI trigger setup. For details, see "MOSI and MISO data conditions" on page 546.



Remote command:
SEARch:TRIGger:SPI:FCONdition on page 1691
SEARch:TRIGger:SPI:MISopattern on page 1692
SEARch:TRIGger:SPI:MOSipattern on page 1692
SEARch:TRIGger:SPI:DPOPerator on page 1692
SEARch:TRIGger:SPI:DPOSition on page 1692
SEARch:TRIGger:SPI:DPTO on page 1693
SEARch:TRIGger:SPI:PALignment on page 1693

#### 13.3.5.2 SPI search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands are listed in Chapter 24.17.4.5, "SPI search results", on page 1693.

## 13.4 UART/RS-232/RS-422/RS-485 (option R&S RTP-K2)

### 13.4.1 The UART / RS232 interface

The Universal Asynchronous Receiver/Transmitter UART converts a word of data into serial data, and vice versa. It is the base of many serial protocols such as RS-232. The UART uses only one line, or two lines for transmitter and receiver.

**Data transfer**

The data is transmitted in words, also referred to as symbols or characters. Each word consists of a start bit, several data bits, an optional parity bit, and one or more stop bits. Several words can form a package, or frame. The end of a package is marked with a reserved word or by a pause between two words.

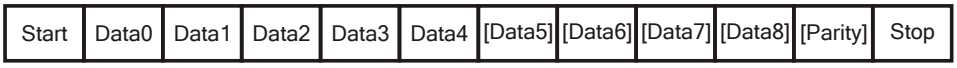



*Figure 13-5: Bit order in a UART word (symbol)*

- The start bit is a logic 0.
- The stop bits and the idle state are always logic 1.

The UART protocol has no clock for synchronization. The receiver synchronizes by means of the start and stop bits, and the bit rate that must be known to the receiver.

**Trigger**

The R&S RTP can trigger on specified parts of UART serial signals:

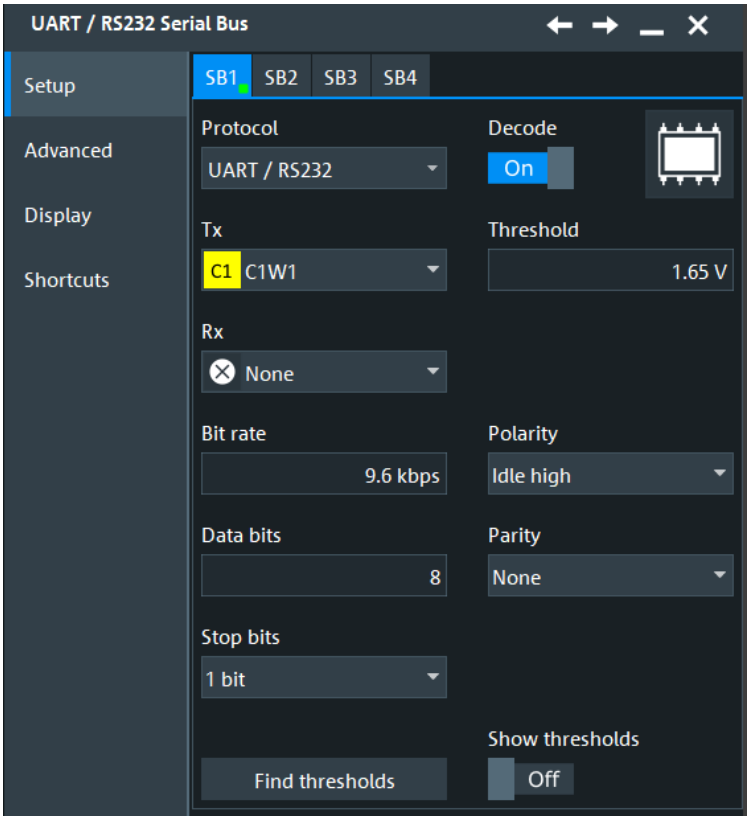
- Start bit
- Packet start
- Parity errors, and breaks
- Stop errors
- A serial pattern at any or a specified position

## 13.4.2 UART configuration

### 13.4.2.1 UART configuration settings

Access: [Bus] > "UART/RS232" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.



See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Source: Tx, Rx**
Select the input channels for the transmitter and receiver signals.

Usually, the source is one of the analog channels. Reference and math waveforms are available as source if the trigger source is one of the analog channels but not the serial bus.

Do not combine a reference waveform with channel or math waveform because the time correlation of these waveforms might differ.

Alternatively, digital channels can be used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.

For triggering on a serial bus, analog or digital channel sources are required.

Remote command:
`BUS<m>:UART:TX:SOURce` on page 1698
`BUS<m>:UART:RX:SOURce` on page 1698

**Threshold**
Sets the threshold value for digitization of signals for each line. If the signal value on the line is higher than the threshold, the signal state is high. Otherwise, the signal state is considered low if the signal value is below the threshold. The interpretation of HIGH and LOW is defined by the Polarity.

**Note:** If the sources are digital channels, the same threshold values are used for the parallel and the serial buses. You can set the thresholds either in the parallel bus configuration or in the serial bus configuration.

| | |
|---|---|
| "Threshold" | Enter the value directly in the field. |
| "Find thresholds" | Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude. |

Remote command:
`BUS<m>:UART:RX:THReshold` on page 1698
`BUS<m>:UART:TX:THReshold` on page 1699
`BUS<m>:UART:TECHnology` on page 1699
`BUS<m>:SETReflevels` on page 1638
`BUS<m>:FAUToset` on page 1638

**Polarity**
Defines the logic levels of the bus. The idle state corresponds to a logic 1. the start bit to a logic 0. "Idle high" (high=1) is used, for example, for control signals, while "Idle low" (low=1) is defined for data lines (RS-232).

Remote command:
`BUS<m>:UART:POLarity` on page 1701

**Bit rate**
Sets the number of transmitted bits per second. To select a bit rate from list of predefined values, tap the icon beside the "Bit rate" field. To enter a specific value, open the keypad. The list of predefined values is also available in the keypad.

Remote command:
BUS<m>:UART:BITRate on page 1700
BUS<m>:UART:BAUDrate on page 1700

**Data bits**
Sets the number of data bits of a word in a range from 5 bits to 8 bits. If no parity bit is used, then 9 data bits are possible.

Remote command:
BUS<m>:UART:SSIZe on page 1701

**Stop bits**
Sets the number of stop bits: 1 or 1.5 or 2 stop bits are possible.

Remote command:
BUS<m>:UART:SBIT on page 1701

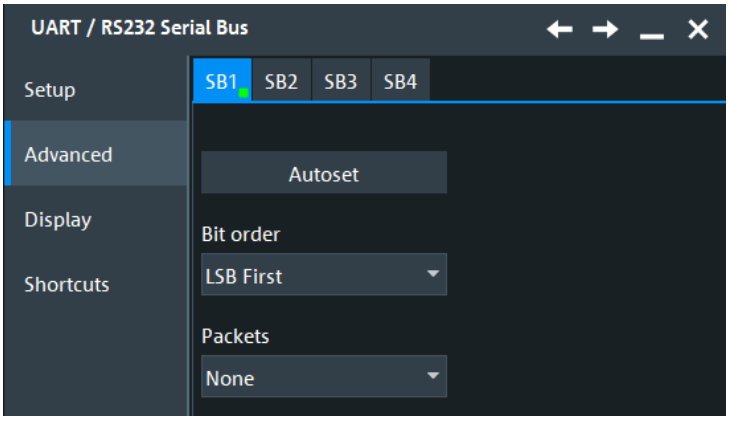
**Parity**
Defines the optional parity bit that is used for error detection.

| | |
|---|---|
| "None" | No parity bit is used. |
| "Odd" | The parity bit is set to "1" if the number of data bits set to "1" is even. |
| "Even" | The parity bit is set to "1" if the number of data bits set to "1" is odd. |
| "Mark" | The parity bit is always a logic 1. |
| "Space" | The parity bit is always a logic 0. |
| "Don't care" | The parity is ignored. |

Remote command:
BUS<m>:UART:PARity on page 1700

### 13.4.2.2 Advanced settings

Access: [Bus] > "Advanced" tab.




**Autoset**
Starts software algorithms for determining the signal threshold levels and bitrate.

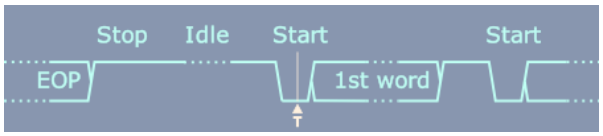
Remote command:
BUS<m>:FAUToset on page 1638

**Bit order**
Defines if a word starts with msb (most significant bit) or lsb (least significant bit). The display of the decoded signal considers this setting, results are displayed in the specified order.

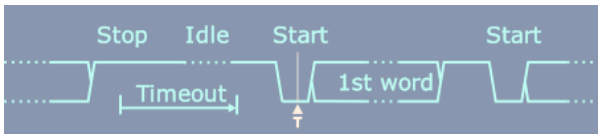
**Packets**
Allows you to define packets of several words in the data stream.

"None" Packets are not considered.

"End word" Defines a pattern as end condition of a packet, for example, a reserved word like CR or LF. The bit pattern editor provides frequently used values in the "Predefined values" list below the pattern table.
A new packet starts with the first start bit after the defined end pattern.




"Timeout" Defines a timeout between a stop bit and the next start bit. Enter the minimum time that marks the end of a packet.
A new packet starts with the first start bit after the timeout.




Remote command:
BUS<m>:UART:PACKets on page 1702
BUS<m>:UART:TOUT on page 1702
BUS<m>:UART:EWORd on page 1703

### 13.4.2.3 Configuring UART protocol

For details on configuration settings, see Chapter 13.4.2.1, "UART configuration settings", on page 551.

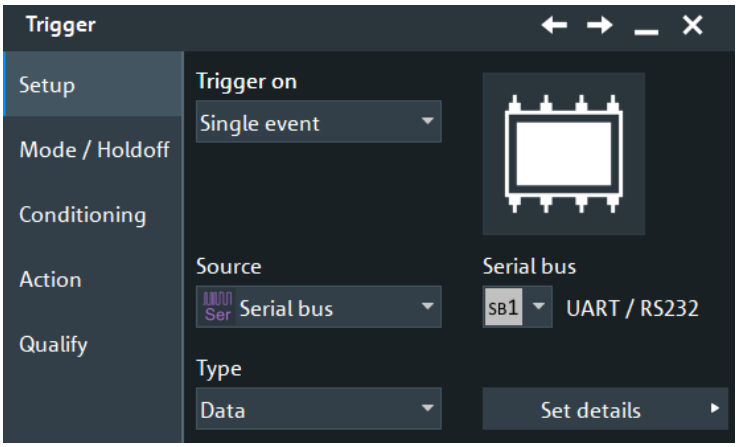
1. Press the [Bus] key on the front panel.
2. At the left-hand side, select the vertical tab of the bus you want to set up.
3. Select the "Setup" tab.
4. Tap the "Protocol" button and select the protocol: "UART".
5. Tap the "TX" button, and select the waveform of the transmitter line.
6. Tap the "Rx" button, and select the waveform of the receiver line.

7. Set the logical thresholds.
8. Set the "Bit rate" and "Stop bits".
9. Set the "Data bits" and "Parity".
10. Set the "Bit order", "Polarity" and "Packets".
11. Enable "Decode".

## 13.4.3 UART trigger

### 13.4.3.1 UART trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"



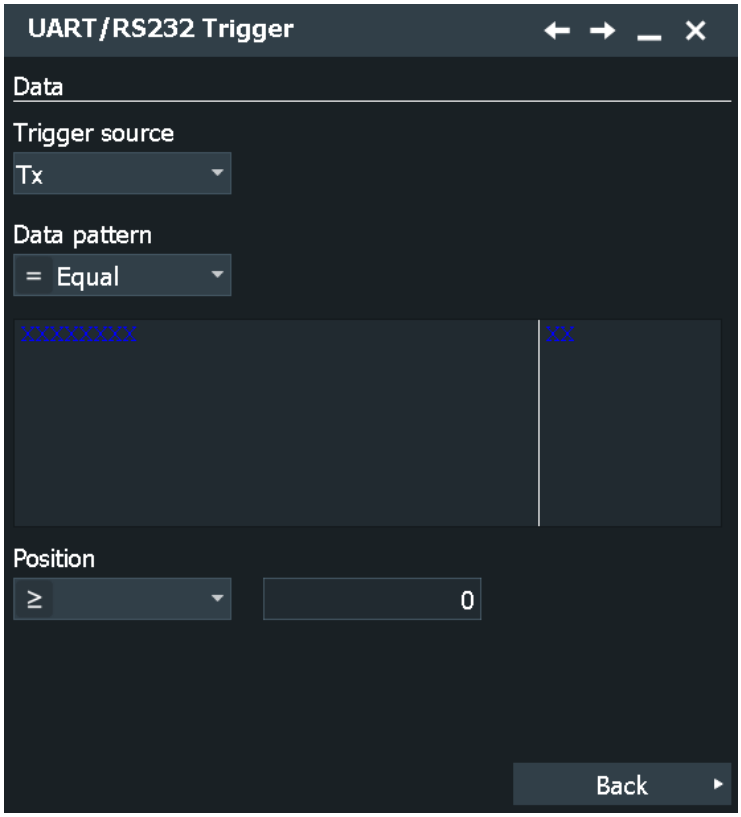

Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Type**
Selects the trigger type for UART analysis.

| | |
|---|---|
| "Start bit" | Triggers on a start bit. The start bit is the first low bit after a stop bit. |
| "Packet start" | Triggers on the begin of a data packet.<br>This trigger type is available, if Packets is enabled. |

"Data" Trigger on a serial pattern at a defined position in the data packet. The pattern can include several subsequent symbols (data frames).




See "Data conditions" on page 556.

"Parity error" Triggers on a parity error indicating a transmission error. This trigger type is only available if a parity is configured for the UART bus.

"Break condition" Triggers if a start bit is not followed by a stop bit, the data line remains at logic 0 for longer than a UART word.

"Stop error" Triggers if the stop bit is a logic 0.

Remote command:
`TRIGger<m>:UART:TYPE` on page 1704

**Trigger source**
Selects the transmitter or receiver line as trigger source.

Remote command:
`TRIGger<m>:UART:SOURce` on page 1704

**Data conditions**
Specify the data conditions if the trigger type is set to "Data".

**Condition ← Data conditions**
Selects the operator for the "Data" pattern: "Equal" or "Not equal".

Remote command:
`TRIGger<m>:UART:FCONdition` on page 1705

**Pattern ← Data conditions**
Specifies the data pattern to be found on the specified trigger source, in binary or hex format. Enter the words in msb first bit order. The starting point of the pattern is defined by Position and Index min, Index max.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

Remote command:
`TRIGger<m>:UART:DATA` on page 1705

**Position ← Data conditions**
Operator for the data position. You can define an exact position, or a position range.

The setting is available if Packets is enabled in the protocol configuration.

Remote command:
`TRIGger<m>:UART:DPOPerator` on page 1704

**Index min, Index max ← Data conditions**
Sets the number of words before the first word of interest. These offset words are ignored. If the Position defines a range, the first and the last words of interest are defined.

The setting is available if packet detection is enabled in the protocol configuration.

Remote command:
`TRIGger<m>:UART:DPOSition` on page 1704
`TRIGger<m>:UART:DPTO` on page 1705

#### 13.4.3.2 Triggering on UART signals

Prerequisites: An UART bus is configured, see Chapter 13.4.3.1, "UART trigger settings", on page 555.

1. Press the [Bus] key and select the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Select the serial bus that is set to UART.
5. Select the "Trigger type" and "Trigger source".
6. For more complex trigger types, enter the data conditions: pattern, position, index. For details, see Chapter 13.2.3, "I²C trigger", on page 530

### 13.4.4 UART decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Serial Bus" dialog > "Setup" tab, enable "Decode".

2. In the "Serial Bus" dialog > "Display" tab, select additional result display settings: "Show table" and "Show binary".
3. If required, enable "Zoom coupling"

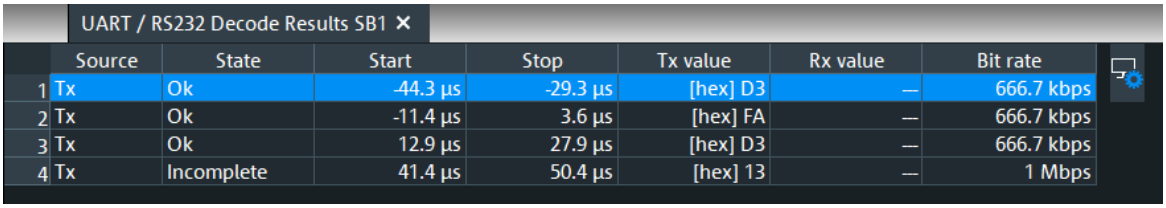
For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode Results Table

The "Decode results" box shows the detailed decoded data for each word.




The decoding process considers the "Bit order" configuration setting of the signal and displays the binary result MSB first.

***Table 13-5: Content of the Decode results table***

| Column | Description |
|---|---|
| Number of Frames | |
| Source | Line, Tx or Rx |
| State | Decoding state of the word.<br>"Insuffcient waveform length" indicates that the word is not completely contained in the acquisition. Change the horizontal scale, or move the reference point to the left to get a longer acquisition. |
| Start | Time of the word start (start bit) |
| Stop | Time of the word stop (stop bit) |
| Tx value | Value of the Tx word. The data format is selected in the "Display" tab. |
| Rx value | Value of the Rx word. The data format is selected in the "Display" tab. |
| Bit rate | Value of the bit rate . |

### Tools

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

**Remote commands**

Remote commands to retrieve decode results are described in Chapter 24.17.5.3, "Decode results", on page 1705.

## 13.5 CAN and CAN FD (option R&S RTP-K3/K9)

CAN is the Controller Area Network, a bus system designed by Bosch for use within automotive network architecture, for example, for brake, power train and engine management. Today, it is also used in many other systems, for example, in industrial machines, aerospace, subsea, merchant marine etc..

More than 20 years after the invention of CAN, communication needs have increased, and CAN has reached it's bandwidth limits in some application fields. Therefore, Bosch specified an improved CAN protocol with flexible data rate - CAN FD. It introduces a higher bit rate in the data phase up to 15 Mbit/s and an extended data field from up to 64 bytes.

The R&S RTP provides decoding, triggering and searching CAN and CAN FD signals with following options:

- CAN: option R&S RTP-K3
- CAN FD: option R&S RTP-K9, requires CAN option R&S RTP-K3

### 13.5.1 The CAN protocol

CAN frame format and in CAN FD frame format can coexist within the same network. CAN FD shares the physical layer, with the CAN protocol as defined in the BOSCH CAN Specification 2.0. The frame format is different. There are two new control bits in the CAN FD frame, the first enabling the new frame format with different data length coding and the second optionally switching to a faster bit rate after the arbitration is decided. New CRC polynomials are introduced to secure the longer CAN FD frames.

### 13.5.2 CAN and CAN FD configuration

#### 13.5.2.1 CAN and CAN FD configuration settings

Access: [Bus] > "CAN" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.

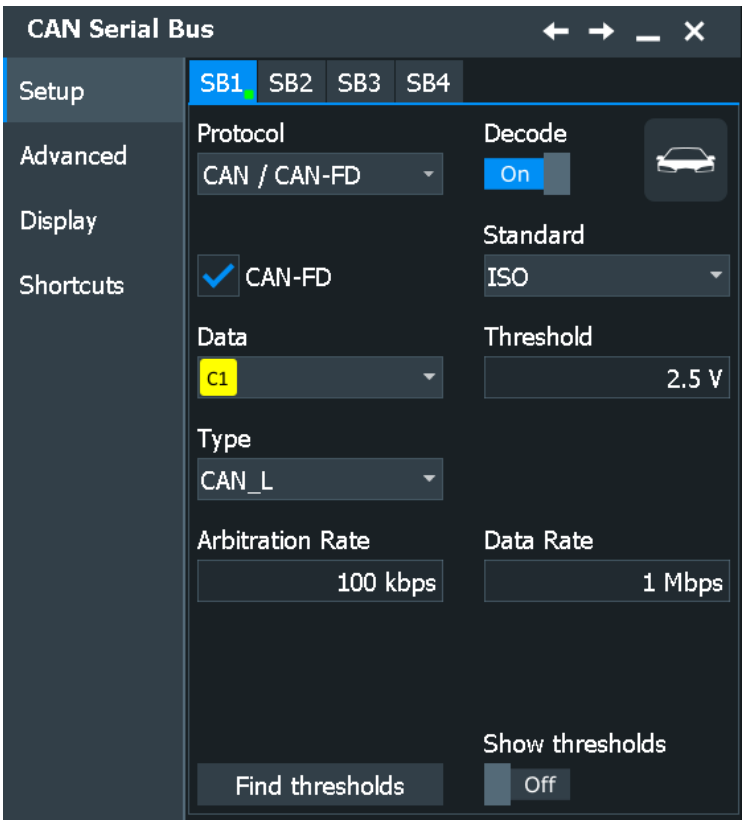



*Figure 13-6: Configuration for CAN FD*

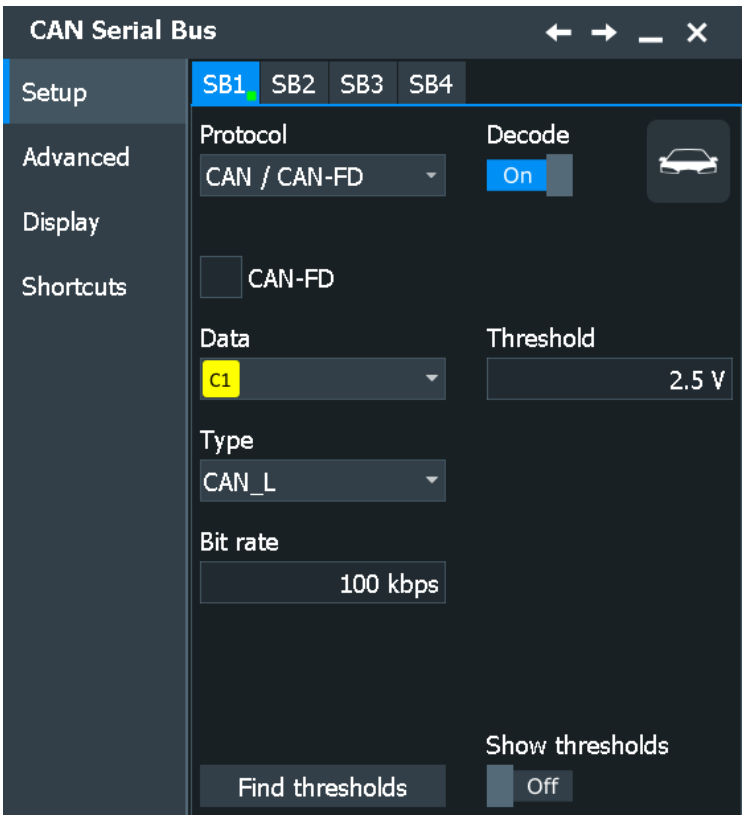



*Figure 13-7: Configuration for CAN*

See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Protocol**
Defines the protocol type of the selected serial bus.

Remote command:
`BUS<m>:TYPE` on page 1637

**Decode**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
`BUS<m>[:STATe]` on page 1637

**Data**
Sets the source of the selected data line.

Usually, the source is one of the analog channels. Reference and math waveforms are available as source if the trigger source is one of the analog channels but not the serial bus.

Alternatively, digital channels can be used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.

For triggering on a serial bus, analog or digital channel sources are required.

A math waveform can be used, for example, if you probe on CAN-High and CAN-Low using two single-ended probes, and the difference between high and low is calculated and displayed using a math waveform.

Remote command:
BUS<m>:CAN:DATA:SOURce on page 1708

**Type**
Selects the CAN-High or CAN-Low line. CAN uses both lines for differential signal transmission.

If you measure with a differential probe, connect the probe to both CAN-H and CAN-L lines, and select the data "Type" = *CAN-H*.

If you use a single-ended probe, connect the probe to either CAN_L or CAN_H, and select the data type accordingly.

Remote command:
BUS<m>:CAN:TYPE on page 1709

**CAN-FD**
Enables the CAN FD protocol configuration and displays additional CAN FD parameters. The setting is available in CAN FD option R&S RTP-K9.

Remote command:
BUS<m>:CAN:FDATa:ENABle on page 1711
BUS<m>:CAN:FDATa:FRAMe<n>:STANdard? on page 1723

**Standard**
Only available for CAN FD buses.

"Non-ISO" Signals are decoded according to the the Bosch CAN FD protocol.

"ISO" Signals are decoded according to the the ISO CAN FD protocol. This protocol has an additional stuff count field before the CRC sequence.

Remote command:
BUS<m>:CAN:FDATa:PSTandard on page 1709

**Threshold**
Sets the threshold value for digitization of the signal. If the signal value on the line is higher than the threshold, the signal state is high (1 or true for the boolean logic). Otherwise, the signal state is considered low (0 or false).

There are several ways to set the threshold:

- "Threshold"
  Enter the value directly in the field.
- "Autoset"
  Starts software algorithms for determining the signal threshold levels and bitrate.
- "Find thresholds"
  Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

**Note:** If the sources are digital channels, the same threshold values are used for the parallel and the serial buses. You can set the thresholds either in the parallel bus configuration or in the serial bus configuration.

Remote command:
BUS<m>:CAN:DATA:THReshold on page 1710
BUS<m>:CAN:TECHnology on page 1710
BUS<m>:SETReflevels on page 1638
BUS<m>:FAUToset on page 1638

**Bit rate (CAN) / Arbitration rate (CAN FD)**
For CAN buses, the "Bit rate" sets the number of transmitted bits per second.

For CAN FD buses, this parameter is called "Arbitration rate" and sets the bit rate of the arbitration phase.

The maximum value of this rate is 1 Mbit/s. The bit rate is uniform and fixed for a given CAN or CAN FD bus.

To select a bit rate from the list of predefined values, tap the button beside the field. To enter a specific value, open the keypad. The list of predefined values is also available in the keypad.

Remote command:
BUS<m>:CAN:BITRate on page 1710

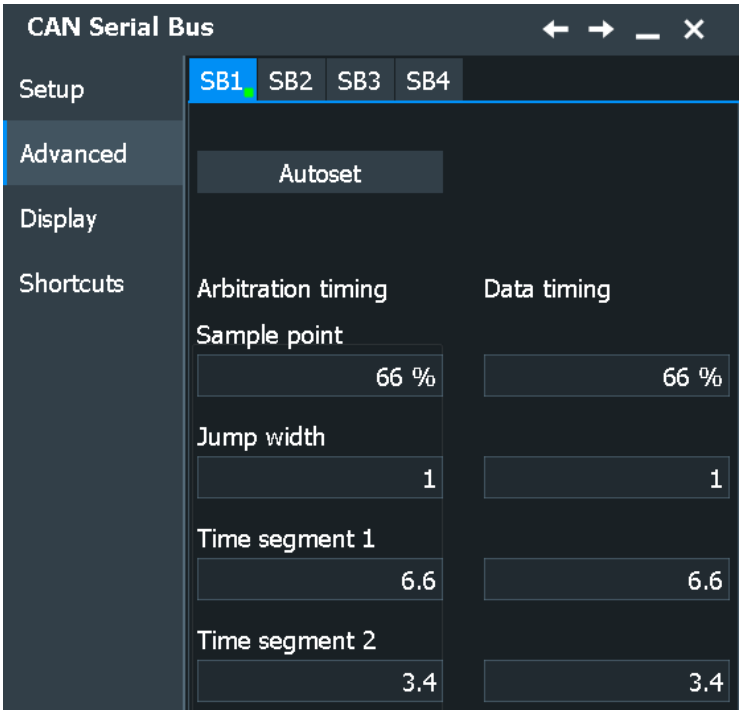
**Data rate**
Sets the bit rate of the data phase. The data rate can be equal or higher than the arbitration rate; and it is uniform and fixed for a given CAN FD bus. The setting is available in CAN FD option R&S RTP-K9.

To select a data rate from the list of predefined values, tap the button beside the field. To enter a specific value, open the keypad. The list of predefined values is also available in the keypad.

Remote command:
BUS<m>:CAN:FDATa:DBITrate on page 1711

#### 13.5.2.2 Advanced settings

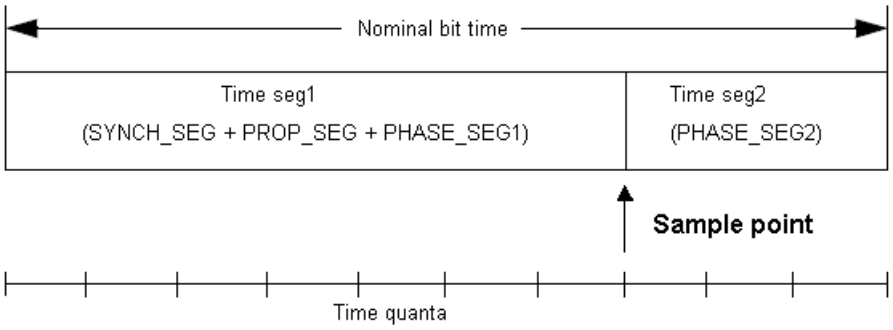
Access: [Bus] > "Advanced" tab.




The CAN bus interface uses an asynchronous transmission scheme. The standard specifies a set of rules to resynchronize the local clock of a CAN node to the message.

The sample point divides the nominal bit period into two distinct time segments. The length of the time segments is defined in time quanta according to network and node conditions during CAN development.

For CAN FD signals, you can define the sysnchronization settings separately for the arbitration phase and data phase.




To specify the bit timing, enter either "Time segment1" and "Time segment 2", or directly the "Sample point". Additionally, set the "Jump width".

**Autoset**
Starts software algorithms for determining the signal threshold levels and bitrate.

Remote command:
`BUS<m>:FAUToset` on page 1638

**Time segment1, Time segment2**
Set the number of time quanta before the sample point (Time segment 1) and after the sample point (Time segment 2).

The "Sample point" percentage value is adjusted accordingly.

Time segments 1 comprises the segments Synch_seg, Prop_seg, and Phase_seg1 which are specified in the CAN standard. Time seg2 matches Phase_seg2 from the standard.

The maximum sum of Time segment 1 and Time segment 2 is 24.

Remote command:
`BUS<m>:CAN:T1Segment` on page 1712
`BUS<m>:CAN:T2Segment` on page 1712
`BUS<m>:CAN:FDATa:T1Segment` on page 1712
`BUS<m>:CAN:FDATa:T2Segment` on page 1712

**Sample point**
Sets the position of the sample point within the bit in percent ot the nominal bit time.

The time quanta values "Time segment1, Time segment2" are adjusted accordingly.

Remote command:
`BUS<m>:CAN:SAMPlepoint` on page 1711
`BUS<m>:CAN:FDATa:SAMPlepoint` on page 1711

**Jump width**
Time segment1 may be lengthened or time segment2 may be shortened due to resynchronization. Resynchronization corrects the phase error of an edge caused by the drift of the oscillators. The jump width defines the maximum number of time quanta for phase correction. The maximum value of the jump width is 4, or *Time seg1 - Time seg2* if this difference is lower than 4.

Remote command:
`BUS<m>:CAN:JWIDth` on page 1713
`BUS<m>:CAN:FDATa:JWIDth` on page 1713

#### 13.5.2.3 Display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

**DBC Files for CAN / CAN FD**

Industry-standard DBC files contain more information than PTT and CSV files and translate the abstract decode results to human language. For each frame, the frame ID and the symbolic name of the ID are given; the frames are also called messages in CAN. The data of a CAN message can consist of several "signals". The DBC file provides the label, unit, start bit, length and other indicators for each signal. For state-encoded signals, the meaning of the states is given.

In the demo example, the message "EngineData" has the decimal ID 2,166,573,756 and consists of 8 data bytes. These 8 bytes are defined as 6 signals. The first one,

"PetrolLevel", starts at bit #24, has a length of 8 bit, and the unit is liter. The signal "IdleRunning" is state-encoded. It has only one bit. The binary value 0 means "Running", and the binary value 1 means "Idle".

**Example: CAN DBC file section**

```
BO_ 2166573756 EngineData: 8 Engine
 SG_ PetrolLevel : 24|8@1+ (1,0) [0|255] "l" ...
 SG_ EngPower : 48|16@1+ (0.01,0) [0|350] "kW" ...
 SG_ EngForce : 32|10@1+ (1,0) [0|1000] "N" ...
 SG_ IdleRunning : 23|1@1+ (1,0) [0|1] "" ...
 SG_ EngTemp : 16|7@1+ (2,-50) [-50|150] "degC" ....
 SG_ EngSpeed : 0|13@1+ (1,0) [0|8000] "rpm" ...
....
 VAL_ 2166573756 IdleRunning 0 "Running" 1 "Idle" ;
```

**PTT and CSV Files for CAN / CAN FD**

Label list files are protocol-specific. A PTT label file for CAN protocols contains three values for each identifier:

- Identifier type, 11-bit or 29-bit long
- Identifier value
- Label, symbolic name of the identifier, specifying its function in the bus network.

**Example: CAN PTT file**

```
# ------------------------------------------------------------------------------
@FILE_VERSION  = 1.00
@PROTOCOL_NAME = can
# ------------------------------------------------------------------------------
# Labels for CAN protocol
#   Column order: Identifier type, Identifier value, Label
# ------------------------------------------------------------------------------
11,064h,Diag_Response
11,1E5h,EngineData
11,0A2h,Ignition_Info
11,1BCh,TP_Console
11,333h,ABSdata
11,313h,Door_Left
11,314h,Door_Right
29,01A54321h,Throttle
29,13A00FA2h,LightState
29,0630ABCDh,Engine_Status
29,03B1C002h,Airbag_Status
29,01234ABCh,NM_Gateway
# ------------------------------------------------------------------------------
```

Remote command:

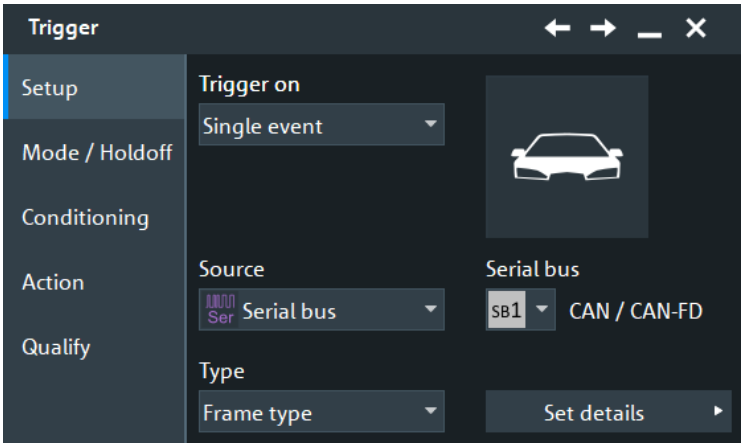
- `BUS<m>:CAN:FRAMe<n>:SYMBol?` on page 1725

### 13.5.3 CAN / CAN FD trigger

The R&S RTP can trigger on various events in a CAN or CAN FD frame. Trigger conditions include start of frame, frame ID, data pattern, or error conditions.

#### 13.5.3.1 Trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"




Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Serial bus**
Selects the serial bus to be triggered on. Make sure to select the correct bus before you enter the settings.

To trigger on a serial bus, the signals sources must be channel signals. If the data or clock source is a math or reference waveform, you cannot trigger on that bus.

Remote command:
`TRIGger<m>:SOURce:SBSelect` on page 1642

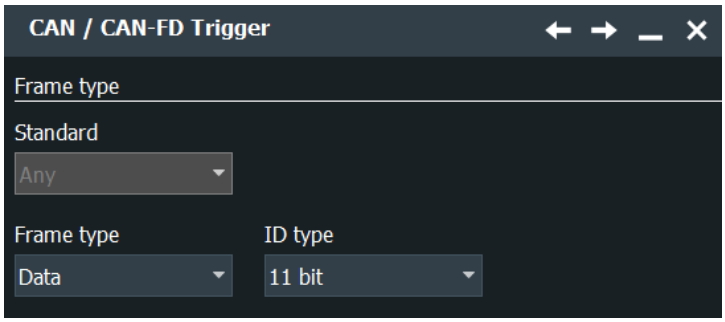
**Protocol**
Defines the protocol type of the selected serial bus.

Remote command:
`BUS<m>:TYPE` on page 1637

**Trigger type**
Selects the trigger type for CAN analysis.

"Start of frame" Triggers on the first edge of the dominant SOF bit (synchronization bit).

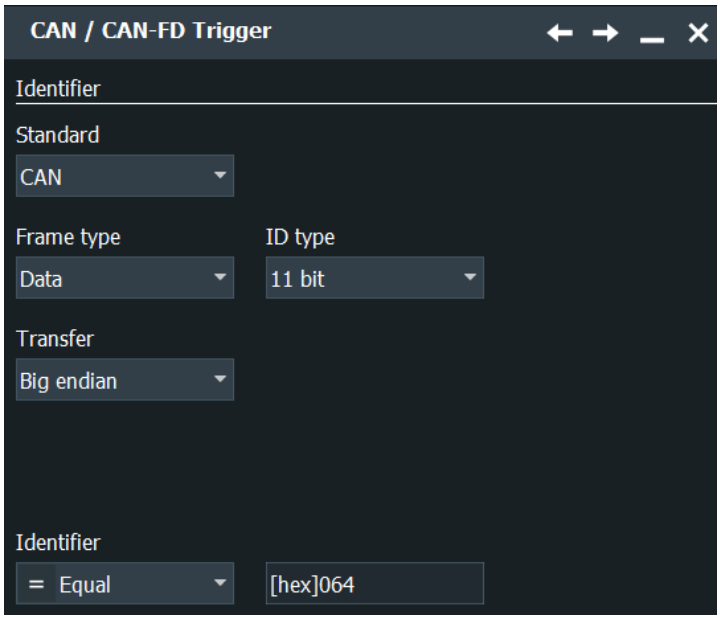
"Frame type"




Triggers on a specified frame type (data, remote, error, or overload). For data and remote frames, also the identifier format is considered.

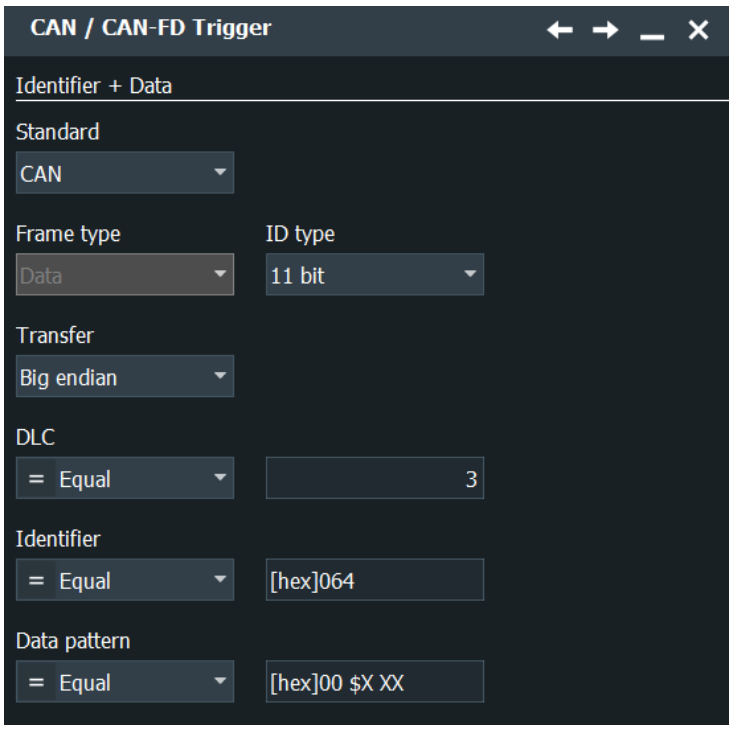
For details, see:
- "Frame type" on page 570
- "ID type" on page 570

"Identifier"




Sets the trigger to a specific message identifier or an identifier range.
See "Identifier setup: Condition, Identifier min, Identifier max" on page 571.

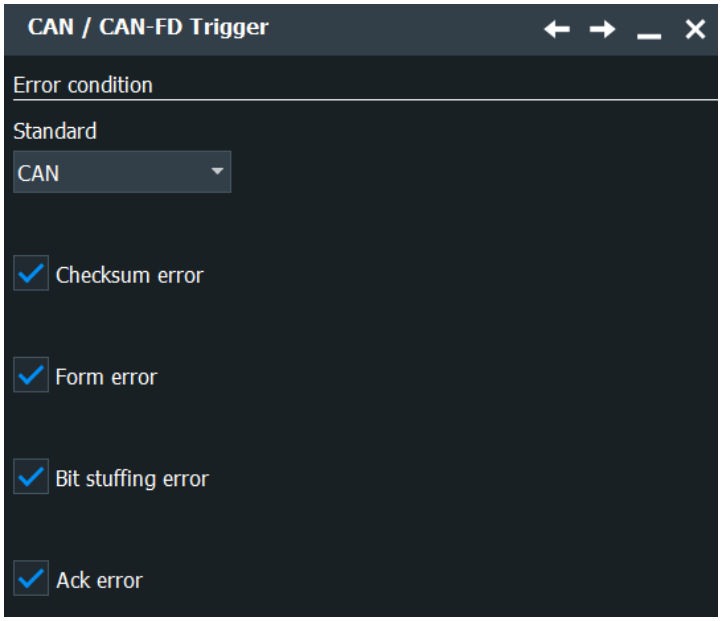
"Identifier + Data"




Sets the trigger to a combination of identifier and data condition. The instrument triggers at the end of the last byte of the specified data pattern.
The identifier conditions are the same as for the "Identifier" trigger type, see "Identifier setup: Condition, Identifier min, Identifier max" on page 571. Data conditions are set with "Data setup: DLC, NDB, Transfer, Condition, Data min, Data max" on page 572.

"Error condition"

Identifies various errors in the frame, see "Error conditions" on page 573.



"Symbolic" The "Symbolic" trigger type is available if a DBC label list file is loaded and applied. It allows you to trigger on a specific data message, or a signal and its value that appears inside the message, see Chapter 13.5.6.1, "Symbolic trigger", on page 581.

Remote command:
`TRIGger<m>:CAN:TYPE` on page 1714

**Standard**
Selects the CAN standard: "CAN", "CAN FD", or "Any". Use "Any" to trigger on either CAN or CAN-FD frame. In this case, the trigger configuration provides all possible settings, for CAN as well as for CAN FD. The setting is available in CAN FD option R&S RTP-K9.

Remote command:
`TRIGger<m>:CAN:FDATa:STANdard` on page 1715

**Frame type**
CAN has four frame types which can be used as trigger condition.

For data and remote frames, the identifier format has to be set with ID type.

"Data" The data frame is the only frame for actual data transmission.

"Remote" Remote frames are only available in the CAN protocol.
The remote frame initiates the transmission of data by another node.
The frame format is the same as of data frames but without the data field.

"Error" When a node recognizes an error, it cancels transmission by sending an error frame.
The instrument triggers seven bit periods after the end of the error flag that is marked by a dominant-recessive edge.
The ID type is irrelevant for error frames.

"Overload" When a node needs a delay between data and/or remote frames, it sends and overload frame.
The instrument triggers seven bit periods after the end of the overload flag that is marked by a dominant-recessive edge.
The ID type is irrelevant for overload frames.

Remote command:
`TRIGger<m>:CAN:FTYPe` on page 1715

**ID type**
Selects the length of the identifier:

"11 bit" Identifier length of the CAN base frame format. The instrument triggers on the sample point of the IDE bit (identifier extension flag).

"29 bit" Identifier length of the CAN extended frame format. The instrument triggers on the sample point of the RTR bit.

"Any" The ID type and ID pattern are not relevant for the trigger condition. If the trigger type is "Identifier", the instrument triggers on any identifier in the specified frame type.
If the trigger type is "Identifier + Data", set the "ID type" to "Any" if you want to trigger only on data.

Remote command:
`TRIGger<m>:CAN:ITYPe` on page 1716

**Identifier setup: Condition, Identifier min, Identifier max**
The identifier setup consists mainly of the condition and one or two identifier patterns. Additionally, ID type and frame type may qualify the identifier.

The trigger point depends on the ID type.

"Frame type" Data frames and remote frames contain an identifier. Select the frame type to be triggered on, or select "Any" if the frame type is not relevant.
In CAN FD, only "Data" frames are available.

"ID type" See: "ID type" on page 570.

"Condition" Defines the operator to set a specific identifier ("Equal" or "Not equal") or an identifier range.

"Identifier min" In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.
The length of the bit patterns is restricted to the selected "ID type".

"Identifier max" The second identifier pattern is required to specify a range with conditions "In range" and "Out of range".

"FD bits" See: "FD bits" on page 571.

Remote command:
`TRIGger<m>:CAN:ICONdition` on page 1716
`TRIGger<m>:CAN:IMIN` on page 1716
`TRIGger<m>:CAN:IMAX` on page 1717

**FD bits**
For standard settings "CAN FD" and "Any", you can trigger on CAN FD-specific bits.

"FDF" The bit determines whether a frame is CAN or CAN-FD. It corresponds to the EDL bit (extended data length), which only exists in CAN FD format. If you do not know if the signal is CAN or CAN FD, you can use this bit to identify the format: FDF = 1 is CAN FD, and FDF = 0 is CAN. Set "X" if the format is not relevant.

"BRS" is the bit rate switch bit. Value 1 means that the bit rate switches from the "Arbitration rate" to the faster "Data rate".

"ESI" is the error state indicator. Set "X" if the bit is not relevant.

Remote command:
`TRIGger<m>:CAN:FDATa:FDF` on page 1717
`TRIGger<m>:CAN:FDATa:BRS` on page 1717
`TRIGger<m>:CAN:FDATa:ESI` on page 1717

**Data setup: DLC, NDB, Transfer, Condition, Data min, Data max**
The data setup consists of the transfer direction, the number of bytes, the condition, and one or two data patterns.

To trigger only on data, set the "ID type" of the identifier setup to "Any".

"Transfer" CAN only:

Sets the byte order (endianness) of the data transfer. With "Big endian", the data is analyzed and evaluated in the order of reception. With "Little endian", the instrument reads the complete data, reverses the byte order of the data, and compares it with the specified data word.

"DLC, NDB" "DLC" sets the Data Length Code, which defines the number of data bytes to be found.
"NDB" shows the number of data bytes that is set by the DLC. DLC and NDB are different in CAN FD for DLCs > 8.
CAN:

For Big Endian transfer direction, you can trigger on a number of bytes less than the data length of the frame, that means, on the first bytes that are transmitted. For Little Endian transfer direction, the exact number of data bytes in the frame must be set.
Example: The data word to be sent is *12 34 56*, and it is sent little endian by the LIN node. With Data length ≥ 2 and Transfer = Big endian, you trigger on the data of the first two bytes, that is *56 34*. With Data length = 3 and Transfer = Little endian, you trigger on the required data word *12 34 56*.
CAN FD:
The data field can have up to 64 bytes, the DLC is defined in the standard. For example, DLC = 9 defines that the data field has 12 bytes, and DLC = 15 sets a 64 byte data field.

"Condition" Sets the operator to set a specific data pattern ("Equal" or "Not equal") or a data range.

"Data min" Defines the data pattern. The pattern length is adjusted to the DLC setting (and vice versa). Enter the pattern MSB first and with big endian byte order.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:CAN:BORDer` on page 1719
`TRIGger<m>:CAN:DCONdition` on page 1718
`TRIGger<m>:CAN:DMIN` on page 1718
`TRIGger<m>:CAN:DMAX` on page 1718
`TRIGger<m>:CAN:DLCCondition` on page 1719
`TRIGger<m>:CAN:DLC` on page 1719
`TRIGger<m>:CAN:NDBYtes?` on page 1720

**Data position**
The data position sets the location in the data field where the instrument looks for the specified data pattern. The position can be defined if the data field of the frame is longer than 8 bytes - if DLC ≥9. The setting is available in CAN FD option R&S RTP-K9.

| | |
|---|---|
| "Position" | Sets the operator to define an exact position ("Equal") or a data range.<br>Use "Any", if the data position is not relevant for the trigger condition. |
| "Data index (min)" | Defines the number of the first data byte at which the data pattern may start. |
| "Data index (max)" | Sets the number of the last byte at which the required data pattern may start if the "Position" operator is "In range". |

Remote command:
`TRIGger<m>:CAN:FDATa:DPOPerator` on page 1720
`TRIGger<m>:CAN:FDATa:DPOSition` on page 1720
`TRIGger<m>:CAN:FDATa:DPTO` on page 1721

**Error conditions**
If a CAN detects an error, it transmits an error flag at the next bit. The R&S RTP detects errors in the message and triggers on these errors even if no CAN node sends an error flag.

- CRC error
  CAN uses the Cyclic Redundancy Check, which is a complex checksum calculation method. The transmitter calculates the CRC and sends the result in the CRC sequence. The receiver calculates the CRC in the same way. A CRC error occurs when the calculated result differs from the received value in the CRC sequence.
- Bit stuffing error
  The frame segments Start Of Frame, Arbitration Field, Control Field, Data Field and CRC Sequence are coded by the bit stuffing method. The transmitter automatically inserts a complementary bit into the bit stream when it detects five consecutive bits of identical value in the bit stream to be transmitted. A stuff error occurs when the 6th consecutive equal bit level in the mentioned fields is detected.
- Form error
  A form error occurs when a fixed-form bit field contains one or more illegal bits.
- Ack error
  An acknowledgement error occurs when the transmitter does not receive an acknowledgment - a dominant bit during the Ack Slot.
- Stuff count error
  A stuff count error occurs if th received stuff count value does not match the value calculated from the own stuff bit count. Only relevant for CAN FD signals in ISO standard.

Remote command:
`TRIGger<m>:CAN:CRCerror` on page 1721
`TRIGger<m>:CAN:BITSterror` on page 1721
`TRIGger<m>:CAN:FORMerror` on page 1722
`TRIGger<m>:CAN:ACKerror` on page 1721
`TRIGger<m>:CAN:FDATa:SCERror` on page 1722

#### 13.5.3.2 Triggering on CAN FD data

The "Identifier + Data" trigger type supports triggering on data bytes of specific value at specific location in the data field of a frame. The "Data Pattern" field provides 8 data bytes to define the data pattern. For data fields longer than 8 byte, you can define the position where the specified pattern starts.

The following examples demonstrate how the data pattern and data position are defined.

**To set up the trigger**

1. Set the basic trigger events:
   a) Select the source: "Serial bus".
   b) Select the serial bus.
   c) Select the protocol: "CAN/CAN FD".
   d) Select the trigger type: "Identifier + Data".
   e) Select the standard:"CAN FD" or "Any".
2. In this example, the identifier does not matter. Set the "ID type = Any".
3. Define the data setup as described in the examples.

**Example: Triggering on the second data byte**

The CAN FD frame has 2 or more data bytes, where the value of the second data byte should be E7.

- Set "DLC ≥ 2".
- Set the data pattern: "= XX E7".

**Example: Triggering on the seventh or later data byte**

The CAN FD frame has 12 or more data bytes, where the value of the seventh or later data byte should be 17.

- Set "DLC ≥ 9".
- Set the data pattern: "= XX XX XX XX XX XX 17 XX".

**Example: Triggering on a data byte at a given position**

The CAN FD frame has 12 or more data bytes. the trigger is set at the 8th data byte starting at the 4th data byte or later, with data byte value = 17.

- Set "DLC ≥ 9".
- Set the data pattern: "= XX XX XX XX XX XX XX 17".
- Set the position of the data pattern: "In range", "4" to "12".

The instrument skips the first 3 data bytes and starts comparing the data pattern with the 4th data byte. So, the byte with value 17 can be found between the 11th and the 19th data byte.

### 13.5.4 CAN and CAN FD decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".

2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

The binary results of data bytes are displayed MSB first.

For CAN protocol, the endianness setting ("Transfer") is a trigger setting and not considered for decoding.

You can also load and apply industry-standard DBC files. The symbolic names from the file are applied to the display of the decoded data, see Chapter 13.5.6.2, "Symbolic decode waveform", on page 582.

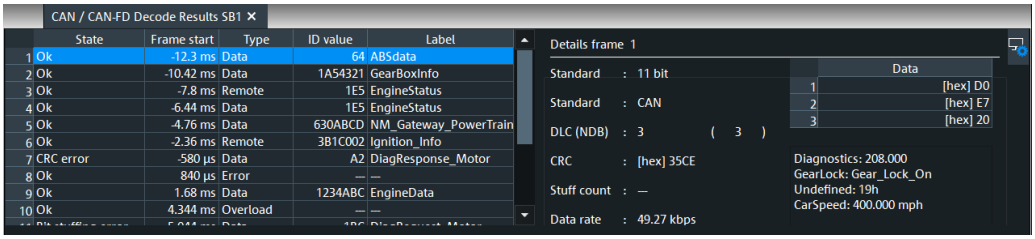



***Table 13-6: Content of the Decode results table***

| Column | Description |
|---|---|
| State | Overall state of the frame.<br>"Insuffcient waveform length" indicates that the frame is not completely contained in the acquisition. Change the horizontal scale, or move the reference point to the left to get a longer acquisition. |
| Frame start | Time of frame start |
| Type | Frame type: Data, Remote, Error, or Overload |
| ID value (hex) | Identifier value, hexadecimal value |

***Table 13-7: Content of the Details frame table***

| Column | Description |
|---|---|
| Standard | Frame format, CAN or CAN FD. |
| Label | Symbolic label name defined in the label list |
| DLC | Data length code, coded number of data bytes |

| Column | Description |
|---|---|
| NDB | Actual number of data bytes |
| Data | Value of the data frame. The data format is selected below the table. Remote frames do not transmit data, therefore "- - -" is displayed. |
| CRC (hex) | Value of the Cyclic Redundancy Check (checksum), hexadecimal value |
| Data rate | Reason of a form error if a form error occurred |

**Tools**

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

**Remote commands**

Remote commands are described in Chapter 24.17.6.3, "Decode results", on page 1722.

## 13.5.5 Search on decoded CAN or CAN FD data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search source to "SerBus" for the configured protocol.

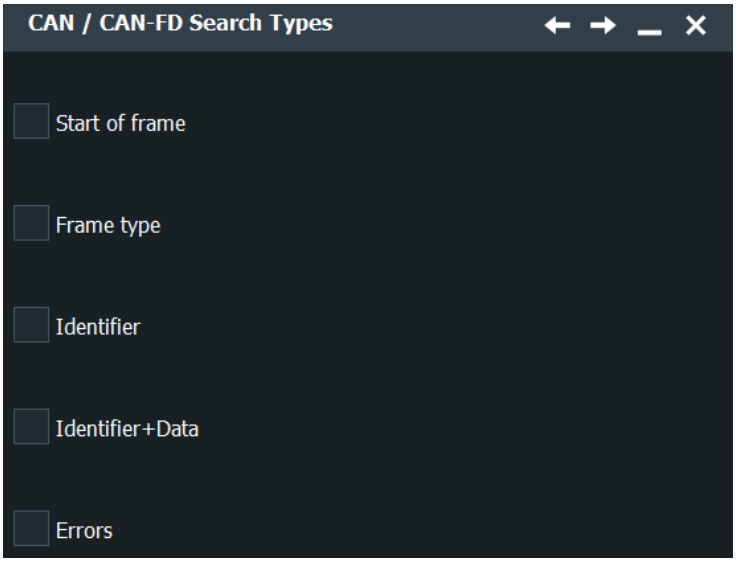
For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

### 13.5.5.1 Search settings

Access: [Bus] > "Shortcuts" tab > "Setup search"

**Search Criteria**
Sets one criterion or an AND-combination of criteria to be searched for. If more than one criterion is selected, all criteria must be fulfilled by a frame for it to be shown in the search results.

If a DBC label list file is applied, an additional criterion "Symbolic" is provided, see Chapter 13.5.6.3, "Symbolic search", on page 583.




"Start of frame" Searches for the first edge of the dominant SOF bit (synchronization bit).

"Frame type" Searches for a specified frame type (data, remote, error, or overload). For data and remote frames, also the identifier format is considered.

For details, see:
- "Frame type" on page 578
- "ID type" on page 578

"Identifier" Searches for a specific message identifier or an identifier range. See "Identifier setup: Condition, Identifier min, Identifier max" on page 578.

"Identifier + Data" Searches for a combination of identifier and data condition. The instrument triggers at the end of the last byte of the specified data pattern.
The identifier conditions are the same as for the "Identifier" search criteria, see "Identifier setup: Condition, Identifier min, Identifier max" on page 578. Data conditions are set with "Data setup: DLC, NDB, Condition, Data min, Data max" on page 579.

"Error condition" Identifies various errors in the frame, see "Error conditions" on page 573.

"Symbolic" The "Symbolic" search criteria is available if a DBC label list file is loaded and applied. It allows you to search for specific data messages, or a signal and its value that appears inside the message. Symbolic search is an alternative to the other search criteria, you can either search for symbolic values or for an AND-combination of the other 5 criteria.
For details, see Chapter 13.5.6.3, "Symbolic search", on page 583.

Remote command:
SEARch:TRIGger:CAN[:SSOFrame] on page 1732
SEARch:TRIGger:CAN:SFTYpe on page 1732
SEARch:TRIGger:CAN:SFIDentifier on page 1732
SEARch:TRIGger:CAN:SIDData on page 1732
SEARch:TRIGger:CAN:SERRor on page 1733
SEARch:TRIGger:CAN:SSYMbolic on page 1750

**Standard**
Selects the CAN standard: "CAN", "CAN FD", or "Any". Use "Any" to search for both CAN and CAN-FD frames. In this case, the search configuration provides all possible settings, for CAN as well as for CAN FD. The setting is available in CAN FD option R&S RTP-K9.

Remote command:
SEARch:TRIGger:CAN:FDATa:STANdard on page 1733

**Frame type**
Selects the frame type.

Remote frames are only available in the CAN protocol.

The frame types are the same as in the CAN trigger setup, see "Frame type" on page 570.

Remote command:
SEARch:TRIGger:CAN:FTYPe on page 1733

**ID type**
Selects the length of the identifier.

| | |
|---|---|
| "11 bit" | Identifier length of the CAN base frame format. The instrument triggers on the sample point of the IDE bit. |
| "29 bit" | Identifier length of the CAN extended frame format. The instrument triggers on the sample point of the RTR bit. |
| "Any" | The ID type is not relevant. If the trigger type is "Identifier + Data", set the "ID type" to "Any" if you want to search only for data. |

Remote command:
SEARch:TRIGger:CAN:ITYPe on page 1734

**Identifier setup: Condition, Identifier min, Identifier max**
The identifier setup consists mainly of the condition and one or two identifier patterns. Additionally, ID type and frame type may qualify the identifier.

The identifier setup settings are the same as in the CAN trigger setup, see "Identifier setup: Condition, Identifier min, Identifier max" on page 571.

FD bits: see "FD bits" on page 579.

Remote command:
SEARch:TRIGger:CAN:ICONdition on page 1734
SEARch:TRIGger:CAN:IMIN on page 1735
SEARch:TRIGger:CAN:IMAX on page 1734

**FD bits**
For standard settings "CAN FD" and "Any", you can search for CAN FD-specific bits. The setting is available in CAN FD option R&S RTP-K9.

For details, see "FD bits" on page 571.

Remote command:
SEARch:TRIGger:CAN:FDATa[:FDF] on page 1740
SEARch:TRIGger:CAN:FDATa:BRS on page 1740
SEARch:TRIGger:CAN:FDATa:ESI on page 1740

**Data setup: DLC, NDB, Condition, Data min, Data max**
The data setup consists of the number of bytes, the condition, and one or two data patterns.

The data setup settings are the same as in the CAN trigger setup, see "Data setup: DLC, NDB, Transfer, Condition, Data min, Data max" on page 572.

The data condition setting is also used for symbolic signal search, see Chapter 13.5.6.3, "Symbolic search", on page 583.

Remote command:
SEARch:TRIGger:CAN:DCONdition on page 1735
SEARch:TRIGger:CAN:DMIN on page 1735
SEARch:TRIGger:CAN:DMAX on page 1736
SEARch:TRIGger:CAN:DLCCondition on page 1736
SEARch:TRIGger:CAN:DLC on page 1736
SEARch:RESult:CAN:FRAMe<m>:NDBYtes? on page 1737

**Data position**
The data position sets the location in the data field where the instrument looks for the specified data pattern. The position can be defined if the data field of the frame is longer than 8 bytes - if DLC ≥9. The setting is available in CAN FD option R&S RTP-K9.

For details, see "Data position" on page 573.

Remote command:
SEARch:TRIGger:CAN:FDATa:DPOPerator on page 1737
SEARch:TRIGger:CAN:FDATa:DPOSition on page 1738
SEARch:TRIGger:CAN:FDATa:DPTO on page 1738

**Error Condition**
Selects the error type to be searched for. You can select one or more error types as search condition.

The error types are the same as in the CAN trigger setup, see "Error conditions" on page 573.

Remote command:
SEARch:TRIGger:CAN:CRCerror on page 1739
SEARch:TRIGger:CAN:BITSterror on page 1738
SEARch:TRIGger:CAN:FORMerror on page 1739
SEARch:TRIGger:CAN:ACKerror on page 1738
SEARch:TRIGger:CAN:FDATa:SCERror on page 1739

#### 13.5.5.2 Search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands are listed in Chapter 24.17.6.5, "Search results", on page 1741.

#### 13.5.5.3 Searching CAN FD data

The "Identifier + Data" search supports the search for data bytes of specific value at a specific location in the data field of a frame. The "Data Pattern" field provides 8 data bytes to define the pattern. For data fields longer than 8 byte, you can define the position where the specified pattern starts.

The following example demonstrates how the data pattern and data position is defined.

**To set up the search**

1. Set the "Source", the signal to be searched: "SerBus". Select the bus that is configured for CAN FD.
2. Set the search criteria:
   a) Select the standard: "CAN FD".
   b) Select the search type: "Identifier + Data".
3. In this example, the identifier does not matter. Set the "ID type = Any".
4. Define the data setup as described in the example.

**Example: Searching for a specific byte anywhere in the frame**

The CAN FD frame has 8 or more data bytes, containing at least one data byte with value = CB anywhere in the data field.

- Set "DLC ≥ 8".
- Set the data pattern: "= CB".

### 13.5.6 Symbolic trigger, decode and search

The R&S RTP can read and apply industry-standard DBC files to the decoded signal and provides settings for symbolic triggering and symbolic search.

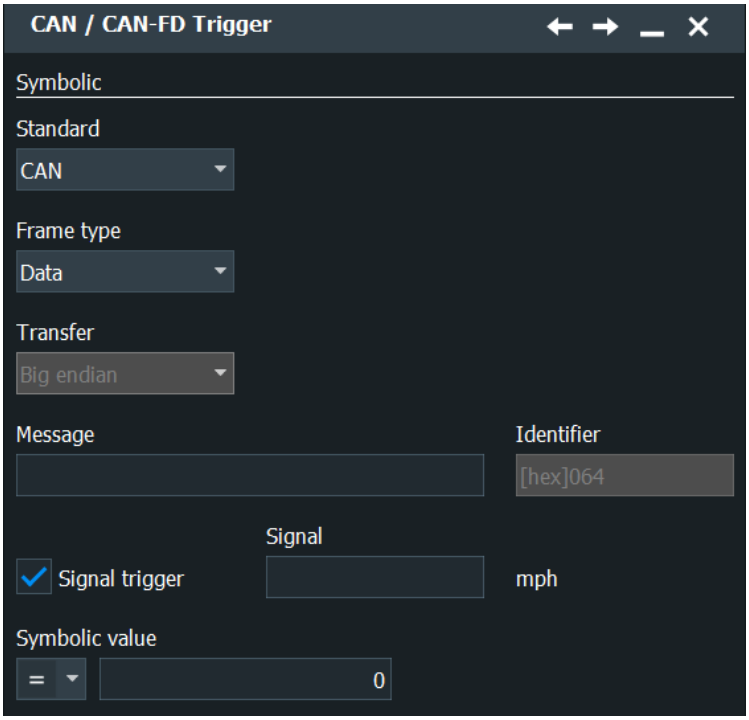
For a description of DBC files, see Chapter 13.5.2.3, "Display settings", on page 565.

#### 13.5.6.1 Symbolic trigger

The "Symbolic" trigger type is available if a DBC label list file is loaded and applied. It allows you to trigger on a specific data message, or a signal and its value that appears inside the message.

Access:

1. Press [Bus].
2. In the "Setup" tab, set "Protocol = CAN/ CAN-FD".
3. In the "Shortcuts" tab, select "Setup trigger".
4. In the trigger dialog, set the "Type" = "Symbolic".




The "Frame type" is automatically set to "Data", and the "Identifier" is shown for information.

Specific settings for the symbolic trigger are:

Message....581
Signal trigger....582
Signal....582
Condition....582
Value, Value min....582
Value max....582

**Message**
Sets the message to be triggered or searched for.

The setting is used by symbolic trigger and symbolic search. Thus, you always trigger on and search for the same message if symbolic trigger and symbolic search is used at the same time.

Remote command:
TRIGger<m>:CAN:SYMBolic:MSGValue on page 1747

**Signal trigger**
Enables the trigger on a specific signal value that is part of the selected message.

Remote command:
TRIGger<m>:CAN:SYMBolic:TSIGnals on page 1747

**Signal**
Sets the signal name to be triggered or searched for.

The setting is used by symbolic trigger and symbolic search. Thus, you always trigger on and search for the same signal if symbolic trigger and symbolic search is used at the same time.

Remote command:
TRIGger<m>:CAN:SYMBolic:SIGValue on page 1748

**Condition**
Sets the operator to set a specific data pattern or symbolic value ("Equal" or "Not equal") or a data range.

Remote command:
TRIGger<m>:CAN:DCONdition on page 1718

**Value, Value min**
Defines the data pattern or selects a symbolic data value.

Remote command:
TRIGger<m>:CAN:SYMBolic:DMIN on page 1748
TRIGger<m>:CAN:SYMBolic:SGEValue on page 1748

**Value max**
The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
TRIGger<m>:CAN:SYMBolic:DMAX on page 1748

#### 13.5.6.2 Symbolic decode waveform

If a DBC file is applied, the symbolic names from the file are applied to the display of the decoded data. The result table lists the signal values and units in the "Symbolic Data" column, and the comb display shows the signal names in addition to the signal values and units.

Remote command:

- BUS<m>:CAN:FRAMe<n>:SDATa? on page 1749

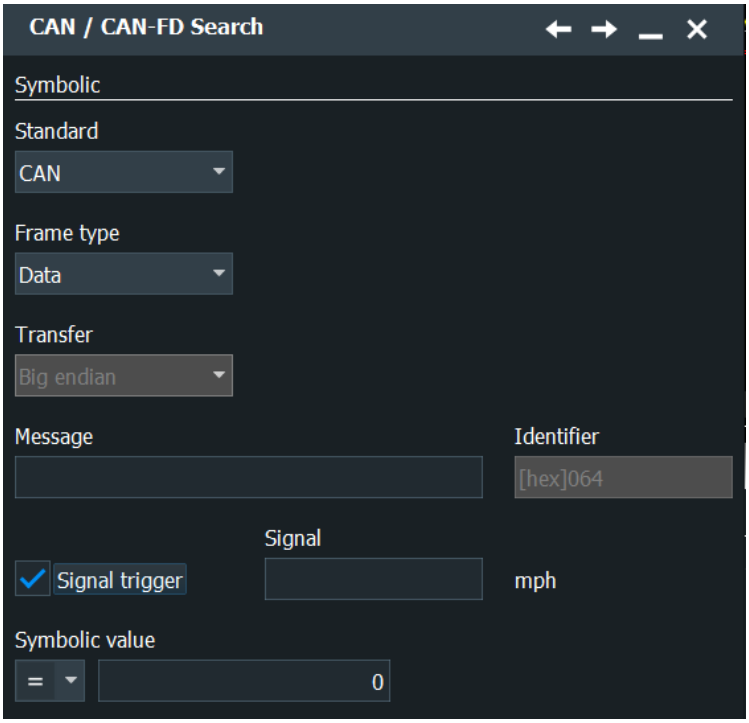
- BUS<m>:CAN:FRAMe<n>:SDEXport? on page 1729

### 13.5.6.3 Symbolic search

Access:

1. Press [Bus].
2. In the "Setup" tab, set "Protocol = CAN/ CAN-FD".
3. In the "Shortcuts" tab, select "Setup search".
4. In the search dialog, set the "Type" = "Symbolic".

If a DBC file is applied, the symbolic search for messages and signal, which are defined in the DBC file, is available. Symbolic search is an alternative to the other search criteria, you can either search for symbolic values or for an AND-combination of the other 5 criteria.




If symbolic search is active, the "Frame type" is automatically set to "Data", and the "Identifier" is shown for information.

**Symbolic**
Enables the symbolic search and disables all other search criteria.

Remote command:
SEARch:TRIGger:CAN:SSYMbolic on page 1750

**Message**
Sets the message to be triggered or searched for.

The setting is used by symbolic trigger and symbolic search. Thus, you always trigger on and search for the same message if symbolic trigger and symbolic search is used at the same time.

Remote command:
`SEARch:TRIGger:CAN:SYMBolic:MSGValue` on page 1750

**Search signals**
Enables the search for a specific signal value that is part of the selected message.

Remote command:
`SEARch:TRIGger:CAN:SYMBolic:SSIGnals` on page 1750

**Signal**
Sets the signal name to be triggered or searched for.

The setting is used by symbolic trigger and symbolic search. Thus, you always trigger on and search for the same signal if symbolic trigger and symbolic search is used at the same time.

Remote command:
`SEARch:TRIGger:CAN:SYMBolic:SIGValue` on page 1751

**Condition**
Sets the operator to set a specific data pattern or symbolic value ("Equal" or "Not equal") or a data range.

This condition is also used to search for data, see "Data setup: DLC, NDB, Condition, Data min, Data max" on page 579.

Remote command:
`SEARch:TRIGger:CAN:DCONdition` on page 1735

**Value, Value (min)**
Defines the data pattern or selects a symbolic data value.

Remote command:
`SEARch:TRIGger:CAN:SYMBolic:DMIN` on page 1751
`SEARch:TRIGger:CAN:SYMBolic:SGEValue` on page 1752

**Value (max)**
The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`SEARch:TRIGger:CAN:SYMBolic:DMAX` on page 1751

#### 13.5.6.4 Symbolic search results

If a DBC file is applied, you can search for symbolic messages and signals as described in Chapter 13.5.6.3, "Symbolic search", on page 583. As ususal, the search results

are shown in a table. You can enable the search zoom window to view the frame with the selected result in more detail.

The figure shows the result of a search for an "EngineData" message and the signal "IdleRunning = Running" inside the message. The result marker (blue line) is set to the start of the frame that fulfills the search condition. The search zoom window Search1 is active. It has been moved to the right until the "IdleRunning" bit with value 0 is visible in the zoom.

Remote command:

- `SEARch:RESult:CAN:FRAMe<m>:SDATa?` on page 1749

## 13.6 LIN (option R&S RTP-K3)

The Local Interconnect Network (LIN) is a simple, low-cost bus system used within automotive network architectures. LIN is usually a subnetwork of a CAN bus. The primary purpose of LIN is the integration of uncritical sensors and actuators with low-bandwidth requirements. Common applications in a motor vehicle are the control of doors, windows, wing mirrors, and wipers.

### 13.6.1 The LIN protocol

This chapter provides an overview of protocol characteristics, frame format, identifiers and trigger possibilities. For detailed information, order the LIN specification on http://www.lin-subbus.org/ (free of charge).

#### LIN characteristics

Main characteristics of LIN are:

- Single-wire serial communications protocol, based on the UART byte-word interface
- Single primary, multiple secondaries - usually up to 12 nodes
- Primary-controlled communication: primary coordinates communication with the LIN schedule and sends identifier to the secondaries
- Synchronization mechanism for clock recovery by secondary nodes without crystal or ceramics resonator

The R&S RTP supports several versions of the LIN standard: v1.3, v2.0, v2.1 and the American SAE J2602.

#### Data transfer

Basic communication concept of LIN:

- Communication in an active LIN network is always initiated by the primary.
- The primary sends a message header including the synchronization break, the synchronization byte, and the message identifier.

- The identified node sends the message response: one to eight data bytes and one checksum byte.
- Header and response form the message frame.

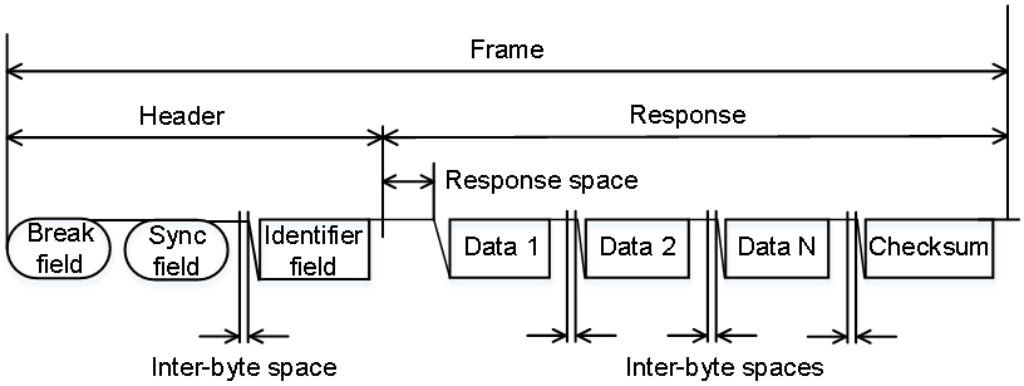



*Figure 13-8: LIN frame with header and response*

The data is transmitted in bytes using the UART byte-word interface without the parity bit. Each byte consists of a start bit, 8 bits and a stop bit.

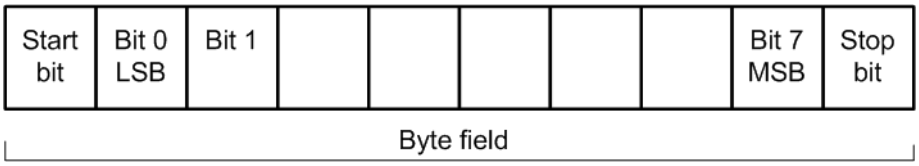



*Figure 13-9: Structure of a byte field*

Data bytes are transmitted LSB first.

The identifier byte consists of 6 bits for the frame identifier and two parity bits. This combination is known as protected identifier.

### Trigger

The R&S RTP can trigger on various parts of LIN frames. The data line must be connected to an input channel, triggering on math and reference waveforms is not possible.

You can trigger on:

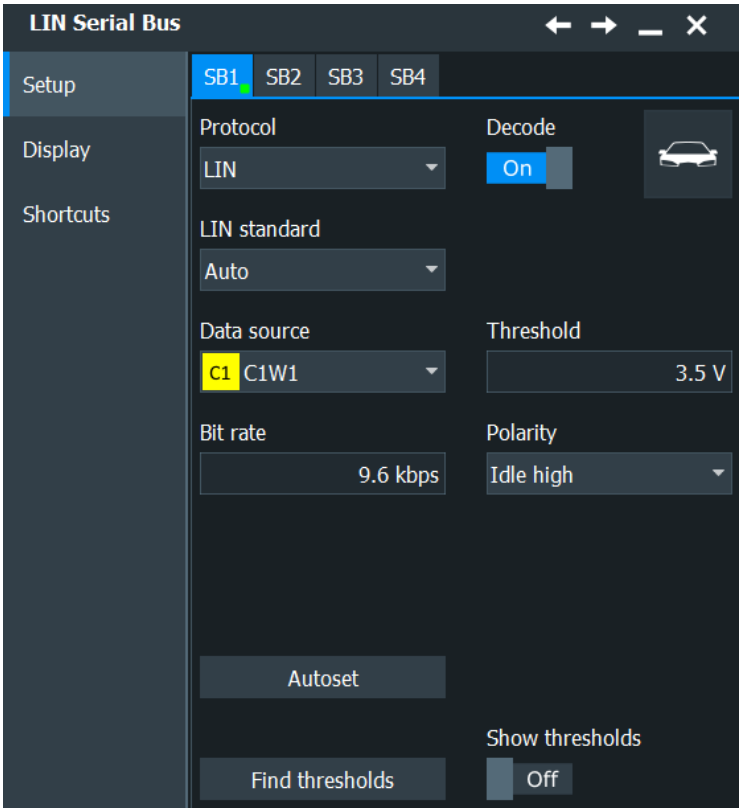
- Frame start (synchronization field)
- Specific identifier or identifier range
- Data pattern in the message
- Wake up signal
- Checksum error (error in data), parity error (error in identifier)

## 13.6.2 LIN configuration

### 13.6.2.1 LIN configuration settings

Access: [Bus] > "LIN" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**LIN standard**
Selects the version of the LIN standard that is used in the DUT. The setting mainly defines the checksum version used during decoding.

The most common version is LIN 2.x. For mixed networks, or if the standard is unknown, set the LIN standard to "Auto".

Remote command:
`BUS<m>:LIN:STANdard` on page 1754

**Data**
Sets the source waveform of the data line.

Usually, the source is one of the analog channels. Reference and math waveforms are available as source if the trigger source is one of the analog channels but not the serial bus.

Alternatively, digital channels can be used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.

For triggering on a serial bus, analog or digital channel sources are required.

Remote command:
`BUS<m>:LIN:DATA:SOURce` on page 1752

**Threshold**
Sets the threshold value for digitization of the signal. If the signal value on the line is higher than the threshold, the signal state is high. Otherwise, the signal state is considered low if the signal value is below the threshold. The interpretation of HIGH and LOW is defined by the Polarity.

There are several ways to set the threshold:

- "Threshold"
  Enter the value directly in the field.
- "Autoset"
  Starts software algorithms for determining the signal threshold levels and bitrate.
- "Find thresholds"
  Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

**Note:** If the sources are digital channels, the same threshold values are used for the parallel and the serial buses. You can set the thresholds either in the parallel bus configuration or in the serial bus configuration.

Remote command:
`BUS<m>:LIN:DATA:THReshold` on page 1753
`BUS<m>:LIN:TECHnology` on page 1753
`BUS<m>:SETReflevels` on page 1638
`BUS<m>:FAUToset` on page 1638

**Bit rate**
Sets the number of transmitted bits per second. The maximum bit rate for LIN is 20 kbit/s.

To select a bit rate from list of predefined values, tap the icon beside the "Bit rate" field. To enter a specific value, open the keypad. The list of predefined values is also available in the keypad.

If the "LIN standard" is "J2602", the bit rate is 10.417 kbit/s and cannot be changed.

Remote command:
`BUS<m>:LIN:BITRate` on page 1754

**Polarity**
Defines the idle state of the bus. The idle state is the recessive state and corresponds to a logic 1.

Remote command:
`BUS<m>:LIN:POLarity` on page 1754

**Show thresholds**
If enabled, the threshold levels are displayed in the diagram.

Remote command:
BUS<m>:THReshold on page 1638

### 13.6.2.2 LIN display

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

**LIN Specific Label Settings**

Label lists are protocol-specific. Label lists for LIN are available in CSV and PTT format.

You can also apply an existing LIN description file.

A LIN label file contains two values for each identifier:

- Identifier value
- Symbolic name for the identifier

**Example of a LIN PTT file**

```
# ------------------------------------------------------------------------------
@FILE_VERSION = 1.0
@PROTOCOL_NAME = lin
# ------------------------------------------------------------------------------
# Labels for LIN protocol
#   Column order: Identifier, Label
# ------------------------------------------------------------------------------
# Labels for standard addresses
0x3F,Temperature
1Ch,Left brake
20h,Right brake
# Following ID is provided as integer
33,Mirror
0x37,Indoor lights
# Labels for reserved addresses
0x3C,Master_Request_Frame
0x3D,Slave_Response_Frame
# ------------------------------------------------------------------------------
```

The LIN description file (ldf) contains information about the assignment of the LIN frames to the nodes. The format and syntax are defined in the LIN specification.

A LIN ldf file contains the following sections:

- Header
- Node section
- Signal section
- Frame section

- Schedule table
- Signal encoding section
- Encoding to signal mapping

**Example of a LIN LDF file**

```
LIN_description_file;
LIN_protocol_version = "2.0";
LIN_language_version = "2.0";
LIN_speed = 19.2 kbps;

Nodes {
    Master: Master1, 10 ms, 0.1 ms;
    Slaves: Slave_Motor, Slave_Sensor;
}

Signals {
  Sensor1:1,0x0,Slave_Sensor, Master1;
  Temperature1:12,0,Slave_Sensor, Master1;
}

Frames {
  Sensor_Frame: {Error,30;
                 Temperature, 20;
                 Status, 0;}
  Master_Frame: {Error,15;
                 Speed, 10;}
}

Schedule_tables {
  Sensor_Frame delay 10.000 ms;
        Master_Frame delay 15.000 ms;
}

Signal_encoding_types {
  Speed {logical_value, 0, "Motor_off";
         logical_value, 1, "Speed1";
         logical_value, 2, "Speed2";
         logical_value, 3, "Speed3";
    }
}

Signal_representation {
  Speed1_encoding:Speed1
  Temperature1_encoding:Temperature1
}
```

Remote commands:

- `BUS<m>:LIN:FRAMe<n>:SDATa?` on page 1763
- `BUS<m>:LIN:FRAMe<n>:SDEXport?` on page 1763

#### 13.6.2.3 Configuring LIN signals

For configuration, assign the lines to the input channels, and define the active states and the logical thresholds.

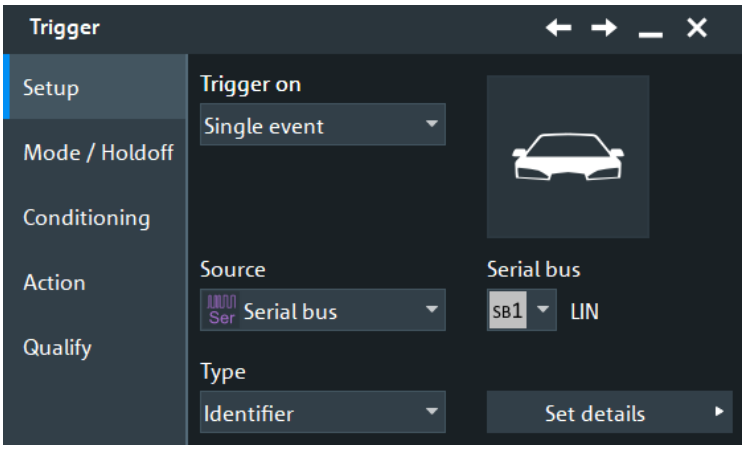
For details on configuration settings, see Chapter 13.6.2.1, "LIN configuration settings", on page 587.

1. Press the [Bus] key on the front panel.
2. At the left hand-side, select the vertical tab of the bus you want to set up.
3. Select the tab of the bus you want to set up, for example "SB1".
4. Tap the "Protocol" button and select the protocol: "LIN".
5. Set the "Data source".
6. Set the logical thresholds.
7. Set the "LIN standard", "Bit rate" and "Polarity" according to your signal.
8. Enable "Decode".

### 13.6.3 LIN trigger

#### 13.6.3.1 LIN trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"



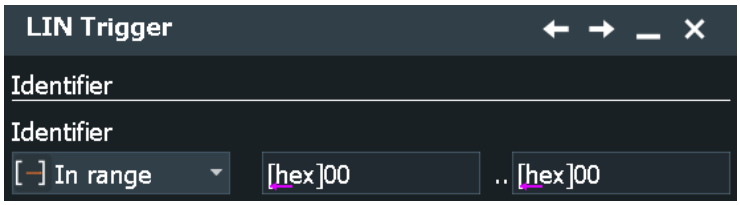
Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

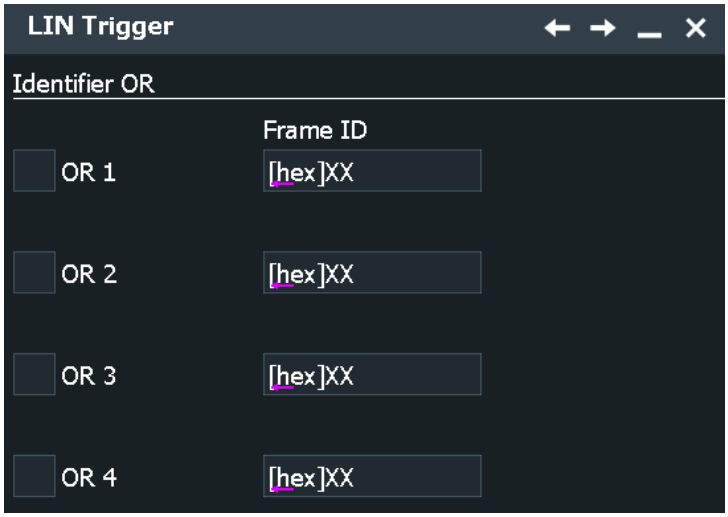
**Trigger type**
Selects the trigger type for LIN analysis.

"Start of frame (Sync)" Triggers on the stop bit of the sync field.

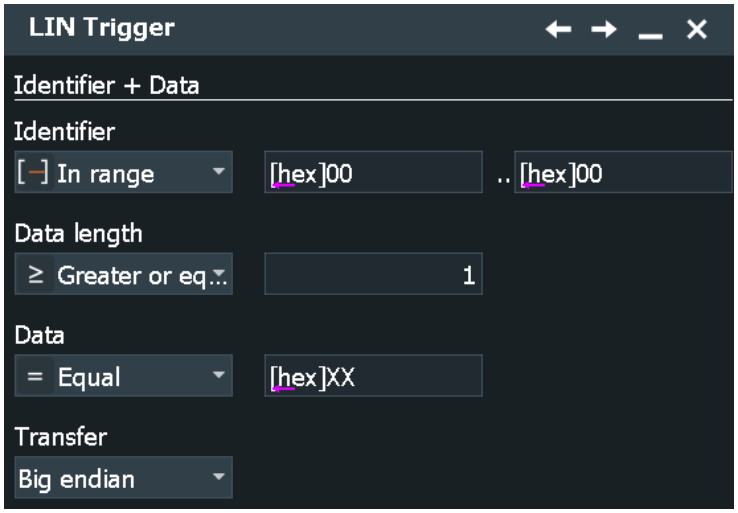
"Identifier" Sets the trigger to one specific identifier or an identifier range. Enter only the 6-bit identifier without parity bits, not the protected identifier. Description of trigger type specific settings: "Identifier setup: Condition, Frame ID min, Frame ID max" on page 593.




"Identifier OR" Sets the trigger to a combination of up to four identifiers. Description of trigger type specific settings: "Identifier OR setup: Monitor, Frame ID" on page 594



"Identifier + Data" Sets the trigger to a combination of identifier and data condition. The instrument triggers at the end of the last byte of the specified data pattern.




The identifier conditions are the same as for the "Identifier" trigger type, see Identifier setup: Condition, Frame ID min, Frame ID max. Data conditions are set with Data setup: Data length, Transfer, Condition, Data min, Data max.

"Wakeup frame" Triggers after a wakeup frame.

"Error condition" Identifies various errors in the frame, see "Error conditions" on page 595.

Remote command:
`TRIGger<m>:LIN:TYPE` on page 1755

**Identifier setup: Condition, Frame ID min, Frame ID max**
The identifier setup consists of the condition and one or two identifier pattern.

"Condition" Defines the operator to set a specific identifier ("Equal" or "Not equal") or an identifier range.

"Frame ID min / Frame ID" Defines the bit pattern of the identifier. Enter only the 6-bit identifier without parity bits, not the protected identifier.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Frame ID max" The second identifier pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:LIN:ICONdition` on page 1756
`TRIGger<m>:LIN:IMIN` on page 1756
`TRIGger<m>:LIN:IMAX` on page 1757

**Identifier OR setup: Monitor, Frame ID**
Sets the trigger to a combination of up to four identifiers. Enter the patterns in the "Frame ID" fields. In binary and hex format, characters 1, 0, and X (do not care) are allowed. For each identifier pattern to be triggered on, enable "Monitor".

Remote command:
`TRIGger<m>:LIN:IDOR<n>:ENABle` on page 1759
`TRIGger<m>:LIN:IDOR<n>[:VALue]` on page 1759

**Data setup: Data length, Transfer, Condition, Data min, Data max**
The data setup consists of the transfer direction, the number of bytes, the condition, and one or two data patterns.

"Transfer" Sets the byte order (endianness) of the data transfer. With "Big endian", the data is analyzed and evaluated in the order of reception. With "Little endian", the instrument reads the complete data, reverses the byte order of the data, and compares it with the specified data word.
According to the standard, LIN data is transmitted in little endian transfer order. The "Little endian" setting allows you to enter the required data word directly into "Data min", and the instrument triggers correctly.

"Data length" Sets the length of the bit pattern to be found, in bytes.

For Big Endian transfer direction, you can trigger on a number of bytes less than the data length of the frame, that means, on the first bytes that are transmitted. For Little Endian transfer direction, the exact number of data bytes in the frame must be set.
Example: The data word to be sent is *12 34 56*, and it is sent little endian by the LIN node. With Data length ≥ 2 and Transfer = Big endian, you trigger on the data of the first two bytes, that is *56 34*. With Data length = 3 and Transfer = Little endian, you trigger on the required data word *12 34 56*.

"Condition" Sets the operator to define a specific data pattern ("Equal" or "Not equal") or an data range.

"Data min" Defines the data pattern. The pattern length is adjusted to the data length setting (and vice versa), maximum is 8 bytes.
Enter the pattern MSB first and with big endian byte order, and set the correct "Transfer" direction. The data is compared byte by byte.
In binary format, use the following characters: 1; 0; or X (do not care).
The use of X is restricted to the operators "Equal" and "Not equal".

"Data max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:LIN:BORDer` on page 1758
`TRIGger<m>:LIN:DLECondition` on page 1758
`TRIGger<m>:LIN:DLENgth` on page 1759
`TRIGger<m>:LIN:DCONdition` on page 1757
`TRIGger<m>:LIN:DMIN` on page 1757
`TRIGger<m>:LIN:DMAX` on page 1757

**Error conditions**
Triggers if one or more of the following errors occur:

- Checksum error
  The checksum verifies the correct data transmission. It is the last byte of the frame response. The checksum includes not only the data but also the protected identifier (PID). To identify checksum errors caused by data, additional settings are required: Enter the bit pattern of the identifier ("Frame ID"), the number of data bytes ("Data length"), and select the used "LIN standard".
  See also: LIN standard.
- Identifier parity error
  Parity bits are the bits 6 and 7 of the identifier. They verify the correct transmission of the identifier.
- Sync error
  Synchronization error

Remote command:
`TRIGger<m>:LIN:CHKSerror` on page 1760
`TRIGger<m>:LIN:ERRPattern` on page 1760
`TRIGger<m>:LIN:CRCDatalen` on page 1761
`TRIGger<m>:LIN:STANdard` on page 1761
`TRIGger<m>:LIN:IPERror` on page 1760
`TRIGger<m>:LIN:SYERror` on page 1759

### 13.6.3.2 Triggering on LIN signals

Prerequisites: An LIN bus is configured, see Chapter 13.2.2.3, "Configuring I²C protocol", on page 529.

1. Press the [Bus] key and select the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Select the serial bus that is set to LIN.
5. Select the "Trigger type".
6. For more complex trigger types, enter the address and/or data conditions: address, identifier, frame ID.
   For details, see Chapter 13.6.3.1, "LIN trigger settings", on page 591.

## 13.6.4 LIN decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".
2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode Results Table

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

The endianness setting ("Transfer") is a trigger setting and not considered for decoding. The binary results of data bytes are displayed MSB first.

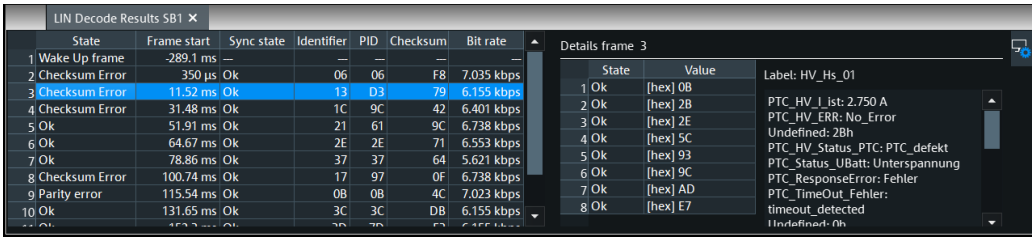



***Table 13-8: Content of the Decode results table***

| Column | Description |
|---|---|
| State | Overall state of the frame. |
| Frame start | Time of frame start |
| Sync state | Result of synchronization |
| Identifier | Identifier value |
| Label | Symbolic label name defined in the label list |
| PID | Protected identifier |
| Values | Value of the data bytes. The data format is selected below the table. |
| Checksum | Checksum value |
| Bit rate | Value of the bit rate |

***Table 13-9: Content of the Details frame table***

| Column | Description |
|---|---|
| State | State of the field |
| Value | Value of the field |
| Label | Symbolic label name defined in the label list |

**Tools**

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

**Remote commands**

Remote commands are described in Chapter 24.17.7.3, "Decode results", on page 1761.

## 13.6.5 Search on decoded LIN data

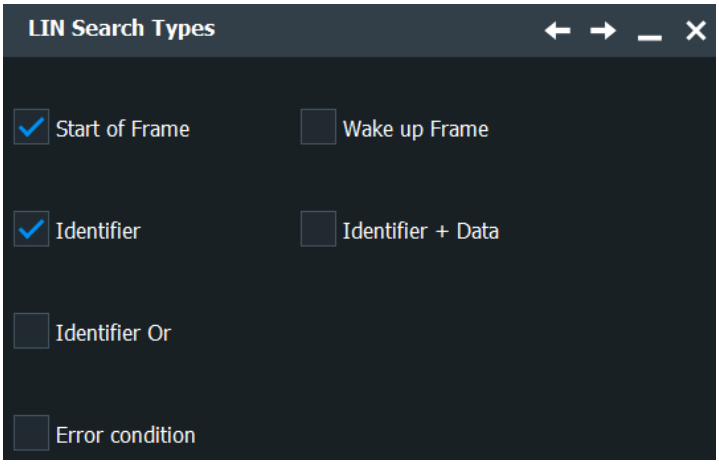
Using the search functionality, you can find various events in the decoded data, the same events which you also can trigger on. Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search "Source" to the serial bus that is configured for the protocol to be analyzed.

See also Chapter 11, "Search functions", on page 449.

### 13.6.5.1 LIN search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"




**Search types**
Sets the type to be searched for.

"Start of frame (Sync)" Searches for the stop bit of the sync field.

"Identifier" Searches for one specific identifier or an identifier range.
See "Identifier setup: Condition, Frame ID min, Frame ID max" on page 598

"Identifier OR" Searches for a combination of up to four identifiers.
See "Identifier OR setup: Monitor, Frame ID" on page 598

| | |
|---|---|
| "Identifier + Data" | Searches for a combination of identifier and data condition. The instrument triggers at the end of the last byte of the specified data pattern.<br>The identifier conditions are the same as for the "Identifier" trigger type, see "Identifier setup: Condition, Frame ID min, Frame ID max" on page 593.<br>Data conditions are set with "Data setup: Data length, Transfer, Condition, Data min, Data max" on page 594. |
| "Wakeup frame" | Searches for wakeup frames. |
| "Error condition" | Identifies various errors in the frame, see "Error conditions" on page 595. |

Remote command:
`SEARch:TRIGger:LIN:SSOFrame` on page 1769
`SEARch:TRIGger:LIN:SFIDentifier` on page 1770
`SEARch:TRIGger:LIN:IDENtifieror` on page 1770
`SEARch:TRIGger:LIN:SIDData` on page 1770
`SEARch:TRIGger:LIN:WUFRame` on page 1771
`SEARch:TRIGger:LIN:SERRor` on page 1770

**Identifier setup: Condition, Frame ID min, Frame ID max**
The identifier setup consists of the condition and one or two frame ID patterns.

The identifier setup settings are the same as in the LIN trigger setup, see "Identifier setup: Condition, Frame ID min, Frame ID max" on page 593.

Remote command:
`SEARch:TRIGger:LIN:ICONdition` on page 1771
`SEARch:TRIGger:LIN:IMIN` on page 1771
`SEARch:TRIGger:LIN:IMAX` on page 1772

**Data setup: Condition, Data min, Data max, Data length, Transfer**
The data setup consists of the transfer direction, the data length, the condition, and one or two data patterns.

The data setup settings are the same as in the LIN trigger setup, see "Data setup: Data length, Transfer, Condition, Data min, Data max" on page 594.

Remote command:
`SEARch:TRIGger:LIN:DCONdition` on page 1773
`SEARch:TRIGger:LIN:DMIN` on page 1773
`SEARch:TRIGger:LIN:DMAX` on page 1773
`SEARch:TRIGger:LIN:DLECondition` on page 1774
`SEARch:TRIGger:LIN:DLENgth` on page 1775
`SEARch:TRIGger:LIN:BORDer` on page 1774

**Identifier OR setup: Monitor, Frame ID**
The identifier OR setup consists of the monitor and frame ID.

The identifier OR setup settings are the same as in the LIN trigger setup, see "Identifier OR setup: Monitor, Frame ID" on page 594

Remote command:
SEARch:TRIGger:LIN:IDOR<m>:ENABle on page 1772
SEARch:TRIGger:LIN:IDOR<m>[:VALue] on page 1772

**Error Condition**
Selects the error type to be searched for. You can select one or more error types as search condition.

The error types are the same as in the LIN trigger setup, see "Error conditions" on page 595.

Remote command:
SEARch:TRIGger:LIN:IPERror on page 1775
SEARch:TRIGger:LIN:SYERror on page 1775
SEARch:TRIGger:LIN:CHKSerror on page 1776
SEARch:TRIGger:LIN:ERRPattern on page 1776
SEARch:TRIGger:LIN:CRCDatalen on page 1776
SEARch:TRIGger:LIN:STANdard on page 1777

#### 13.6.5.2 LIN search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands are listed in Chapter 24.17.7.5, "LIN search results", on page 1777.

## 13.7 MIL-1553 (option R&S RTP-K6)

### 13.7.1 The MIL-STD-1553

The MIL-STD-1553 specification defines the characteristics of a serial data bus originally designed for use in the military avionics. Nowadays it is also used in spacecraft on-board data handling.

The bus is a 2-wire bus that uses differential signals.

A MIL-STD-1553 system consists of the following components:

- Bus Controller (BC): initiates and coordinates the data flow in the system.
- Remote Terminal (RT): interfaces various subsystems with the data bus. A system consists of up to 31 RTs and each RT can have 31 subaddresses.

The subaddresses 0 and 31 refer to a mode code command.

- Bus Monitor (BM) (optional): listens to all messages and can record selected data for real-time or off-line analysis.

The information is transmitted over the bus in defined series of words using Manchester code, where each bit is transmitted as high-low for a logical 1 or a low-high for a logical 0. There are three types of words: command, data and status.

**Command Word**

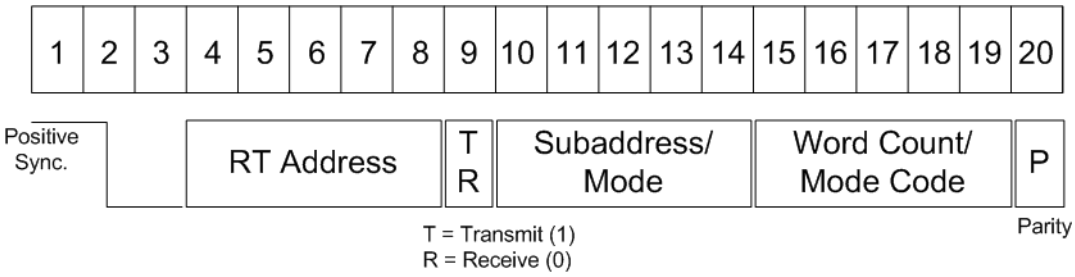



*Figure 13-10: Structure of a command word*

The format of a command word consists of the following parts (see Figure 13-10):

- Sync: an invalid Manchester waveform.
- Remote Terminal (RT) Address: the unique address of the corresponding RT.
- Transmit/Receive (T/R): indicates the action required from the RT.
- Subaddress/Mode Code: indicates the RT subaddress. The subaddresses 0 and 31 signalize the transmission of a mode code.
- Data Word Count /Mode Code: indicates the number of words that are sent/ received by the RT. A maximum of 32 words is allowed. This field may be used for the transmission of the mode code value.
- Parity: checks if there are bit errors during the transmission. The total number of logic 1 bits for the word (sync bits not included) shall be odd.

**Data Word**

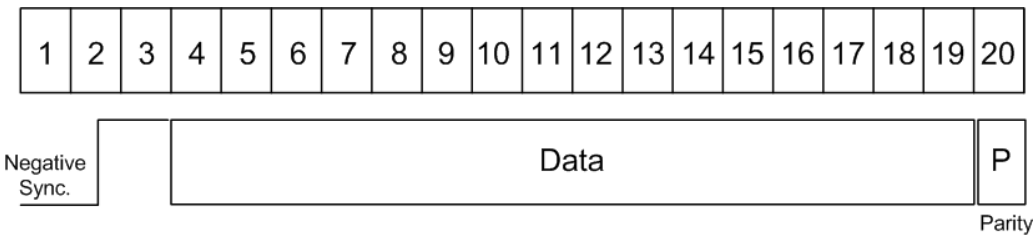



*Figure 13-11: Structure of a data word*

The format of a data word consists of the following parts (see Figure 13-11):

- Sync: an invalid Manchester waveform.
- Data: the transferred information (16 bit).
- Parity: checks if there are bit errors during the transmission. The total number of logic 1 bits for the word (sync bits not included) shall be odd.

**Status Word**

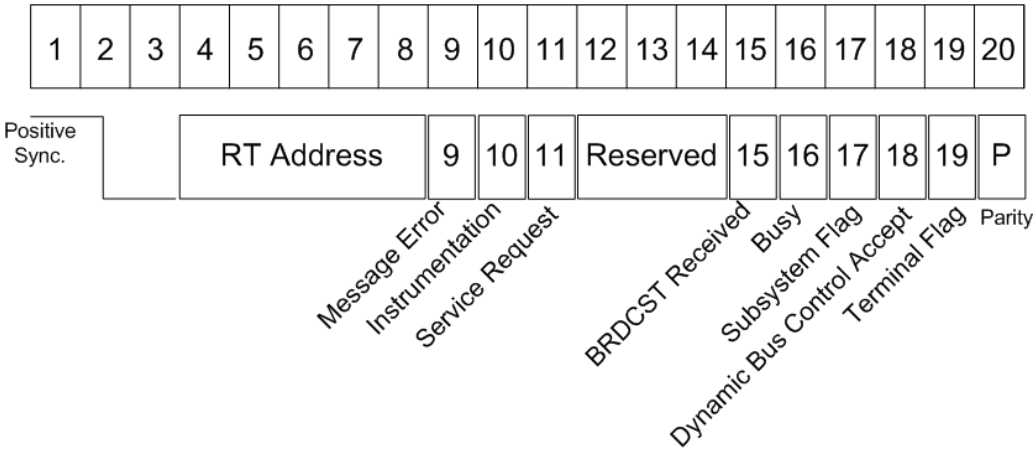



*Figure 13-12: Structure of a status word*

The format of a status word consists of the following parts (see Figure 13-12):

- Sync: an invalid Manchester waveform.
- Remote Terminal (RT) Address: the unique address of the corresponding RT.
- Message error: indicates an error in the command/data word transmission from the BC. A logic 1 indicates presence of a message error and a logic 0 indicates its absence.
- Instrumentation: helps to distinguish between a status word and a command word. The logic state of this bit shall be 0.
- Service Request: indicates that the RT requires service. A logic 1 indicates a presence of a service request and logic 0 indicates its absence.
- Reserved: bits reserved for future uses.
- Broadcast Command: a logic 1 indicates that the preceding valid command word was a broadcast command and a logic 0 that it was not.
- Busy: a busy state indicates that the RT or the subsystem is not able to transfer data. A logic 1 indicates a presence of a busy condition and logic 0 indicates its absence.
- Subsystem Flag: flags a subsystem fault. A logic 1 indicates a presence of a flag and logic 0 indicates its absence.
- Dynamic Bus Control Acceptance: a logic 1 indicates acceptance of a dynamic bus control and a logic 0 a rejection.
- Terminal Flag: flags an RT fault condition. A logic 1 indicates a presence of a flag and logic 0 indicates its absence.
- Parity: checks if there are bit errors during the transmission. The total number of logic 1 bits for the word (sync bits not included) shall be odd.

For comfortable analysis, you can load an editable label list, to interpret transferred numeric values as meaningful text labels.

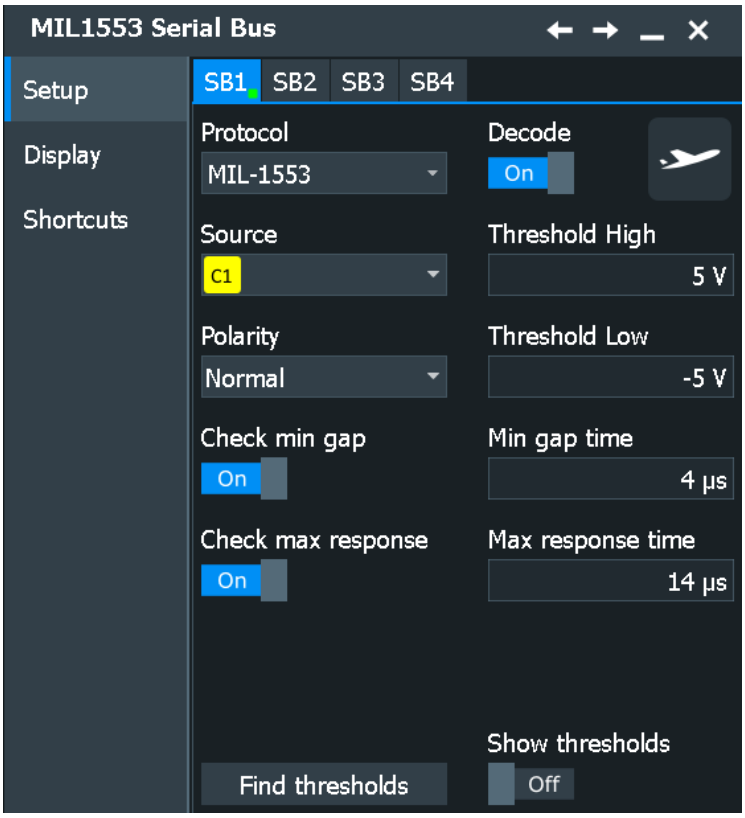
## 13.7.2 MIL-STD-1553 configuration

### 13.7.2.1 MIL-STD-1553 configuration settings

Access: [Bus] > "Decode" tab > "Protocol" = *MIL-STD-1553*

Access: [Bus] > "MIL-1553" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Source**
Sets the source of the selected data line. Waveform 1 of channel signals, math waveforms, and reference waveforms can be used.

For triggering on a serial bus, a channel signal is required.

Remote command:
`BUS<m>:MILStd:SOURce` on page 1783

**Polarity**
Selects the wire on which the bus signal is measured : "Normal" or "Inverted". The setting affects the digitization of the signal.

Remote command:
BUS<m>:MILStd:POLarity on page 1784

**Thresholds**
Threshold values are used for digitization of the signal.

Sets the threshold value for digitization of signals for each line. If the signal value on the line is higher than the threshold, the signal state is high (1 or true for the Boolean logic). Otherwise, the signal state is considered low (0 or false) if the signal value is below the threshold.

| | |
|---|---|
| "Threshold high" and "Threshold low" | Upper and lower threshold levels. You can enter the values directly in the fields. |
| "Find thresholds" | Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude. |

Remote command:
BUS<m>:MILStd:THReshold:HIGH on page 1785
BUS<m>:MILStd:THReshold:LOW on page 1785
BUS<m>:MILStd:PRESet on page 1785
BUS<m>:SETReflevels on page 1638

**Check min gap**
Checks and sets a value for the intermessage gap between the last bit of a message and the following command word sync. The time is measured between the mid bit zero crossings. According to the standard, the minimum idle time is 4 µs.

The minimum gap time is relevant for protocol configuration and error trigger.

If "Check min gap" is enabled, the instrument detects the specified gap during decoding. If the trigger type "Error condition" is selected in addition, the instrument triggers when the gap is shorter then specified.

Remote command:
BUS<m>:MILStd:MINGap:BITS on page 1784
BUS<m>:MILStd:MINGap:SELect on page 1784
TRIGger<m>:MILStd:MINGap:BITS on page 1794
TRIGger<m>:MILStd:MINGap:SELect on page 1795

**Check max response**
Checks and sets a value for the maximum response time between the last bit of a word and the following status word sync. The time is measured between the mid bit zero crossings. According to the standard, the RT shall respond to a valid command word within the time period of 4 to 12 µs.

The max response time is relevant for protocol configuration and error trigger.

If "Check max response" is enabled, the instrument detects the specified gap during decoding. If the trigger type "Error condition" is selected in addition, the instrument triggers when the response time is longer than specified.

Remote command:
BUS<m>:MILStd:MAXResponse:BITS on page 1783
BUS<m>:MILStd:MAXResponse:SELect on page 1784
TRIGger<m>:MILStd:MAXResponse:BITS on page 1794
TRIGger<m>:MILStd:MAXResponse:SELect on page 1794

**Show thresholds**
If enabled, the threshold levels are displayed in the diagram.

Remote command:
BUS<m>:THReshold on page 1638

### 13.7.2.2 Display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

**MIL Specific Label Settings**

Label lists are protocol-specific. A MIL-STD-1553 label file contains four values for each identifier:

- "RTA": hexadecimal remote terminal address value
- "Sub Addr": hexadecimal sub address value
- "Sub Address Label Name": the label name corresponding to the value of the sub-address.
- "Symbolic label": symbolic name of addressed device, specifying the device function, and the label of the sub address.

**Example: MIL PTT file**

```
# ------------------------------------------------------------
# Labels for MIL.1553 protocol
# Column order: RT address, RT label, Subaddress, Subaddress Label
# ------------------------------------------------------------------------
@PROTOCOL_NAME = mil1553
0Ah,Engine,01h,Thrust
03h,Main panel,07h,Altimeter
03h,Main panel,01h,Speed
0Eh,Only RTA
```

### 13.7.2.3 Configuring MIL-STD-1553

For configuration assign the line to the input channel, set the threshold and the timing conditions.

For details on configuration settings, see Chapter 13.7.2.1, "MIL-STD-1553 configuration settings", on page 602.

1. Press the [Bus] key on the front panel.

2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, select the protocol: "MIL-STD-1553".
4. Tap the "Polarity" button, and select the waveform of the data line.
5. Set the logical thresholds.
6. If necessary, tap the "Check min gap" button to select it and set the minimum gap time.
7. If necessary, tap the "Check max response" button to select it and set the maximum response time.

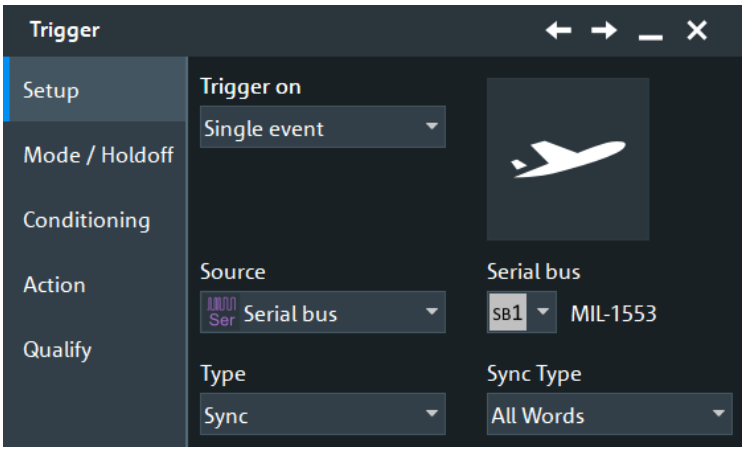
## 13.7.3 MIL-STD-1553 trigger

### 13.7.3.1 Trigger settings MIL-STD-1553

Access: [Bus] > "Shortcuts" tab > "Setup trigger"

Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

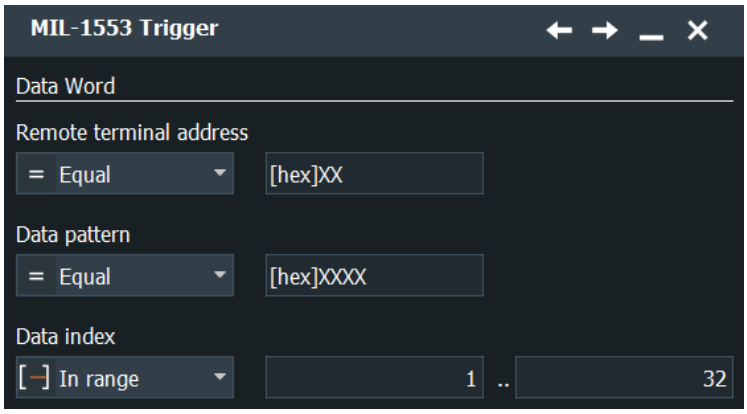



**Trigger Type**
Selects the trigger type for MIL-STD-1553 analysis.

"Sync" Triggers on a sync impulse.

"Word" Triggers on the selected word type.

"Data Word" Triggers on a specified data word or data word range.

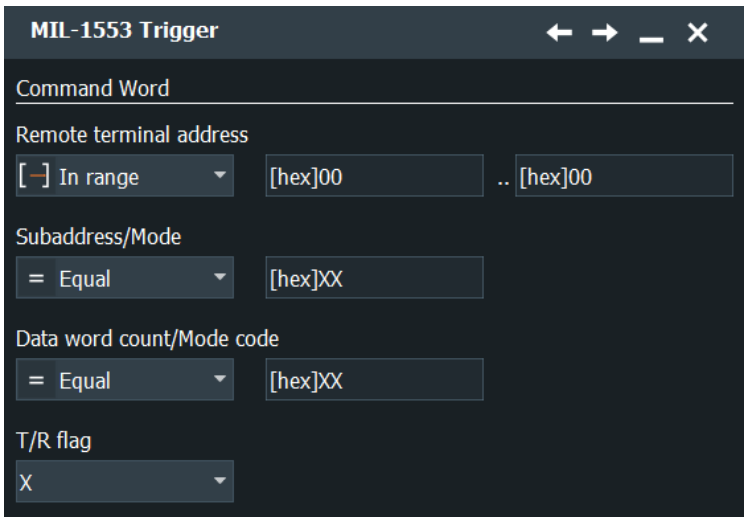



"Command/ Status Word"

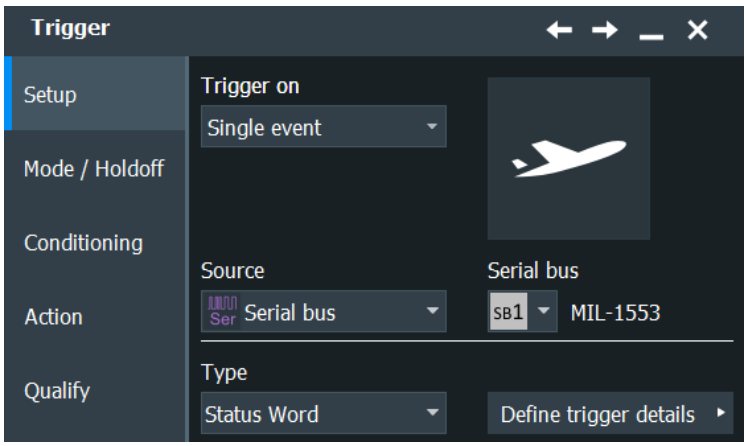
Triggers on a specified command word or on a status word.

"Command Word"

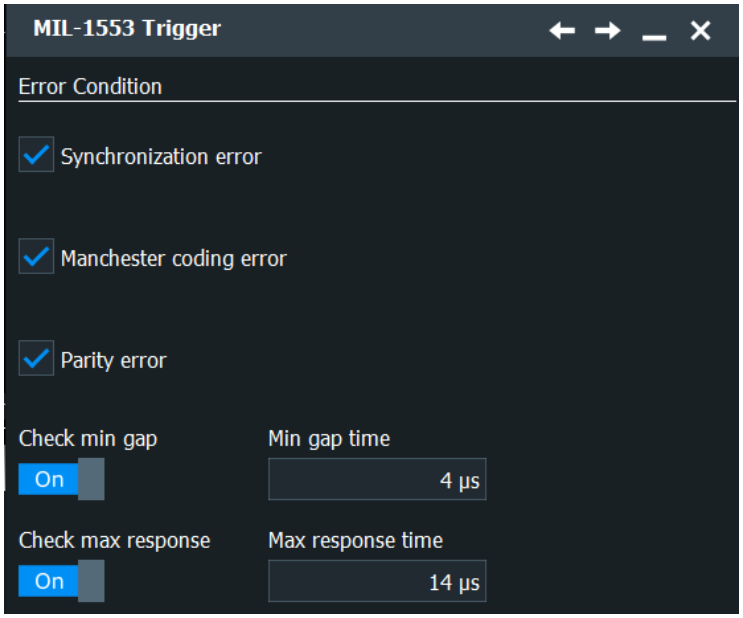
Triggers on a specified command word.



"Status Word" Triggers on a specified status word.




"Error Condition" Triggers on any combination of protocol errors.




Remote command:
TRIGger<m>:MILStd:TYPE on page 1787

**Sync Type / Word Type**
Triggers on a sync impulse or word type. You can select to trigger on "Command/ Status", on "All" or on "Data" sync pulses / word types.

Remote command:
TRIGger<m>:MILStd:TPSPecifier on page 1797

**Remote Terminal Address**
The RTA setup consists of the condition and one or two RTA patterns.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"RTA Min/ RTA" Specifies the value or sets the start value of a range.

"RTA Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger<m>:MILStd:CDST:RCONdition` on page 1787
`TRIGger<m>:MILStd:CDST:RMAX` on page 1788
`TRIGger<m>:MILStd:CDST:RMIN` on page 1788
`TRIGger<m>:MILStd:CMD:RCONdition` on page 1787
`TRIGger<m>:MILStd:CMD:RMAX` on page 1788
`TRIGger<m>:MILStd:CMD:RMIN` on page 1788
`TRIGger<m>:MILStd:DATA:RCONdition` on page 1787
`TRIGger<m>:MILStd:DATA:RMAX` on page 1788
`TRIGger<m>:MILStd:DATA:RMIN` on page 1788

**Data Pattern**
Sets the data to be triggered on. The trigger data setup consists of a comparison condition and one or two data patterns.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"Data Min/ Data" Specifies the value or sets the start value of a range.

"Data Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger<m>:MILStd:DATA:DCONdition` on page 1791
`TRIGger<m>:MILStd:DATA:DMAX` on page 1792
`TRIGger<m>:MILStd:DATA:DMIN` on page 1792

**Data Index**
The MIL-MIL-STD-1553 standard defines the length of a message to a series of up to 32 words. Data index sets the range within this series of the data words that is considered for the analysis. The data index setup consists of the condition and one or two data index values.

"Condition" Sets the comparison condition to a specific value or a range.

"Index Min/Index" Specifies the value or sets the start value of a range.

"Index Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
TRIGger<m>:MILStd:DATA:ICONdition on page 1792
TRIGger<m>:MILStd:DATA:IMAX on page 1793
TRIGger<m>:MILStd:DATA:IMIN on page 1793

**11-Bit Information**
The 11-Bit information sets bits 9 to 19 if there is a command or status word. The 11-Bit information consists of the condition and one or two 11-Bit information patterns.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"Info Min/Info" Specifies the value or sets the start value of a range.

"Info Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
TRIGger<m>:MILStd:CDST:ICONdition on page 1791
TRIGger<m>:MILStd:CDST:IMAX on page 1791
TRIGger<m>:MILStd:CDST:IMIN on page 1791

**Subaddress/ Mode**
The subaddress/mode setup consists of the condition and one or two subaddress/ mode patterns.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"Subaddress Min / Subaddress/Mode"
Specifies the value or sets the start value of a range.

"Subaddress Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
TRIGger<m>:MILStd:CMD:SCONdition on page 1789
TRIGger<m>:MILStd:CMD:SMAX on page 1790
TRIGger<m>:MILStd:CMD:SMIN on page 1790

**Data Word Count/Mode Code**
The data word count/ mode code setup consists of the condition and one or two patterns.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"Word Count Min/ Count Code"
Specifies the value or sets the start value of a range.

"Word Count Max"
Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
TRIGger<m>:MILStd:CMD:CCONdition on page 1788
TRIGger<m>:MILStd:CMD:CMAX on page 1789
TRIGger<m>:MILStd:CMD:CMIN on page 1789

**T/R (Transmit/receive)**
Toggles the data direction of the selected command: 1 (transmit), 0 (receive), or X (either).

Remote command:
TRIGger<m>:MILStd:CMD:TR on page 1790

**Status Flags**
Specifies the values of the status flags. You can use the following characters: 1; 0; or X (do not care).

For details, see "Status Word" on page 601.

Remote command:
TRIGger<m>:MILStd:STATus:BCReceived on page 1795
TRIGger<m>:MILStd:STATus:BUSY on page 1795
TRIGger<m>:MILStd:STATus:DBCaccept on page 1795
TRIGger<m>:MILStd:STATus:INSTrument on page 1795
TRIGger<m>:MILStd:STATus:MERRor on page 1796
TRIGger<m>:MILStd:STATus:SREQuest on page 1796
TRIGger<m>:MILStd:STATus:SUBSystem on page 1796
TRIGger<m>:MILStd:STATus:TERMinal on page 1796

**Error Condition**
Specify the error conditions to be triggered on.

**Synchronization Error ← Error Condition**
Triggers if a sync impulse does not fulfill the technical requirements or when the transmission is not valid.

Remote command:
TRIGger<m>:MILStd:ERRor:SYNC on page 1794

**Manchester Coding Error ← Error Condition**
Triggers if there is an error in the Manchester coding of the signal.

Remote command:
TRIGger<m>:MILStd:ERRor:MANChester on page 1793

**Parity Error ← Error Condition**
Checks the parity of every word and triggers if the parity is even.

Remote command:
TRIGger<m>:MILStd:ERRor:PARity on page 1793

**Check min gap ← Error Condition**
Checks and sets a value for the intermessage gap between the last bit of a message and the following command word sync. The time is measured between the mid bit zero crossings. According to the standard, the minimum idle time is 4 µs.

The minimum gap time is relevant for protocol configuration and error trigger.

If "Check min gap" is enabled, the instrument detects the specified gap during decoding. If the trigger type "Error condition" is selected in addition, the instrument triggers when the gap is shorter then specified.

Remote command:
`BUS<m>:MILStd:MINGap:BITS` on page 1784
`BUS<m>:MILStd:MINGap:SELect` on page 1784
`TRIGger<m>:MILStd:MINGap:BITS` on page 1794
`TRIGger<m>:MILStd:MINGap:SELect` on page 1795

**Check max response ← Error Condition**
Checks and sets a value for the maximum response time between the last bit of a word and the following status word sync. The time is measured between the mid bit zero crossings. According to the standard, the RT shall respond to a valid command word within the time period of 4 to 12 µs.

The max response time is relevant for protocol configuration and error trigger.

If "Check max response" is enabled, the instrument detects the specified gap during decoding. If the trigger type "Error condition" is selected in addition, the instrument triggers when the response time is longer than specified.

Remote command:
`BUS<m>:MILStd:MAXResponse:BITS` on page 1783
`BUS<m>:MILStd:MAXResponse:SELect` on page 1784
`TRIGger<m>:MILStd:MAXResponse:BITS` on page 1794
`TRIGger<m>:MILStd:MAXResponse:SELect` on page 1794

### 13.7.3.2 Triggering on MIL-STD-1553

Prerequisites: A bus is configured for the MIL-STD-1553 signal to be analyzed.

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Select the serial bus that is set to MIL-STD-1553.
5. Select the "Trigger type".
6. For more complex trigger types, enter the data pattern conditions.
   For details, see Chapter 13.7.3.1, "Trigger settings MIL-STD-1553", on page 605.

### 13.7.4 MIL-STD-1553 decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".
2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

#### Decode Results Table

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

***Table 13-10: Content of the decode result table***

| Column | Description |
|---|---|
| State | Overall state of the word |
| Start | Time of word start in relation to the trigger point |
| Stop | Time of word stop in relation to the trigger point |
| Type | Word type |
| RTA | Remote terminal address |
| Info | The hexadecimal value of the 9th to 1th bit of a command/status word |
| Label | Symbolic label name defined in the label list |
| Data | The values of the data bytes |

#### Remote commands

Remote commands to retrieve decode results are described in Chapter 24.17.8.3, "Decode results", on page 1797.

### 13.7.5 Search on decoded MIL data

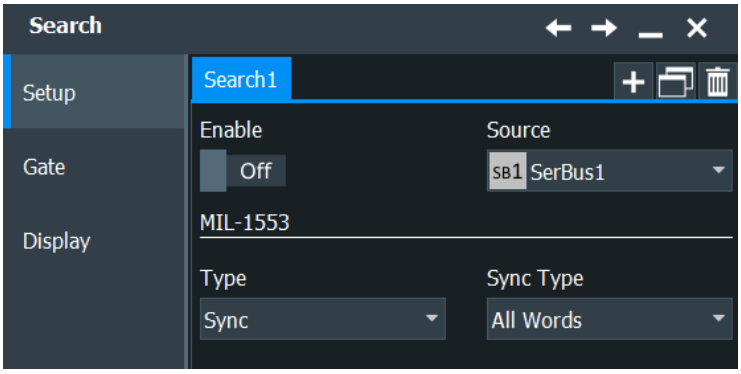
Using the search functionality, you can find various events in the decoded data, the same events which you also can trigger on. Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search "Source" to the serial bus that is configured for the protocol to be analyzed.

See also Chapter 11, "Search functions", on page 449.

#### 13.7.5.1 MIL search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"




**Type**
The search criterion is defined by "Type". All trigger types are also available for search.

Additional search parameters are provided under "Detailed Search Parameter Setup".

"Sync" Searches for a sync impulse.

"Word" Searches for the selected word type.

"Data Word" Searches for the specified data word. Additional search parameters: remote terminal address, data pattern and data index.

"Command / Status Word" Searches for command or status words. Additional search parameters: remote terminal address, and 11-bit information.

"Command Word" Searches for a command word. Additional search parameters: remote terminal address, subaddress / mode, and data word count / mode code.

"Status Word" Searches for a status word. Additional search parameters: remote terminal address, and status flags.

"Error condition" Identifies various errors in the frame, see "Error Condition" on page 610.

Remote command:
`SEARch:TRIGger:MILStd:TYPE` on page 1801

**Sync Type / Word Type**
Searches for a sync impulse/ word type. You can search for "Command/Status", "All" or "Data" sync pulses/ word types.

Remote command:
`SEARch:TRIGger:MILStd:TPSPecifier` on page 1807

**Remote terminal address setup: Condition, RTA**
The remote terminal address setup consists of the condition and one or two RTA patterns.

The RTA setup settings are the same as in the MIL trigger setup, see "Remote Terminal Address" on page 607.

Remote command:
`SEARch:TRIGger:MILStd:CDST:RCONdition` on page 1802
`SEARch:TRIGger:MILStd:CMD:RCONdition` on page 1802
`SEARch:TRIGger:MILStd:DATA:RCONdition` on page 1802
`SEARch:TRIGger:MILStd:CDST:RMIN` on page 1802
`SEARch:TRIGger:MILStd:CMD:RMIN` on page 1802
`SEARch:TRIGger:MILStd:DATA:RMIN` on page 1802
`SEARch:TRIGger:MILStd:CDST:RMAX` on page 1803
`SEARch:TRIGger:MILStd:CMD:RMAX` on page 1803
`SEARch:TRIGger:MILStd:DATA:RMAX` on page 1803

**Data pattern setup: Condition, Data min, Data max**
The data pattern setup consists of the condition and one or two data patterns.

The data pattern setup settings are the same as in the MIL trigger setup, see "Data Pattern" on page 608.

Remote command:
`SEARch:TRIGger:MILStd:DATA:DCONdition` on page 1803
`SEARch:TRIGger:MILStd:DATA:DMIN` on page 1804
`SEARch:TRIGger:MILStd:DATA:DMAX` on page 1804

**Data index setup: Condition, Index min, Index max**
The data index setup consists of the condition and one or two index patterns.

The data index setup settings are the same as in the MIL trigger setup, see "Data Index" on page 608.

Remote command:
`SEARch:TRIGger:MILStd:DATA:ICONdition` on page 1805
`SEARch:TRIGger:MILStd:DATA:IMIN` on page 1805
`SEARch:TRIGger:MILStd:DATA:IMAX` on page 1805

**11-Bit information setup: Condition, Info min, Info max**
The 11-bit information setup consists of the condition and one or two 11-bit information patterns.

The 11-bit information setup settings are the same as in the MIL trigger setup, see "11-Bit Information" on page 609.

Remote command:
`SEARch:TRIGger:MILStd:CDST:ICONdition` on page 1803
`SEARch:TRIGger:MILStd:CDST:IMIN` on page 1804
`SEARch:TRIGger:MILStd:CDST:IMAX` on page 1804

**Subaddress / Mode setup: Condition, Subaddress min, Subaddress max**
The subaddress/mode setup consists of the condition and one or two subaddress/ mode patterns.

The subaddress/mode setup settings are the same as in the MIL trigger setup, see "Subaddress/ Mode" on page 609.

Remote command:
`SEARch:TRIGger:MILStd:CMD:SCONdition` on page 1803
`SEARch:TRIGger:MILStd:CMD:SMIN` on page 1804
`SEARch:TRIGger:MILStd:CMD:SMAX` on page 1804

**Data word count / Mode code setup: Condition, Word count min, Word count max**
The data word count/mode code setup consists of the condition and one or two patterns.

The subaddress/mode setup settings are the same as in the MIL trigger setup, see "Data Word Count/Mode Code" on page 609.

Remote command:
`SEARch:TRIGger:MILStd:CMD:CCONdition` on page 1803
`SEARch:TRIGger:MILStd:CMD:CMIN` on page 1804
`SEARch:TRIGger:MILStd:CMD:CMAX` on page 1804

**T/R (Transmit/receive)**
Specifies the data direction of the selected command.

For details, see "T/R (Transmit/receive)" on page 610.

Remote command:
`SEARch:TRIGger:MILStd:CMD:TR` on page 1806

**Status flags setup**
Specifies the values (X, 0, 1) of the status flags.

The status flags setup settings are the same as in the MIL trigger setup, see "Status Flags" on page 610.

Remote command:
`SEARch:TRIGger:MILStd:STATus:BCReceived` on page 1806
`SEARch:TRIGger:MILStd:STATus:BUSY` on page 1806
`SEARch:TRIGger:MILStd:STATus:DBCaccept` on page 1806
`SEARch:TRIGger:MILStd:STATus:INSTrument` on page 1806
`SEARch:TRIGger:MILStd:STATus:MERRor` on page 1806
`SEARch:TRIGger:MILStd:STATus:SREQuest` on page 1807
`SEARch:TRIGger:MILStd:STATus:SUBSystem` on page 1807
`SEARch:TRIGger:MILStd:STATus:TERMinal` on page 1807

**Error Condition**
Selects the error type to be searched for. You can select one or more error types as search condition.

The error types are the same as in the MIL trigger setup, see "Error Condition" on page 610

Remote command:
SEARch:TRIGger:MILStd:ERRor:MANChester on page 1806
SEARch:TRIGger:MILStd:ERRor:PARity on page 1806
SEARch:TRIGger:MILStd:ERRor:SYNC on page 1806
SEARch:TRIGger:MILStd:ERRor:TIMing on page 1806

#### 13.7.5.2 MIL search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

The serach results are the same as Chapter 13.7.4, "MIL-STD-1553 decode results", on page 612

Remote commands:

- SEARch:RESult:MILStd:WCOunt? on page 1808
- SEARch:RESult:MILStd:WORD<m>:INFO? on page 1810
- SEARch:RESult:MILStd:WORD<m>:RTADdress? on page 1809
- SEARch:RESult:MILStd:WORD<m>:STARt? on page 1808
- SEARch:RESult:MILStd:WORD<m>:STATus? on page 1808
- SEARch:RESult:MILStd:WORD<m>:STOP? on page 1809
- SEARch:RESult:MILStd:WORD<m>:SYMBol? on page 1809
- SEARch:RESult:MILStd:WORD<m>:TYPE? on page 1808

## 13.8 ARINC 429 (option R&S RTP-K7)

### 13.8.1 ARINC 429 basics

The ARINC 429 is a specification that defines the characteristics of an avionic data bus used on commercial and transport aircraft.

In an ARINC 429 system, a single transmitter/source is connected to 1 to 20 receivers/sinks on one twisted wire pair. The bus uses differential signals. The ARINC 429 standard uses a simplex communication - data may be transmitted in only one direction. The information is transmitted over the bus in defined series of words.

**Word Format**

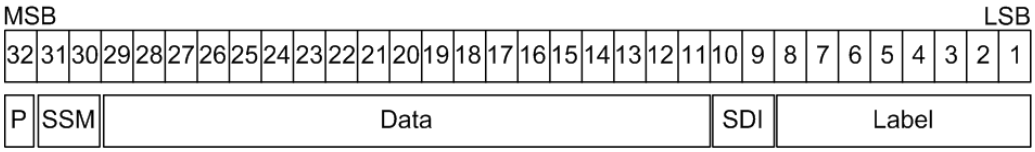



*Figure 13-13: Structure of an ARINC 429 word*

An ARINC 429 word is 32-bits and consists of the following parts (see Figure 13-13):

- Parity: the most significant bit (MSB). Checks if there are bit errors during the transmission. The total number of logic 1 bits for the word shall be odd.
- Sign/Status Matrix (SSM): the value of these bits depends on the data type. It may be used to report the status of hardware equipment.
- Data:
  - Binary (BNR): stores the data as a binary number.
  - Binary Coded Decimal (BCD): uses 4 data field bits to represent a decimal digit.
  - Discrete data: a combination of BNR and/ or BCD or individual bits that express specific equipment conditions.
  - Maintenance data and acknowledgment
  - Williamsburg / Buckhorn protocol: a bit-oriented protocol that is used for file transfer.
- Source/Destination Identifier (SDI): indicates the intended receiver or the transmitting subsystem.
- Label: gives information about the word's data type.

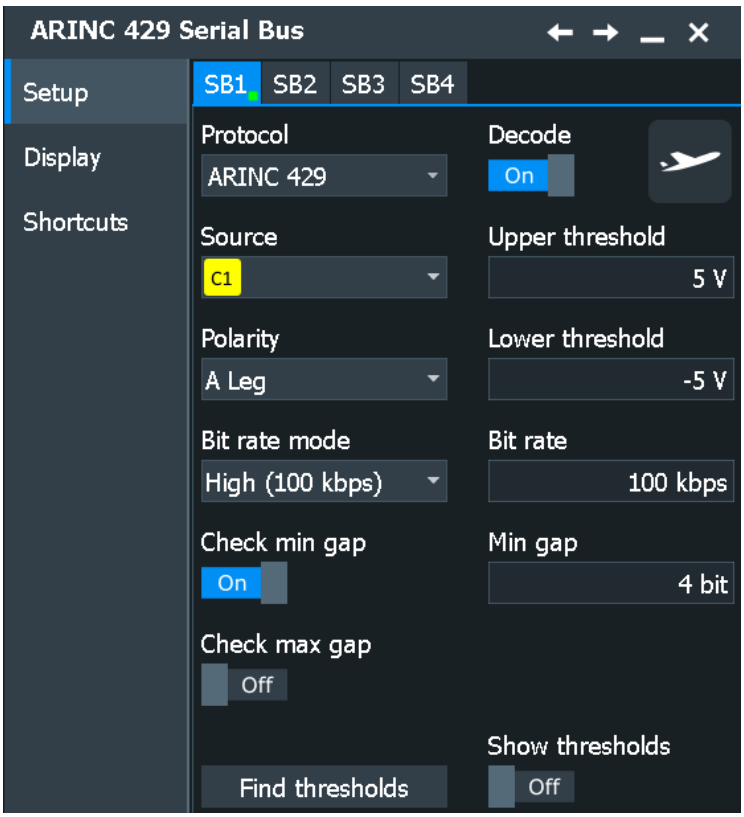
For comfortable analysis, you can load an editable label list, to interpret transferred numeric values as meaningful text labels.

## 13.8.2 ARINC 429 configuration

### 13.8.2.1 ARINC 429 configuration settings

Access: [Bus] > "ARINC 429" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Data**
Sets the source of the selected data line. Usually, the source is one of the analog channels. Reference and math waveforms are only available if the trigger source is one of the input channels but not the serial bus.

For triggering on a serial bus, a channel signal is required.

Remote command:
`BUS<m>:ARINc:SOURce` on page 1811

**Polarity**
Selects the wire on which the bus signal is measured : "A Leg" or "B Leg". The setting affects the digitization of the signal.

Remote command:
`BUS<m>:ARINc:POLarity` on page 1813

**Thresholds**
Sets the threshold value for digitization of the data signal. If the signal value on the line is higher than the threshold, the signal state is high (1 or true for the Boolean logic). Otherwise, the signal state is considered low (0 or false) if the signal value is below the threshold.

There are several ways to set the thresholds:

- "Upper threshold" and "Lower threshold"
  Upper and lower threshold levels. You can enter the values directly in the fields.

- "Find thresholds"

Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

Remote command:
BUS<m>:ARINc:THReshold:HIGH on page 1813
BUS<m>:ARINc:THReshold:LOW on page 1813
BUS<m>:ARINc:PRESet on page 1813
BUS<m>:SETReflevels on page 1638

**Bit Rate**
Selects the number of transmitted bits per second. The value can be set to high speed (100 kbps) or low speed (12.0- 14.5 kbps).

Remote command:
BUS<m>:ARINc:BRValue on page 1811
BUS<m>:ARINc:BRMode on page 1811

**Timing: Min gap, Max gap**
Defines the idle time between two words, which is needed for word synchronization. The beginning of the first bit after the gap marks the start of a new word.

You can define a minimum idle time "Min gap", and/or a maximum time "Max gap". The standard defines a minimum of 4-bit times to separate two subsequent words.

Timing settings are relevant for protocol configuration and error trigger.

If "Min gap" and/or "Max gap" are enabled, the instrument detects the specified gaps during decoding. If the trigger type "Error condition" is selected in addition, the instrument triggers when the gap is shorter or longer than the specified gaps, respectively.

Remote command:
BUS<m>:ARINc:MAXGap:BITS on page 1812
BUS<m>:ARINc:MAXGap:SELect on page 1811
BUS<m>:ARINc:MINGap:BITS on page 1812
BUS<m>:ARINc:MINGap:SELect on page 1812
TRIGger<m>:ARINc:MINGap:BITS on page 1817
TRIGger<m>:ARINc:MINGap:SELect on page 1816
TRIGger<m>:ARINc:MAXGap:BITS on page 1817
TRIGger<m>:ARINc:MAXGap:SELect on page 1817

**Show thresholds**
If enabled, the threshold levels are displayed in the diagram.

Remote command:
BUS<m>:THReshold on page 1638

#### 13.8.2.2 Display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

**ARINC 429 Specific Label Settings**

Label lists are protocol-specific. An ARINC 429 label file contains two values for each identifier:

- "Arinc Label": the ARINC 429 label value, that identifies the data type and the parameters associated with it. The usual data format is octal.
- "Symbolic label": symbolic name of the label, specifying its function.

**Example: ARINC 429 PTT file**

```
# ------------------------------------------------------------------------
@FILE_VERSION = 1.0
@PROTOCOL_NAME = arinc429
# ------------------------------------------------------------------------
# Labels for ARINC 429 protocol
#   Column order: Arinc Label, Symbolic Label
# ------------------------------------------------------------------------
# ----Definition----
001o, Distance to Go
002o, Time to Go
010o, Present Position - Latitude
011o, Present Position - Longitude
014o, Magnetic Heading
015o, Wind Speed
075o, Gross Weight
125o, Universal Time Coordinated
# ------------------------------------------------------------------------
s
```

### 13.8.2.3 Configuring ARINC 429 signals

For configuration assign the line to the input channel, set the threshold and the timing conditions.

For details on configuration settings, see Chapter 13.8.2.1, "ARINC 429 configuration settings", on page 617.

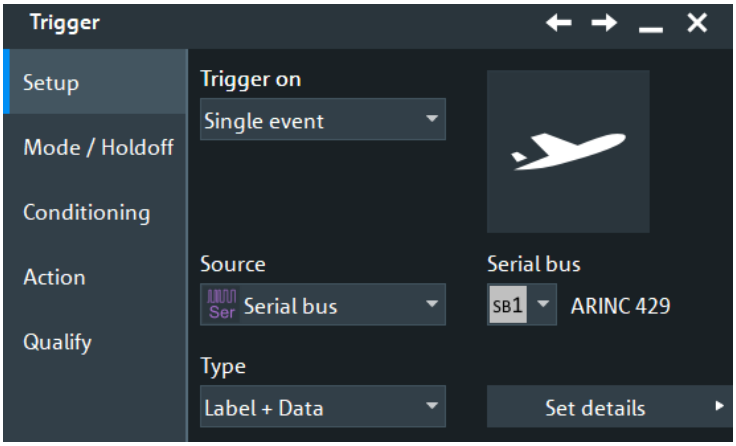
1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, select the protocol: "ARINC 429".
4. Tap the "Polarity" button, and select the waveform of the data line.
5. Set the logical thresholds.
6. Tap the "Bit Rate" button and set it for high or low speed.
7. If necessary, tap the "Min Gap" button to select it and set the minimum gap time.
8. If necessary, tap the "Max Response" button to select it and set the maximum response time.

9. Enable "Decode".

## 13.8.3 ARINC 429 trigger

### 13.8.3.1 ARINC 429 trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"




Make sure that:

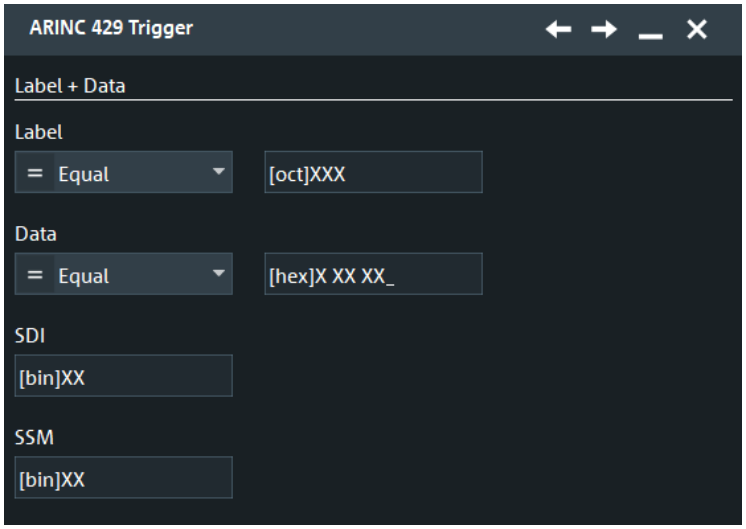
- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Trigger Type**
Selects the trigger type for ARINC 429 analysis.

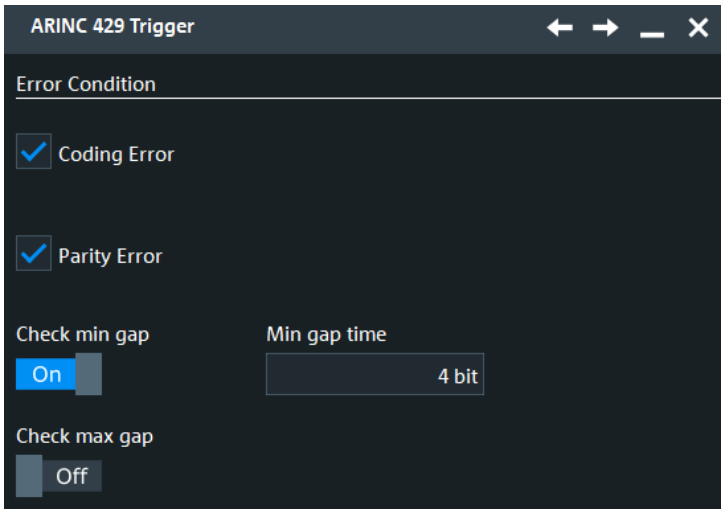
"Word Start" Sets the trigger to the start of the word.

"Word Stop" Sets the trigger to the stop of the word.

"Label + Data" Sets the trigger on a defined word format. You can define the label, the data and the SDI / SSM bits separately, see "Label + Data" on page 622.




"Error Condition"
Identifies various errors in the word, see "Error Conditions" on page 623.




Remote command:
`TRIGger<m>:ARINc:TYPE` on page 1814

**Label + Data**
Sets the trigger on a defined word format. You can define the label, the data and the SDI / SSM bits separately.

**Label setup: Condition, Label Min, Label Max ← Label + Data**
The label setup consists of the condition and one or two label patterns.

"Condition" Defines the operator to set a specific label ("Equal" or "Not equal") or a label range.

"Label Min" Defines the bit pattern of the label.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Label Max" The second label pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:ARINc:LABel:CONDition` on page 1816
`TRIGger<m>:ARINc:LABel:MIN` on page 1816
`TRIGger<m>:ARINc:LABel:MAX` on page 1816

**Data setup: Condition, Data Min, Data Max ← Label + Data**
The data setup consists of the condition and one or two data patterns.

"Condition" Defines the operator to set a specific data ("Equal" or "Not equal") or a data range.

"Data Min" Defines the bit pattern of the data.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:ARINc:DATA:CONDition` on page 1814
`TRIGger<m>:ARINc:DATA:MIN` on page 1815
`TRIGger<m>:ARINc:DATA:MAX` on page 1815

**SDI / SSM ← Label + Data**
Sets the values for the source/destination identifier (SDI) and the sign/status matrix (SSM) bits.

Remote command:
`TRIGger<m>:ARINc:SDI` on page 1817
`TRIGger<m>:ARINc:SSM` on page 1817

**Error Conditions**
Specifies the error conditions to be triggered on.

**Coding error ← Error Conditions**
Triggers on a coding error.

Remote command:
`TRIGger<m>:ARINc:ERRor:CODing` on page 1815

**Parity Error ← Error Conditions**
Checks the parity and triggers if the parity is even.

Remote command:
`TRIGger<m>:ARINc:ERRor:PARity` on page 1815

**Timing: Min gap, Max gap ← Error Conditions**
Defines the idle time between two words, which is needed for word synchronization. The beginning of the first bit after the gap marks the start of a new word.

You can define a minimum idle time "Min gap", and/or a maximum time "Max gap". The standard defines a minimum of 4-bit times to separate two subsequent words.

Timing settings are relevant for protocol configuration and error trigger.

If "Min gap" and/or "Max gap" are enabled, the instrument detects the specified gaps during decoding. If the trigger type "Error condition" is selected in addition, the instrument triggers when the gap is shorter or longer than the specified gaps, respectively.

Remote command:
`BUS<m>:ARINc:MAXGap:BITS` on page 1812
`BUS<m>:ARINc:MAXGap:SELect` on page 1811
`BUS<m>:ARINc:MINGap:BITS` on page 1812
`BUS<m>:ARINc:MINGap:SELect` on page 1812
`TRIGger<m>:ARINc:MINGap:BITS` on page 1817
`TRIGger<m>:ARINc:MINGap:SELect` on page 1816
`TRIGger<m>:ARINc:MAXGap:BITS` on page 1817
`TRIGger<m>:ARINc:MAXGap:SELect` on page 1817

#### 13.8.3.2 Triggering on ARINC 429

Prerequisites: A bus is configured for the ARINC 429 signal to be analyzed.

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Select the serial bus that is set to ARINC 429.
5. Select the "Trigger type".
6. For more complex trigger types, enter the data pattern conditions.
   For details, see Chapter 13.8.3.1, "ARINC 429 trigger settings", on page 621.

### 13.8.4 ARINC 429 label list

Label lists are protocol-specific. An ARINC 429 label file contains two values for each identifier:

- "Arinc Label": the ARINC 429 label value, that identifies the data type and the parameters associated with it. The usual data format is octal.
- "Symbolic label": symbolic name of the label, specifying its function.

**Example: ARINC 429 PTT file**

```
# ------------------------------------------------------------------------
@FILE_VERSION = 1.0
@PROTOCOL_NAME = arinc429
```

```
# ------------------------------------------------------------------------
# Labels for ARINC 429 protocol
#   Column order: Arinc Label, Symbolic Label
# ------------------------------------------------------------------------
# ----Definition----
001o, Distance to Go
002o, Time to Go
010o, Present Position - Latitude
011o, Present Position - Longitude
014o, Magnetic Heading
015o, Wind Speed
075o, Gross Weight
125o, Universal Time Coordinated
# ------------------------------------------------------------------------
s
```

See also: Chapter 13.1.4, "Content and format of the PTT file", on page 520.

Remote command:

- `BUS<m>:ARINc:WORD<n>:SYMBol?` on page 1821

## 13.8.5 ARINC 429 decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".
2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode Results Table

The "Decode results" table contains information about all decoded frames.

ARINC 429 Decode Results SB1

| | State | Start | Stop | Label [oct] | SDI [bin] | Data [hex] | SSM [bin] | Label name | Data (32 bit) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Ok | -1.69 ms | -1.37 ms | 101 | 10 | 2AAAA | 11 | Bus | [hex] EAAA AA82 |
| 2 | Coding Error | -1.32 ms | -1.22 ms | 000 | 00 | 00 | 00 | | [hex] 0000 0000 |
| 3 | Ok | -950 µs | -630 µs | 101 | 01 | 54455 | 11 | Bus | [hex] 7511 5582 |
| 4 | Ok | -580 µs | -260 µs | 101 | 01 | C955 | 11 | Bus | [hex] 6325 5582 |
| 5 | Ok | -210 µs | 110 µs | 101 | 10 | 22222 | 10 | Bus | [hex] 4888 8A82 |
| 6 | Ok | 160 µs | 480 µs | 102 | 00 | 24243 | 00 | Left Wing | [hex] 8909 0C42 |
| 7 | Ok | 530 µs | 850 µs | 103 | 10 | 26263 | 10 | Right Wing | [hex] 4989 8EC2 |
| 8 | Parity Error | 900 µs | 1.22 ms | 104 | 00 | 24683 | 10 | | [hex] 491A 0C22 |

*Table 13-11: Content of the Decode results table*

| Column | Description |
|---|---|
| State | Overall state of the frame. |
| Start | Time of word start in relation to the trigger point |
| Stop | Time of word stop in relation to the trigger point |
| Label | The value of the label bytes |
| SDI | The state of the SDI bits |
| DATA | All 32 bits of the word. |
| SSM | The state of the SSM bits |
| Label Name | The label name |
| Data (32 bit) | The value of the data bytes |

**Tools**

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

**Remote commands**

Remote commands are described in Chapter 24.17.9.3, "Decode results", on page 1818.

## 13.8.6 Search on decoded ARINC 429 data

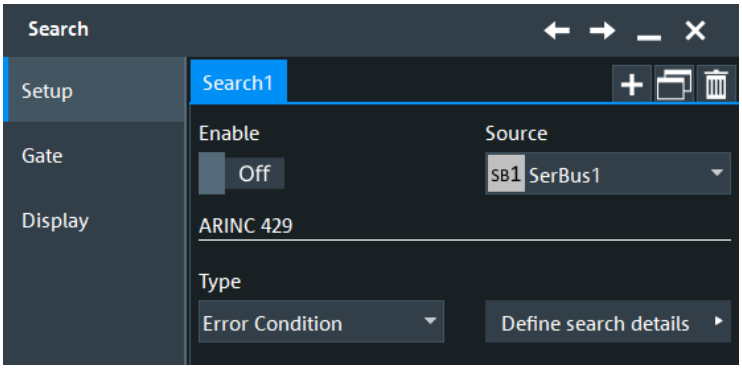
Using the search functionality, you can find various events in the decoded data, the same events which you also can trigger on. Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search "Source" to the serial bus that is configured for the protocol to be analyzed.

See also Chapter 11, "Search functions", on page 449.

### 13.8.6.1 ARINC 429 search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"




**Type**
The search criterion is defined by "Type". All trigger types are also available for search.

Additional search parameters are provided under "Detailed Search Parameter Setup".

"Word Start" Searches for the start word.

"Word Stop" Searches for the stop word.

"Label + Data" Searches for a defined word format. You can search for the label, the data, the SDI, and SSM bits separately.
For details, see "Label + Data" on page 622.

"Error condition"
Identifies various errors in the frame, see "Error Conditions" on page 623.

Remote command:
`SEARch:TRIGger:ARINc:TYPE` on page 1821

**Label setup: Condition, Label min, Label max**
The label setup consists of the condition and one or two label patterns.

The label setup settings are the same as in the ARINC trigger setup, see "Label setup: Condition, Label Min, Label Max" on page 622.

Remote command:
`SEARch:TRIGger:ARINc:LABel:CONDition` on page 1822
`SEARch:TRIGger:ARINc:LABel:MIN` on page 1822
`SEARch:TRIGger:ARINc:LABel:MAX` on page 1823

**Data setup: Condition, Data min, Data max**
The data setup consists of the condition and one or two data patterns.

The data setup settings are the same as in the ARINC trigger setup, see "Data setup: Condition, Data Min, Data Max" on page 623.

Remote command:
`SEARch:TRIGger:ARINc:DATA:CONDition` on page 1822
`SEARch:TRIGger:ARINc:DATA:MIN` on page 1822
`SEARch:TRIGger:ARINc:DATA:MAX` on page 1823

**SDI / SSM setup: SDI, SSM**
The SDI / SSM setup consists of the SDI and SSM.

The SDI / SSM setup settings are the same as in the ARINC trigger setup, see "SDI / SSM" on page 623.

Remote command:
SEARch:TRIGger:ARINc:SDI on page 1823
SEARch:TRIGger:ARINc:SSM on page 1823

**Error Condition**
Selects the error type to be searched for. You can select one or more error types as search condition.

The error types are the same as in the ARINC trigger setup, see "Error Conditions" on page 623

Remote command:
SEARch:TRIGger:ARINc:ERRor:CODing on page 1823
SEARch:TRIGger:ARINc:ERRor:PARity on page 1824
SEARch:TRIGger:ARINc:ERRor:TIMing on page 1824

### 13.8.6.2 ARINC search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands:

- SEARch:RESult:ARINc:WCOunt? on page 1826
- SEARch:RESult:ARINc:WORD<m>:DATA? on page 1825
- SEARch:RESult:ARINc:WORD<m>:LABel? on page 1825
- SEARch:RESult:ARINc:WORD<m>:PATTern? on page 1825
- SEARch:RESult:ARINc:WORD<m>:SDI? on page 1826
- SEARch:RESult:ARINc:WORD<m>:SSM? on page 1825
- SEARch:RESult:ARINc:WORD<m>:STARt? on page 1827
- SEARch:RESult:ARINc:WORD<m>:STATe? on page 1827
- SEARch:RESult:ARINc:WORD<m>:STOP? on page 1826
- SEARch:RESult:ARINc:WORD<m>:SYMBol? on page 1826

# 13.9 Ethernet 10BASE-T and 100BASE-TX (option R&S RTP-K8)

Twisted-pair Ethernet technologies are based on the family of standards IEEE 802.3, issued by the Institute of Electrical and Electronics Engineers (IEEE).

R&S RTP-K8 is a firmware option that enables the R&S RTP to analyze Ethernet protocol variants 10BASE-T and 100BASE-TX, by decoding the signal and searching within the decoded events. It is possible to trigger on 10BASE-T signals and 100BASE-TX signals. The option is compatible with the standards IEEE 802.3i of 1990 (10BASE-T) and IEEE 802.3u of 1995 (100BASE-TX).

- The Ethernet protocol ..... 629
- Ethernet configuration ..... 629
- Ethernet trigger ..... 635
- Ethernet decode results ..... 639
- Search on decoded Ethernet data ..... 641

## 13.9.1 The Ethernet protocol

The two Ethernet protocol variants that R&S RTP-K8 can process have the following features:

- 10BASE-T uses Manchester coding (or phase encoding, PE). In terms of a logical Boolean operation, the Manchester value of each bit is the exclusive disjunction (XOR) of the original data value and the clock value. A "0" is expressed by a high-to-low transition, a "1" by a low-to-high transition. These transitions, which occur at the middle of each bit period, make the signal self-clocked.
- 100BASE-TX uses a 4B5B multi-level transmit (MLT-3) encoding. The protocol sequentially cycles through a sequence of the voltage levels -1 V, 0 V, +1 V, and 0 V. To transmit a "1" bit, MLT-3 moves to the next state; to transmit a "0" bit, it stays in the same state. 4B5B block coding is used to map groups of four bits onto groups of five bits. Also, the signal is scrambled.

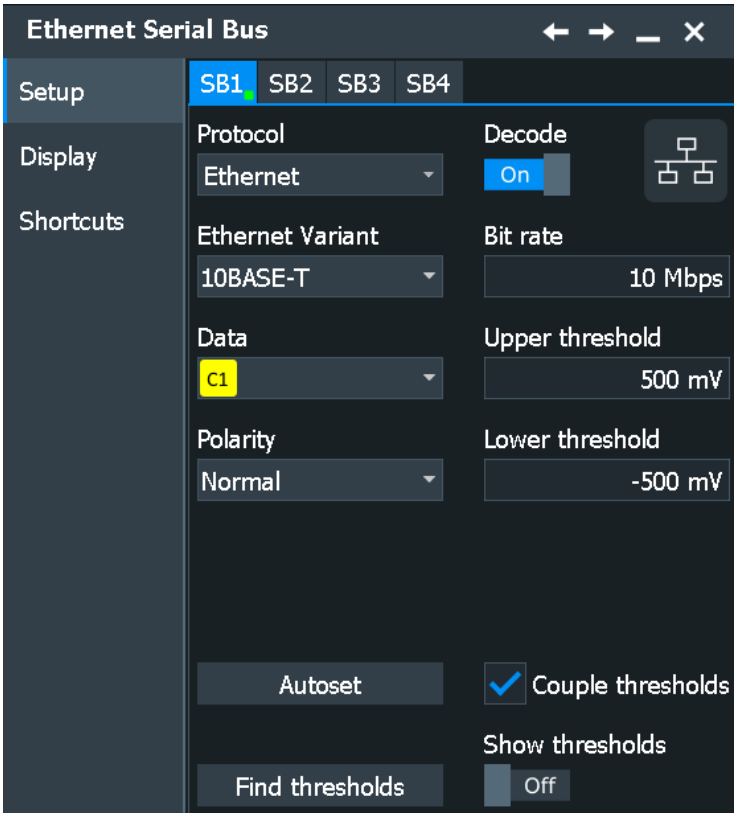
All Ethernet-over-twisted-pair technologies use wires with four twisted pairs of cables (and 8P8C connectors), but 10BASE-T and 100BASE-TX only require two pairs of wires.

## 13.9.2 Ethernet configuration

### 13.9.2.1 Ethernet configuration settings

Access: [Bus] > "Ethernet" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Protocol**
Defines the protocol type of the selected serial bus.

Remote command:
`BUS<m>:TYPE` on page 1637

**Decode**
Enables the decoding of the selected bus. The signal icon of the bus appears on the signal bar.

Remote command:
`BUS<m>[:STATe]` on page 1637

**Ethernet Variant**
Defines the Ethernet protocol variant and transmission speed.

"10BASE-T"
Selects the Ethernet protocol variant 10BASE-T (standard data rate 10 Mbit/s).

"100BASE-TX"
Selects the Ethernet protocol variant 100BASE-TX, which provides 100 Mbit/s use data rate. Due to 4b/5b encoding, the raw data rate on the line is 125 Mbit/s. This value is used by R&S RTP-K8 as the bit rate default for 100BASE-TX.

Remote command:
BUS<m>:ETHernet:VARiant on page 1828

**Source**
Defines the source settings for the data signal.

Permitted source selections are the analog, mathematical, and reference channels.

As soon as the serial bus trigger has been selected, the only permitted source selections are the analog channels "C1" – "C4", which are required for triggering.

Remote command:
BUS<m>:ETHernet:SOURce on page 1828

**Polarity**
Defines the polarity ("Normal" or "Inverted") of the data signal. This setting is only available in 10BASE-T.

Remote command:
BUS<m>:ETHernet:POLarity on page 1828

**Thresholds**
Sets the threshold value for the digitization of each signal line. If the signal value on the line is higher than the upper threshold, the signal state is high. Otherwise, if the signal value is below the lower threshold, the signal state is considered low.

| | |
|---|---|
| "Thresholds" | Enter the values directly: upper threshold in the upper field, lower threshold in the lower field. |
| "Autoset" | Starts software algorithms for determining the signal threshold levels and bitrate. |
| "Find thresholds" | Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude. |
| "Couple thresholds" | Couples threshold settings between upper and lower threshold. |

Remote command:
BUS<m>:ETHernet:THReshold:HIGH on page 1829
BUS<m>:ETHernet:THReshold:LOW on page 1829
BUS<m>:ETHernet:PRESet on page 1829
BUS<m>:SETReflevels on page 1638
BUS<m>:FAUToset on page 1638

**Bit rate**
Defines the transmission speed setting for the data signal:

- 10BASE-T: default bit rate 10 Mbps
- 100BASE-TX: default bit rate 125 Mbps

In both variants, the permitted bit rates range from 10 kbps to 150 Mbps. Switching the variant adjusts the bit rate, independent of the previous setting.

Remote command:
BUS<m>:ETHernet:BITRate on page 1830

**Show thresholds**
If enabled, the threshold levels are displayed in the diagram.

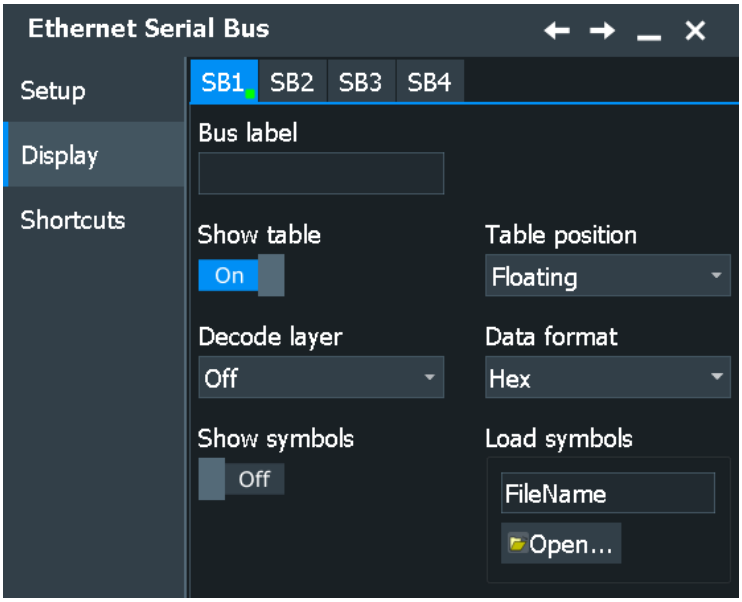
Remote command:
`BUS<m>:THReshold` on page 1638

### 13.9.2.2 Ethernet display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

To enhance the decode possibilities of the Ethernet protocol, you can use an additional setting in the "Display" tab: "Decode layer".




**Ethernet Specific Label Settings**

Label lists are protocol-specific. An Ethernet label file contains two values for each identifier:

- "Ethernet Header Display": the Ethernet header display value (e.g. `FF:FF:FF:FF:FF:FF/48`)
- "Symbolic label": user-defined symbolic name of the header, specifying its function (e.g. `BroadCast`).

For example, an ethernet label list can be a vendor-specific MAC address list:

```
# ----------------------------------------------------------------------------
@FILE_VERSION  = 1.00
@PROTOCOL_NAME = ethernet
# ----------------------------------------------------------------------------
# Labels for Ethernet protocol
#   Column order: Ethernet Header Display, Label
```

```
#-------------------------------------------------------------------------
# Supported MAC Address Format
# xx:xx:xx  standard 24 bit manufactory header
# xx:xx:xx:xx:xx:xx/yy  support other length headers
# yy should be the header length in decimal
# yy should be between 24 - 48
# -------------------------------------------------------------------------
00:00:0C, Cisco
00:01:13, Olympus
00:01:14, KandaTsu
00:04:07, TopconPo
00:0B:64, KiebackP
00:1B:C5:06:C0:00/36, LuxconSy
00:1B:C5:06:D0:00/36, TesElect
00:1B:C5:06:E0:00/36, TwoDimen
00:1B:C5:06:F0:00/36, LlcEmzio
00:1F:BE, Shenzhen
00:21:8F, Avantgar
00:21:90, GoliathS
00:21:91, D-Link
00:21:92, BaodingG
00:50:C2:5F:60:00/36, Cambridg
00:50:C2:5F:70:00/36, Metrolog
00:50:C2:5F:80:00/36, GrupoEpe
18:42:2F, AlcatelL
18:44:62, RiavaNet
18:46:17, SamsungE
18:8E:D5, TpVision
18:E7:F4, Apple
40:D8:55:1C:80:00/36, SensataT
40:D8:55:1C:90:00/36, Andy-L
40:D8:55:1C:A0:00/36, RigelEng
40:D8:55:1C:B0:00/36, MgSRL
40:D8:55:1C:D0:00/36, YxlonInt
40:D8:55:1C:E0:00/36, PeterHub
40:D8:55:1C:F0:00/36, OmnikNew
40:D8:55:1D:00:00/36, WebeasyB
FC:F8:B7, TronteqE
FC:FA:F7, Shanghai
FC:FB:FB, Cisco
FC:FE:77, HitachiR
FF:FF:FF:FF:FF:FF/48, BroadCast
```

In the figure below, you can see a decoded broadcast symbol. The symbol name is displayed in the "Destination symbol" column of the "Decode restuls" table.

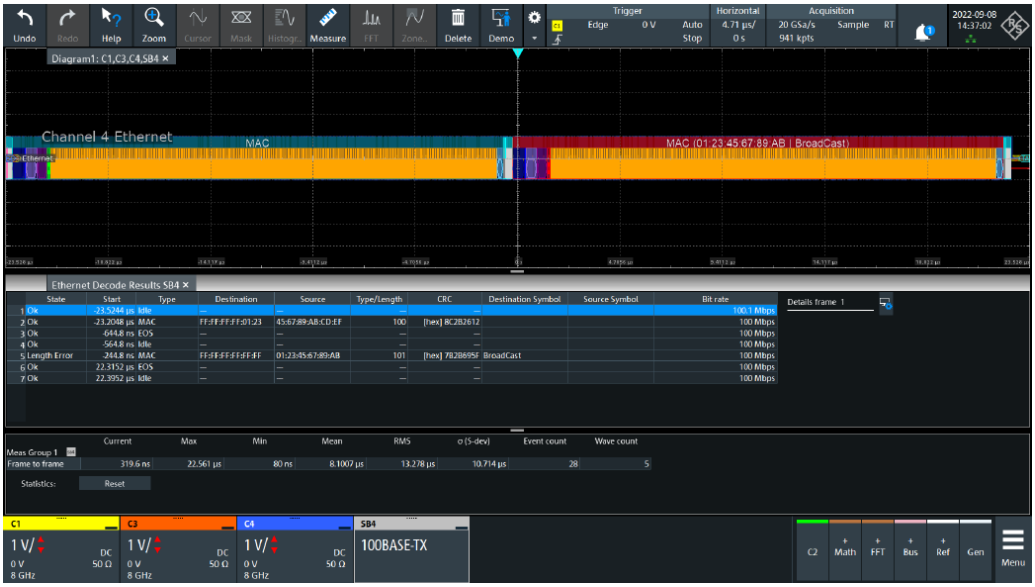



*Figure 13-14: Decoded Ethernet results with applied label*

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the end results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

Available are:

- "Edges": Shows where the decoder identified transitions through the threshold.
- "Binary": Shows the bits as identified by the decoder.
- "Descramble": Shows the bits after signal has been descrambled.
- "5b Symbol": Shows the symbol values after the 4b/5b decoder.

### 13.9.2.3 Configuring Ethernet signals

For configuration, assign the lines to the input channels and define the active states and the logical thresholds.

For details on configuration settings, see Chapter 13.9.2.1, "Ethernet configuration settings", on page 629.

1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, select the protocol: "Ethernet".
4. Tap "Ethernet Variant" and select the variant ("10BASE-T" or "100BASE-TX") you want to set up.

   **Note:** Note that no triggering on the serial bus is available.
   To trigger the signal, use the edge trigger on the source channel.

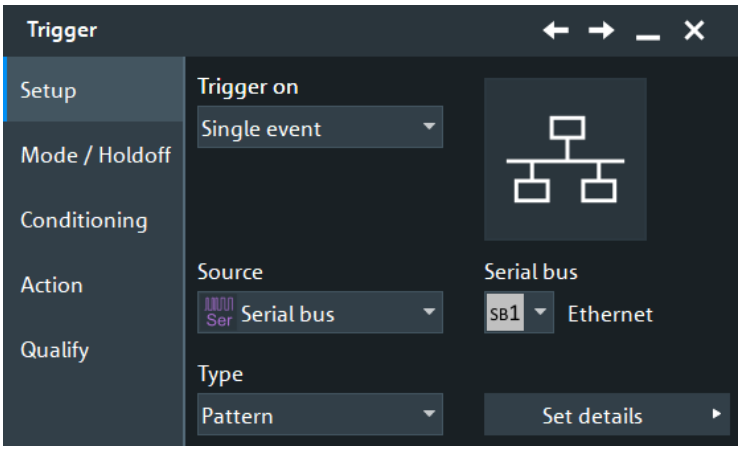
5. For the variant "10BASE-T", select the polarity ("Normal" or "Inverted") of the data signal.
6. Set the logical thresholds.
7. Enable "Decode".

## 13.9.3 Ethernet trigger

If you need information on how to get started with triggering on Ethernet 10BASE-T signals, see Chapter 13.9.3.2, "Triggering on Ethernet signals", on page 639. Otherwise proceed with the Ethernet trigger settings.

### 13.9.3.1 Ethernet trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"




Make sure that:

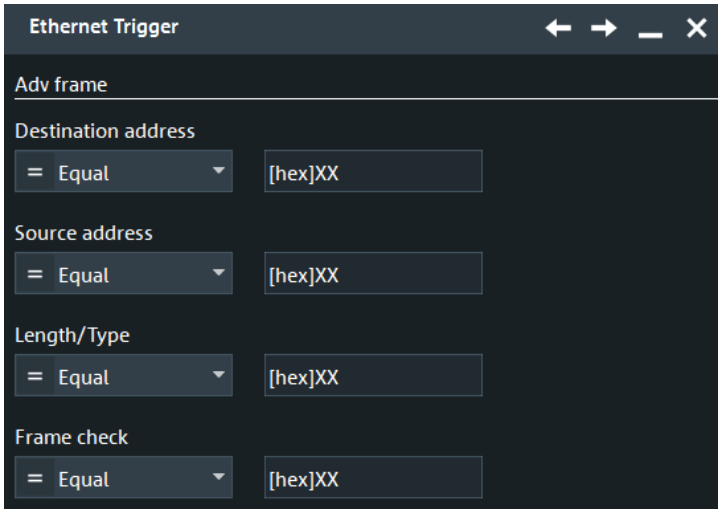
- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Type**
Selects the trigger type for the Ethernet analysis.

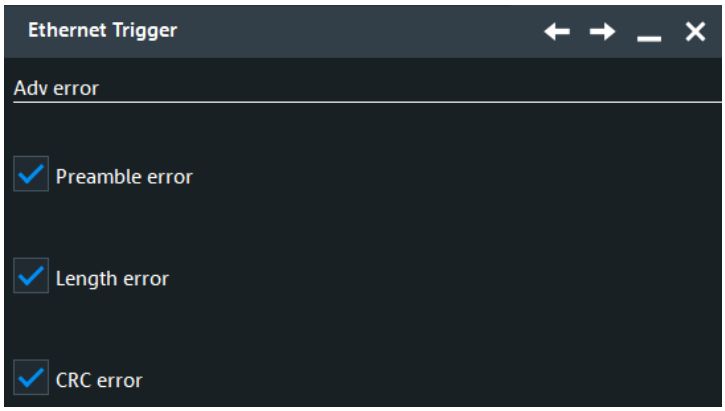
"Frame Start" Available only for "Variant > 10BASE-T".
Sets the trigger to the start frame delimiter (SFD). The start of frame (SOF) condition is the occurrence of the preamble; the trigger instant is after the SFD.

"Pattern" Available only for "Variant > 10BASE-T".
Sets the trigger to any bit pattern (data) that can be freely specified, starting from the beginning of the frame. The trigger instant is after the last bit of the specified data pattern.

"Adv frame " Triggers on an advanced frame.




"Adv error" Triggers on any selected errors.




Remote command:
`TRIGger<m>:ETHernet:TYPE` on page 1831

**Ethernet Variant**
Defines the Ethernet protocol variant and transmission speed.

"10BASE-T"
Selects the Ethernet protocol variant 10BASE-T (standard data rate 10 Mbit/s).

"100BASE-TX"
Selects the Ethernet protocol variant 100BASE-TX, which provides 100 Mbit/s use data rate. Due to 4b/5b encoding, the raw data rate on the line is 125 Mbit/s. This value is used by R&S RTP-K8 as the bit rate default for 100BASE-TX.

Remote command:
BUS<m>:ETHernet:VARiant on page 1828

**Data Pattern**
Specifies the data pattern that is to be triggered on.

Remote command:
TRIGger<m>:ETHernet:PATTern on page 1831

**Bit rate**
Defines the transmission speed setting for the data signal:

- 10BASE-T: default bit rate 10 Mbps
- 100BASE-TX: default bit rate 125 Mbps

In both variants, the permitted bit rates range from 10 kbps to 150 Mbps. Switching the variant adjusts the bit rate, independent of the previous setting.

Remote command:
BUS<m>:ETHernet:BITRate on page 1830

**Bit index**
Defines the position of the first bit of the data pattern.

| | |
|---|---|
| "Bit index operator" | Sets the operator ("Equal", "Greater or equal", or "In range"). |
| "Bit index" | Sets the bit index (data position), or the start value of a bit index range. |
| "Bit index to" | Sets end value of a bit index range (data position range). Available only, if the "Bit index operator" is set to "In range". |

**Destination Address**
Defines the destination address.

| | |
|---|---|
| "Condition" | Sets the comparison condition to a specific value or a range. |
| "Destinataion Address (Min)" | Specifies the value or sets the start value of a range. |
| "Destinataion Adress (Max)" | Sets the maximum value of a range for "Condition" = "In range"/"Out of range". |

Remote command:
TRIGger<m>:ETHernet:FRAMe:DCONdition on page 1833
TRIGger<m>:ETHernet:FRAMe:DMAX on page 1834
TRIGger<m>:ETHernet:FRAMe:DMIN on page 1834

**Source Address**
Defines the source address, the physical address of the device that sends the frame.

| | |
|---|---|
| "Condition" | Sets the comparison condition to a specific value or a range. |
| " Source Address (Min)" | Specifies the value or sets the start value of a range. |
| "Source Adress (Max)" | Sets the maximum value of a range for "Condition" = "In range"/"Out of range". |

Remote command:
TRIGger<m>:ETHernet:FRAMe:SCONdition on page 1834
TRIGger<m>:ETHernet:FRAMe:SMAX on page 1835
TRIGger<m>:ETHernet:FRAMe:SMIN on page 1835

**Length/ Type**
Defines the length/type field value.

"Condition" Sets the operator to trigger on a specific address pattern ("Equal" or "Not equal") or an address range.

"Length/Type (Min)" Defines the length/type pattern for all operators that require one pattern.

"Length/Typ (Max)" Defines the length/type pattern that is required to specify a range with conditions "In range" and "Out of range".

Remote command:
TRIGger<m>:ETHernet:FRAMe:TCONdition on page 1835
TRIGger<m>:ETHernet:FRAMe:TMAX on page 1836
TRIGger<m>:ETHernet:FRAMe:TMIN on page 1836

**Frame check**
Defines the frame check sequence value.

"Condition" Sets the comparison condition to a specific value or a range.

"Frame check (Min)" Specifies the value or sets the start value of a range.

"Frame check (Max)" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
TRIGger<m>:ETHernet:FRAMe:CCONdition on page 1832
TRIGger<m>:ETHernet:FRAMe:CMAX on page 1833
TRIGger<m>:ETHernet:FRAMe:CMIN on page 1833

**Error**
Triggers on enabled errors.

"Preamble error" Triggers on a frame with invalid preamble.

"Length error" Triggers on an incorrect length of the sequence - when additional or missing bits are detected and the sequence of bits is not as expected.

"CRC error" Triggers on a frame that has a mismatch of the cyclic redundancy check (CRC) value between the transmitting and receiving device.

Remote command:
TRIGger<m>:ETHernet:ERRor:CRC on page 1832
TRIGger<m>:ETHernet:ERRor:LENGth on page 1832
TRIGger<m>:ETHernet:ERRor:PREamble on page 1832

#### 13.9.3.2 Triggering on Ethernet signals

Prerequisite: A bus is configured for the Ethernet signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Select the serial bus that is set to Ethernet.
5. Tap "Type" and select the trigger type to be used for Ethernet protocol analysis. Available trigger types are "Frame Start", "Pattern", "Adv frame" and "Adv error".
6. Depending on the selected Ethernet variant, more setup conditions have to be specified.

For information on how to proceed with the configuration settings, see Chapter 13.9.3.1, "Ethernet trigger settings", on page 635.

### 13.9.4 Ethernet decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".
2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

**Decode Results Table**

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

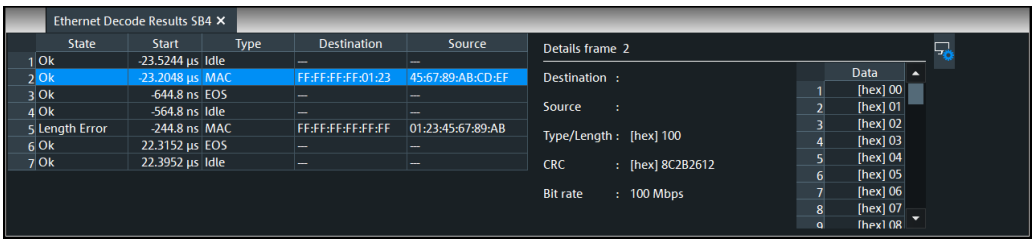



***Table 13-12: Content of the Decode results table***

| Column | Description |
|---|---|
| State | Overall state of the frame: either OK or the relevant error condition (preamble, length) |
| Start | Start time of the frame |
| Type | Frame type |
| Destination | Destination address of the frame |
| Source | Source address of the frame |
| Type/Length | The subprotocol (e.g. HTML, video, etc.) determines what meaning this field has. Since the content of this data area is not decoded, the interpretation of this field is ambivalent. It could either be the word type (specific for the subprotocol) or the word length. |
| Data | Values of the data bytes in a frame. The table shows a truncated version; to see all the bytes in a separate data table, activate "Show details". The data format is always hexadecimal. |
| CRC | FrameCheck (Cyclic Redundancy Code, CRC) |
| Destination Symbol | Translation (or symbolic label) of the destination address, if the label list is enabled. |
| Source Symbol | Translation (or symbolic label) of the source address, if the label list is enabled. |
| Number of Words | Number of words in the frame |
| Bit rate | Value of the bit rate |

### Tools

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

### Remote commands

Remote commands are described in Chapter 24.17.10.3, "Decode results", on page 1836.

### 13.9.5 Search on decoded Ethernet data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

Before you can start the search, you have to configure the bus correctly and acquire decoded data.

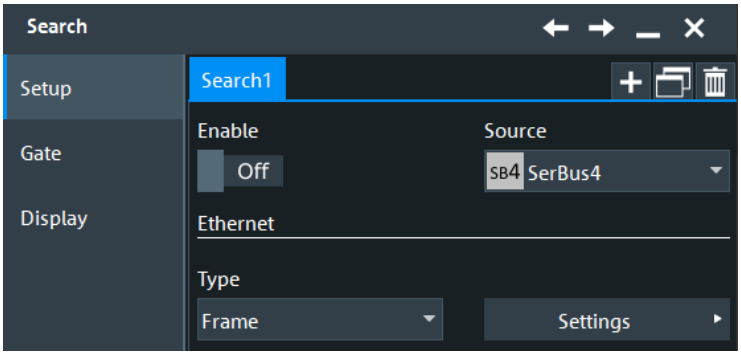
To search on decoded data, set the search source to "SerBus" for the configured protocol.

For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

If you need information on how to get started with searching Ethernet data, see Chapter 13.9.5.3, "Searching Ethernet data", on page 644. Otherwise proceed with the Ethernet search setup.

#### 13.9.5.1 Ethernet search setup

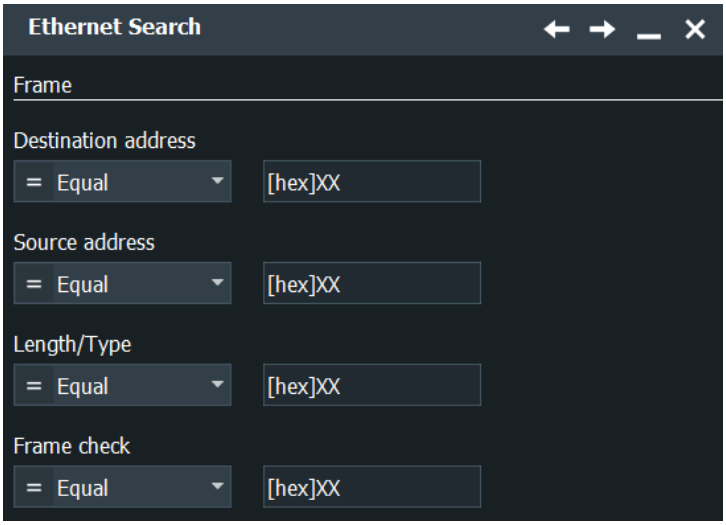
Access: [Bus] > "Shortcuts" tab > "Setup search"




**Type**
Define the event types to be searched. Available event types are "Frame" and "Error".

Remote command:
`SEARch:TRIGger:ETHernet:TYPE` on page 1842

**Frame**
Searches for the following frame conditions: "Destination address", "Source address", "Length/Type", or "Frame check".




**Destination address ← Frame**
To search for a destination address, an address pattern or optionally an address range have to be specified.

Remote command:
SEARch:TRIGger:ETHernet:FRAMe:DCONdition on page 1843
SEARch:TRIGger:ETHernet:FRAMe:DMIN on page 1843
SEARch:TRIGger:ETHernet:FRAMe:DMAX on page 1843

**Source address ← Frame**
To search for a source address, an address pattern or optionally an address range have to be specified.

Remote command:
SEARch:TRIGger:ETHernet:FRAMe:SCONdition on page 1844
SEARch:TRIGger:ETHernet:FRAMe:SMIN on page 1844
SEARch:TRIGger:ETHernet:FRAMe:SMAX on page 1844

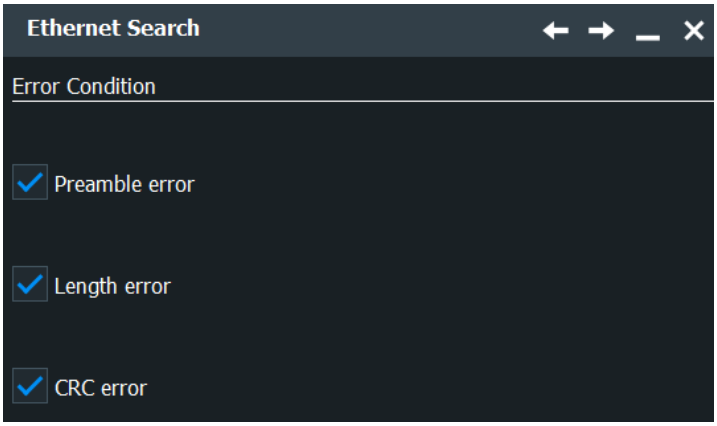
**Length/Type ← Frame**
To search for a frame length or frame type, a type/length pattern or optionally a range of type/length patterns have to be specified.

Remote command:
SEARch:TRIGger:ETHernet:FRAMe:TCONdition on page 1845
SEARch:TRIGger:ETHernet:FRAMe:TMIN on page 1845
SEARch:TRIGger:ETHernet:FRAMe:TMAX on page 1846

**Frame check ← Frame**
To search for a specific pattern, this pattern or optionally a range of patterns have to be specified.

Remote command:
SEARch:TRIGger:ETHernet:FRAMe:CCONdition on page 1846
SEARch:TRIGger:ETHernet:FRAMe:CMIN on page 1846
SEARch:TRIGger:ETHernet:FRAMe:CMAX on page 1847

**Error Condition**
Searches for the following error conditions: "Preamble Error" or "Length Error".




**Preamble Error ← Error Condition**
Searches for any preamble errors.

Remote command:
`SEARch:TRIGger:ETHernet:ERRor:PREamble` on page 1847

**Length Error ← Error Condition**
Searches for any length errors.

Remote command:
`SEARch:TRIGger:ETHernet:ERRor:LENGth` on page 1847

### 13.9.5.2 Ethernet search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands:

- `SEARch:RESult:ETHernet:WCOunt?` on page 1848
- `SEARch:RESult:ETHernet:WORD<m>:TYPE?` on page 1850
- `SEARch:RESult:ETHernet:WORD<m>:FTYPe?` on page 1850
- `SEARch:RESult:ETHernet:WORD<m>:STATe?` on page 1848
- `SEARch:RESult:ETHernet:WORD<m>:STARt?` on page 1849
- `SEARch:RESult:ETHernet:WORD<m>:STOP?` on page 1849
- `SEARch:RESult:ETHernet:WORD<m>:DESTaddress?` on page 1849
- `SEARch:RESult:ETHernet:WORD<m>:SRCaddress?` on page 1850

- SEARch:RESult:ETHernet:WORD<m>:DATA? on page 1850
- SEARch:RESult:ETHernet:WORD<m>:CRC? on page 1851
- SEARch:RESult:ETHernet:WORD<m>:DSYMbol? on page 1851
- SEARch:RESult:ETHernet:WORD<m>:SSYMbol? on page 1851
- SEARch:RESult:ETHernet:WORD<m>:BYTE<n>:VALue? on page 1852

#### 13.9.5.3 Searching Ethernet data

Prerequisite: A serial bus is configured for the Ethernet signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".
3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Tap "Source" and select the serial bus that is set to Ethernet (e.g. "SerBus1", unless already selected).

   The search dialog for Ethernet protocol analysis is opened.
5. Specify search criteria according to Chapter 13.9.5.1, "Ethernet search setup", on page 641.
6. To acquire a waveform, press [Single].

   The R&S RTP performs an Ethernet decode according to the thresholds and protocol settings of the associated serial bus source (here in our example SB1).
7. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog:

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

## 13.10 Ethernet 100BASE-T1 (option R&S RTP-57)

100BASE-T1 is an Ethernet based protocol specialized for the needs of the automotive industry, such as in the automotive networking applications, enabling advanced safety, comfort and infotainment features within the automobile.

Its specifications, with reference to IEEE Standard 802.3-2012, were developed by the OPEN (One-Pair Ether-Net) alliance under BroadR-Reach®.

BroadR-Reach is a Broadcom® point-to-point Ethernet physical layer (PHY) technology. Therefore, 100BASE-T1 is also known as BroadR-Reach Physical Layer (BR-PHY) or Open Alliance BroadR-Reach PHY (OABR PHY). For more details on the specifications, refer to http://www.opensig.org/about/specifications/.

This option is compatible with the MII (IEEE 802.3 Clause 22) and IEEE 802.3 MAC operating at 100Mbps.

- 100BASE-T1 basics....645
- 100BASE-T1 configuration....646
- 100BASE-T1 trigger....650
- 100BASE-T1 decode results....656
- Search on decoded 100BASE-T1 data....657

### 13.10.1 100BASE-T1 basics

The BR-PHY has the following objectives:

- Provides a PHY that supports full duplex operating at 100 Mbps over a pair of unshielded twisted pair (UTP) cable or better cable
- Provides compatibility with the Media Independent Interface (MII) IEEE 802.3 Clause 22 and IEEE 802.3 Media Access Controller (MAC) operating at 100 Mbps
- Achieves bit error rate (BER) of less than $1e^{-10}$

To achieve the above objectives, 100BASE-T1 uses 1000BASE-T PHYs with parts of IEEE 802.3 100BASE-TX in operation at 100 MBps and develops a new solution for the PHY sublayers, i.e. Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer as follows:

- Adopts full duplex communication of 1000BASE-T and therefore echo cancellation on a single twisted-pair channel. This feature reduces cabling while preserving the Ethernet MAC compatibility
- Adopts Pulse Amplitude Modulation (PAM-3) encoding scheme with the following encoding techniques:
  - Data encoding is carried out via a 4b3b encoder that converts the MII data (4B - four bits) with 25 MHz clock to three bits (3B) wide of data that is transmitted during one 33.3 MHz clock period
  - Symbol encoding is carried out via a one-dimensional (1D) PAM-3 encoder that converts the 3-bit groups into pairs of ternary symbols. These symbols are transmitted using three voltage levels (-1 V, 0 V and +1 V). One symbol is transmitted in each symbol period
  - Data scrambling is carried out via a sidestream scramble to randomize the sequence of transmitted symbols and avoid the presence of spectral lines in the signal spectrum

  PAM-3 has a higher spectral efficiency that limits the signaling bandwidth to 33.3 MHz instead of 65 MHz to 80 MHz in 1000BASE-T and 100BASE-TX such that communication occurs in the best part of a twisted-pair channel. This in turn improves return loss, reduces crosstalk and EMI, allows for a more aggressive Electromagnetic Compatibility (EMC) filtering and lower cost (often lower quality) cabling

In terms of the trigger and decoder development, multiple aspects of this protocol are new, specifically:

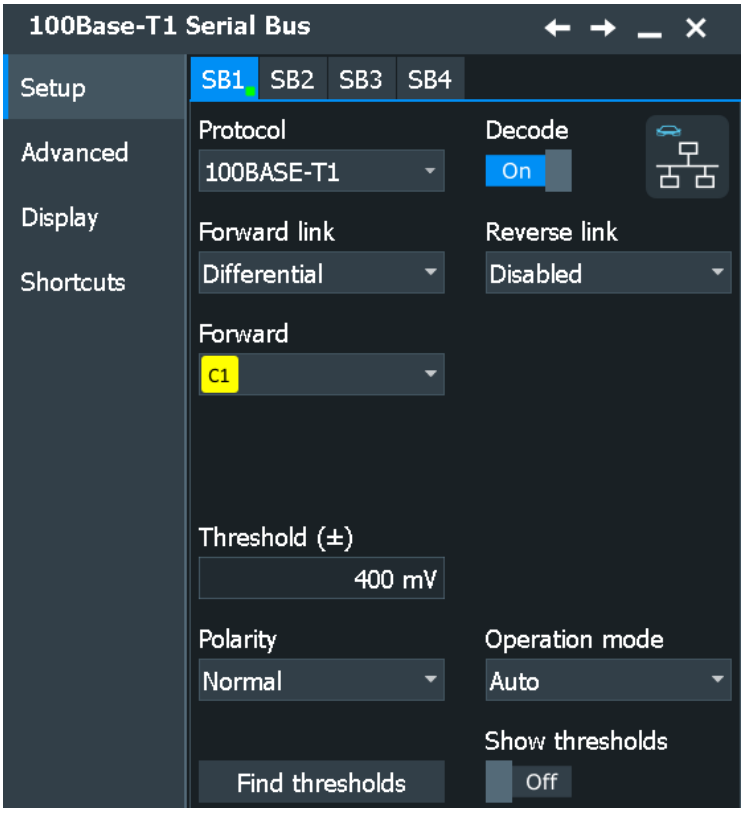
- Sidestream descrambler: Descrambles the randomized sequence of received ternary symbols
- 1D PAM-3 decoder: Decodes the received ternary symbols into groups of three bits according to a conversion chart
- 4b/3b line decoder: Decodes the groups of three bits (result of ternary pair) back to groups of four bits of the data stream

## 13.10.2 100BASE-T1 configuration

### 13.10.2.1 100BASE-T1 configuration settings

Access: [Bus] > "100BASE-T1" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Forward**
Defines the forward signal settings.

You can choose between a differential and single-ended signal and set the forward source channel for the waveform. Only analog channels are available for selection.

Remote command:
BUS<m>:HBTO:FTYP on page 1854
BUS<m>:HBTO:FDIF on page 1853
BUS<m>:HBTO:FDAP on page 1853
BUS<m>:HBTO:FDAM on page 1853

**Reverse**
Defines the reversed signal settings. The reverse direction is optional and can be disabled.

You can choose between a differential and single-ended signal and set the reverse source channel for the waveform. Only analog channels are available for selection.

Remote command:
BUS<m>:HBTO:RTYP on page 1855
BUS<m>:HBTO:RDIF on page 1854
BUS<m>:HBTO:RDAP on page 1854
BUS<m>:HBTO:RDAM on page 1854

**Reverse attenuation**
Sets the attenuation factor. It is used to de-amplify the reverse signal before subtracting it from the forward signal.

The effective signal amplitude passed to the decoder is:

$$Amp = (Fwd_+ - Fwd_-) - 10^{\frac{Attn}{20}}(Rev_+ - Rev_-)$$

Remote command:
BUS<m>:HBTO:ATTN on page 1853

**Polarity**
Selects the polarity of the data signal. You can select between "Normal" and "Inverted".

For "Normal" polarity, active high is defined as +1 V and active low is defined as -1 V.
For "Inverted" polarity, active high is defined as -1 V and active low is defined as +1 V.

Remote command:
BUS<m>:HBTO:POLarity on page 1855

**Thresholds**
Sets the positive and negative thresholds for the digitalization of the signal line.

The signal can have three states:

- High: the signal value is higher than the positive threshold.
- Low: the signal value is below the negative threshold.
- Zero: signal is between the positive and the negative threshold.

"Thresholds" Enter the value directly.

"Find thresholds" Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

Remote command:
BUS<m>:HBTO:THReshold on page 1855
BUS<m>:SETReflevels on page 1638
BUS<m>:FAUToset on page 1638

**Operation mode**
Selects the desired direction of the full-duplex signal for analysis. If set to "Auto", the decoder automatically detects the "Master" or "Slave" mode.

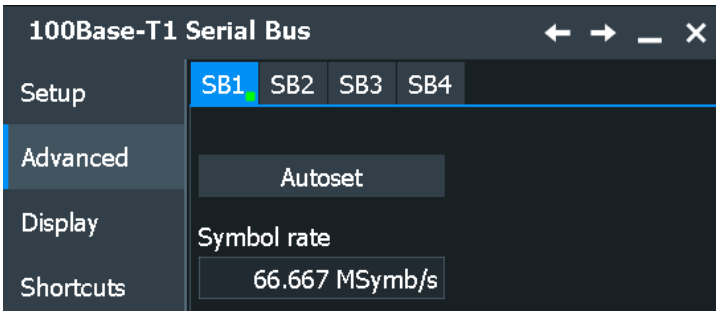
Remote command:
BUS<m>:HBTO:MODE on page 1855

**Show thresholds**
If enabled, the threshold levels are displayed in the diagram.

Remote command:
BUS<m>:THReshold on page 1638

### 13.10.2.2 Advanced settings

Access: [Bus] > "Advanced" tab.




**Symbol rate**
Defines the transmission rate of ternary symbols which is by default 66.67 MSymb/s. This parameter should be rarely changed.

Remote command:
BUS<m>:HBTO:SYMRate on page 1856

**Autoset**
Starts software algorithms for determining the signal threshold levels and bitrate.

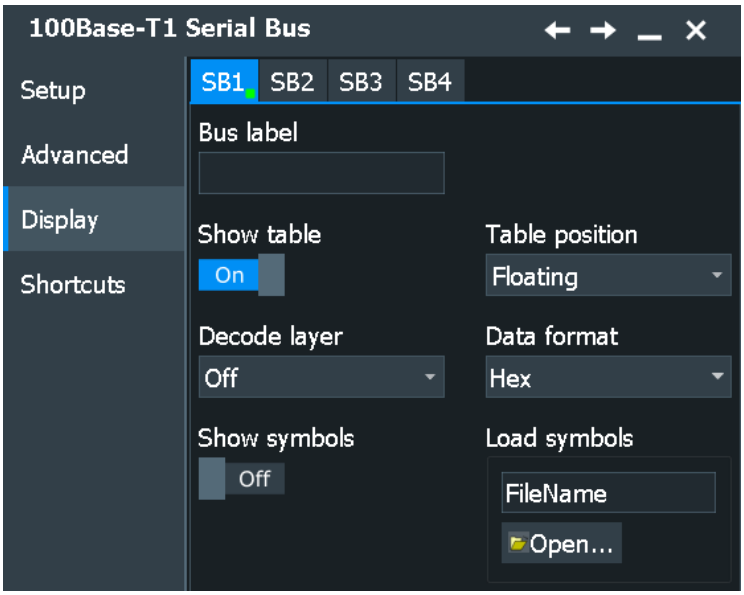
Remote command:
BUS<m>:FAUToset on page 1638

### 13.10.2.3 100BASE-T1 display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

To enhance the decode possibilities of the 100BASE-T1 protocol, you can use an additional setting in the "Display" tab: "Decode layer".




### 100BASE-T1 Specific Symbol Settings

Label lists are protocol-specific. Label lists for 100BASE-T1 are available in CSV and PTT format.

A 100BASE-T1 label file contains two values for each ethernet header:

- Ethernet Header Display
- Symbolic label: name of the address, specifiing its function in the bus network.

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the end results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

| | |
|---|---|
| "Off" | No decode layer is displayed. |
| "Ternary symbols" | Decoded ternary symbols. |
| "Scrambled bits" | Scrambled ternary bits. |
| "Descrambled bits" | Descrambled ternary bits. |
| "Reversed Bits" | Reserved bits |

#### 13.10.2.4 Configuring 100BASE-T1 signals

For configuration, assign the lines to the input channels and define the active states and the logical thresholds.

For details on configuration settings, see Chapter 13.10.2.1, "100BASE-T1 configuration settings", on page 646.

**Serial bus setup**

1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, select the protocol: "100BASE-T1".
4. Select the type and source for the reversed and forward signal.
5. Select the polarity ("Normal" or "Inverted") of the signal.
6. Set the logical thresholds.
7. Enable "Decode".

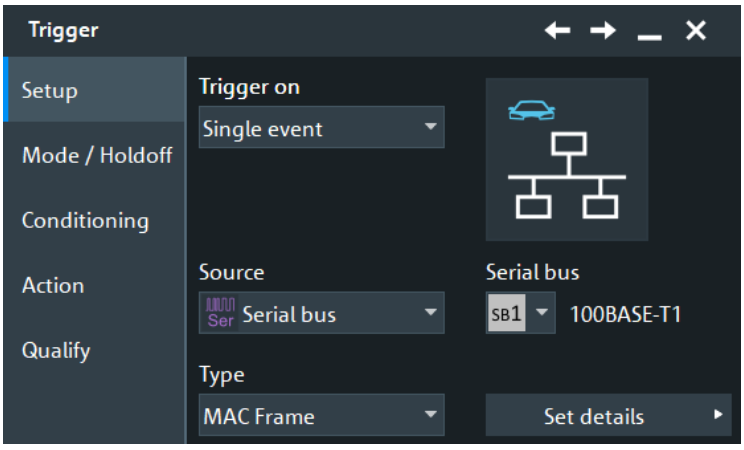
### 13.10.3 100BASE-T1 trigger

#### 13.10.3.1 100BASE-T1 trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"

In this section, all trigger settings are described. Their availability on the instrument depends on the selected trigger type. The user interface of the instrument displays only appropriate settings and guides you through the trigger setup.

For a list of supported trigger conditions, refer to data sheet.



Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Type**
Selects the trigger type for 100BASE-T1 analysis.

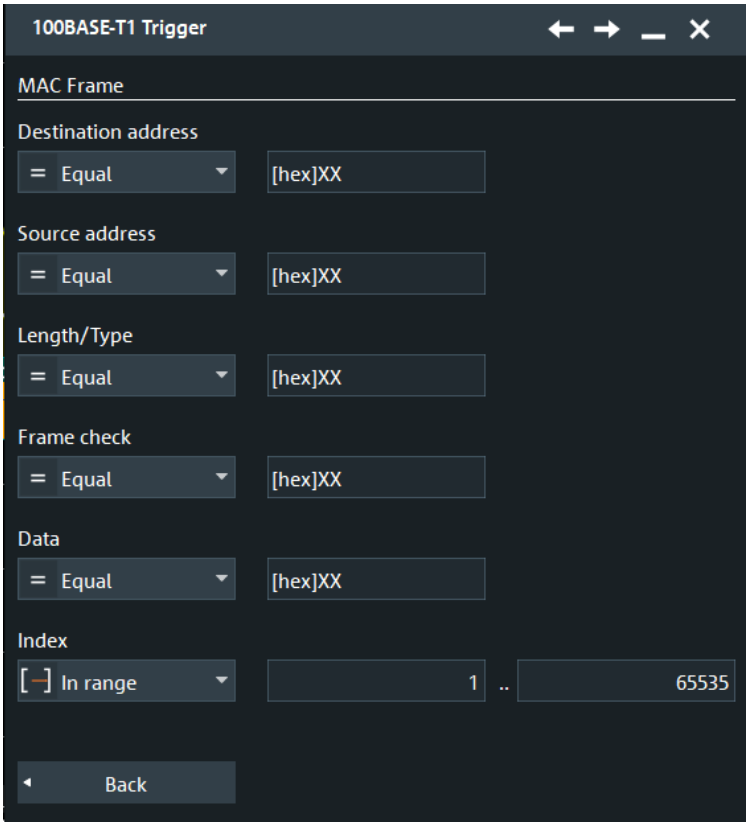
Remote command:
`TRIGger<m>:HBTO:TYPE` on page 1857

**Frame Start ← Type**
Triggers on the start of frame.

**MAC Frame ← Type**
Triggers on a Media Access Control (MAC) frame. You can specify:

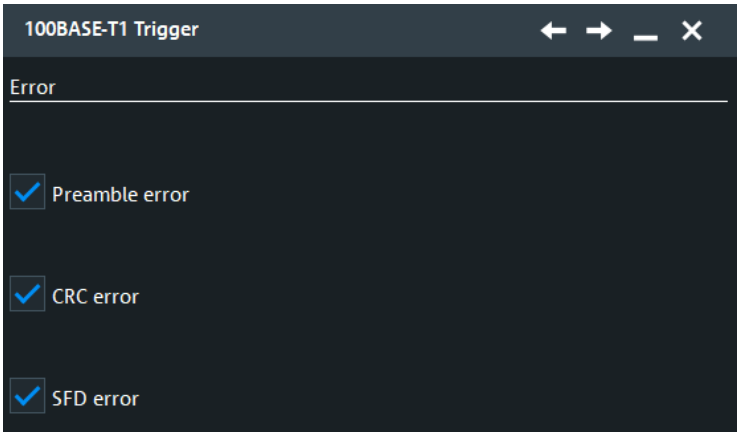
- Destination address
- Source address
- Length/Type
- Frame check
- Data
- Index

The frame contains addresses of the devices and MAC control information that define how to go about transmitting and receiving frames.




**IDLE ← Type**
Triggers on an idle frame. The frame is used for clock synchronization.

**Error ← Type**
Triggers on the specified error frame.




**Destination address**
Sets the specified destination address to be triggered on. The destination address setup consists of the condition and one or two data patterns.

The destination address corresponds to the address of the interface in the device that receives the frame. If the destination address does not match the interface's own Ethernet address, then the interface is free to ignore the rest of the frame.

"Condition" Defines the operator to set a specific destination address, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the destination address pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:HBTO:DADDress:CONDition` on page 1857
`TRIGger<m>:HBTO:DADDress:MIN` on page 1858
`TRIGger<m>:HBTO:DADDress:MAX` on page 1858

**Source address**
Sets the specified source address to be triggered on. The source address setup consists of the condition and one or two data patterns.

The source address is the physical address of the device that sends the frame.

"Condition" Defines the operator to set a specific source address, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the source address pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:HBTO:SADDress:CONDition` on page 1858
`TRIGger<m>:HBTO:SADDress:MIN` on page 1859
`TRIGger<m>:HBTO:SADDress:MAX` on page 1859

**Length/Type**
Sets the specified Length/Type to be triggered on. The Length/Type setup consists of the condition and one or two data patterns.

The value in this field indicates the manner in which the field is being used, either as a length or type field. As a length field, the value in the field indicates the number of logical link control (LLC) data octets that follow in the data field of the frame. As a type field, the value in the field is used to indicate the type of protocol data being carried in the data field of the frame.

"Condition" Defines the operator to set a specific Length/Type, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the Length/Type pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:HBTO:LENGth:CONDition` on page 1859
`TRIGger<m>:HBTO:LENGth:MIN` on page 1860
`TRIGger<m>:HBTO:LENGth:MAX` on page 1860

**Frame check**
Sets the specified frame check to be triggered on. The frame check setup consists of the condition and one or two data patterns.

This field contains a value that is used to check the integrity of the various bits in the frame fields (excluding the preamble / SFD).

"Condition" Defines the operator to set a specific frame check, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the frame check pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:HBTO:CRC:CONDition` on page 1860
`TRIGger<m>:HBTO:CRC:MIN` on page 1861
`TRIGger<m>:HBTO:CRC:MAX` on page 1861

**Data**
Sets the specified data to be triggered on. The data setup consists of the condition and one or two data patterns.

"Condition" Defines the operator to set a specific data, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the data pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
TRIGger<m>:HBTO:DATA:DCONdition on page 1861
TRIGger<m>:HBTO:DATA:DMIN on page 1862
TRIGger<m>:HBTO:DATA:DMAX on page 1862

**Index**
Sets the specified value or range within this series of data that is considered for the analysis. The index setup consists of the condition and one or two index values.

"Condition" Defines the operator to set a specific index, e.g. "Equal" or a range.

"Index Min/Index"
Defines the bit pattern of the index pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Index Max" The second index pattern is required to specify a range with conditions "In range".

Remote command:
TRIGger<m>:HBTO:DATA:ICONdition on page 1862
TRIGger<m>:HBTO:DATA:IMIN on page 1863
TRIGger<m>:HBTO:DATA:IMAX on page 1863

**Errors**
Sets the type of error events to be triggered on.

**Preamble Error ← Errors**
Triggers on a frame with invalid preamble.

Remote command:
TRIGger<m>:HBTO:ERRor:PREamble on page 1863

**CRC Error ← Errors**
Triggers on a frame that has a mismatch of the Cyclic Redundancy Check (CRC) value between the transmitting and receiving device.

Remote command:
TRIGger<m>:HBTO:ERRor:CRC on page 1863

**SFD Error ← Errors**
Triggers on a frame with invalid Start Frame Delimiter (SFD).

Remote command:
TRIGger<m>:HBTO:ERRor:SFD on page 1864

#### 13.10.3.2 Triggering on 100BASE-T1

Prerequisite: A bus is configured for the 100BASE-T1 signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.

2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Select the serial bus that is set to 100BASE-T1.
5. Tap "Type" and select the trigger type to be used for 100BASE-T1 protocol analysis.
6. To refine the trigger settings, configure additional settings, which are available for some trigger types.

For details, see Chapter 13.10.3, "100BASE-T1 trigger", on page 650.

### 13.10.4 100BASE-T1 decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".
2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

#### Decode Results Table

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

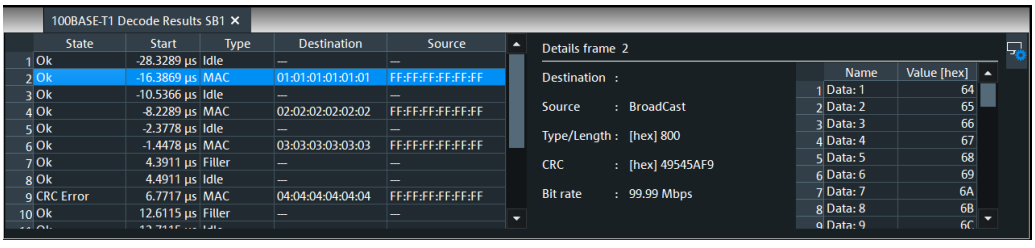


*Table 13-13: Content of the Decode results table*

| Column | Description |
|---|---|
| Frame | Frame count |
| State | Overall state of the frame: either OK or the relevant error condition (preamble, CRC or SFD) |
| Start | Start time of the frame |
| Type | Type of frame (e.g. Idle, MAC or data) |
| Destination | Destination address of the frame |
| Source | Source address of the frame |

*Table 13-14: Content of the Details frame table*

| Column | Description |
|---|---|
| Destination symbol | Symbol defined for the respective destination |
| Source symbol | Symbol defined for the respective source |
| Type/Length | The sub-protocol (e.g. HTML, video, etc.) determines what meaning this field has. Since the content of this data area is not decoded, the interpretation of this field is ambivalent. It could either be the word type (specific for the sub-protocol) or the word length. |
| CRC | Frame Check (Cyclic Redundancy Code, CRC) |
| Bit rate | Value of the bit rate |
| Name | State of the field |
| Value | Hexadecimal value of the field |

### Tools

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

### Remote commands

Remote commands are described in Chapter 24.17.11.3, "Decode results", on page 1864.

## 13.10.5 Search on decoded 100BASE-T1 data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

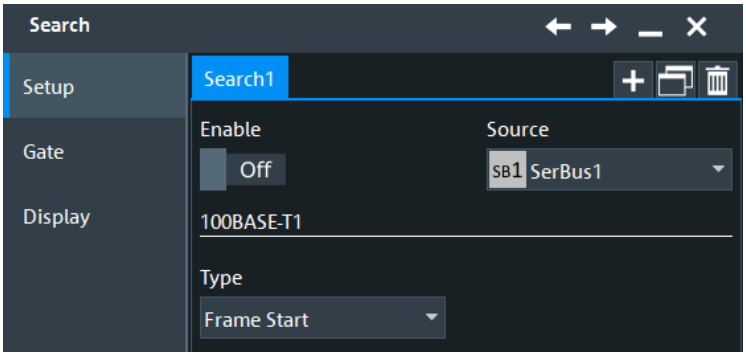
Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search source to "SerBus" for the configured protocol.

For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

#### 13.10.5.1 100BASE-T1 search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"




**Enable**
Enables the search for 100BASE-T1.

**Type**
Selects the search type for 100BASE-T1 analysis.

Remote command:
`SEARch:TRIGger:HBTO:TYPE` on page 1870

**Frame Start ← Type**
Searches for the start of frame.

**MAC Frame ← Type**
Searches for a Media Access Control (MAC) frame that matches the specified Destination address, Source address, Length/Type, Frame check, Data or Index condition.

The frame contains addresses of the devices and MAC control information that define how to go about transmitting and receiving frames.

**FILLER ← Type**
Searches for a Filler frame. The frame is used to maintain transmission activity.

**IDLE ← Type**
Searches for an idle frame. The frame is used for clock synchronization.

**Error ← Type**
Searches for the specified error frame.

**Destination address**
Sets the specified destination address to be searched for. The destination address setup consists of the condition and one or two data patterns.

The destination address corresponds to the address of the interface in the device that receives the frame. If the destination address does not match the interface's own Ethernet address, then the interface is free to ignore the rest of the frame.

"Condition" Defines the operator to set a specific destination address, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the destination address pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`SEARch:TRIGger:HBTO:DADDress:CONDition` on page 1871
`SEARch:TRIGger:HBTO:DADDress:MIN` on page 1871
`SEARch:TRIGger:HBTO:DADDress:MAX` on page 1871

**Source address**
Sets the specified source address to be searched for. The source address setup consists of the condition and one or two data patterns.

The source address is the physical address of the device that sends the frame.

"Condition" Defines the operator to set a specific source address, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the source address pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`SEARch:TRIGger:HBTO:SADDress:CONDition` on page 1872
`SEARch:TRIGger:HBTO:SADDress:MIN` on page 1872
`SEARch:TRIGger:HBTO:SADDress:MAX` on page 1873

**Length/Type**
Sets the specified Length/Type to be searched for. The Length/Type setup consists of the condition and one or two data patterns.

The value in this field indicates the manner in which the field is being used, either as a length or type field. As a length field, the value in the field indicates the number of logical link control (LLC) data octets that follow in the data field of the frame. As a type field, the value in the field is used to indicate the type of protocol data being carried in the data field of the frame.

"Condition" Defines the operator to set a specific Length/Type, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the Length/Type pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`SEARch:TRIGger:HBTO:LENGth:CONDition` on page 1873
`SEARch:TRIGger:HBTO:LENGth:MIN` on page 1873
`SEARch:TRIGger:HBTO:LENGth:MAX` on page 1874

**Frame check**
Sets the specified frame check to be searched for. The frame check setup consists of the condition and one or two data patterns.

This field contains a value that is used to check the integrity of the various bits in the frame fields (excluding the preamble / SFD).

"Condition" Defines the operator to set a specific frame check, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the frame check pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`SEARch:TRIGger:HBTO:CRC:CONDition` on page 1874
`SEARch:TRIGger:HBTO:CRC:MIN` on page 1874
`SEARch:TRIGger:HBTO:CRC:MAX` on page 1875

**Data**
Sets the specified data to be searched for. The data setup consists of the condition and one or two data patterns.

"Condition" Defines the operator to set a specific data, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the data pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`SEARch:TRIGger:HBTO:DATA:DCONdition` on page 1875
`SEARch:TRIGger:HBTO:DATA:DMIN` on page 1875
`SEARch:TRIGger:HBTO:DATA:DMAX` on page 1876

**Index**
Sets the specified value or range within this series of data that is considered for the search. The index setup consists of the condition and one or two index values.

"Condition" Defines the operator to set a specific index, e.g. "Equal" or a range.

"Index Min/Index"
Defines the bit pattern of the index pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Index Max" The second index pattern is required to specify a range with conditions "In range".

Remote command:
`SEARch:TRIGger:HBTO:DATA:ICONdition` on page 1876
`SEARch:TRIGger:HBTO:DATA:IMIN` on page 1877
`SEARch:TRIGger:HBTO:DATA:IMAX` on page 1877

**Errors**
Sets the type of error events to be searched for.

**Preamble Error ← Errors**
Searches for a frame with invalid preamble.

Remote command:
`SEARch:TRIGger:HBTO:ERRor:PREamble` on page 1877

**CRC Error ← Errors**
Searches for a frame that has a mismatch of the Cyclic Redundancy Check (CRC) value between the transmitting and receiving device.

Remote command:
`SEARch:TRIGger:HBTO:ERRor:CRC` on page 1877

**SFD Error ← Errors**
Searches for a frame with invalid Start Frame Delimiter (SFD).

Remote command:
`SEARch:TRIGger:HBTO:ERRor:SFD` on page 1878

#### 13.10.5.2 100BASE-T1 search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands are described in Chapter 24.17.11.5, "Search results", on page 1878.

#### 13.10.5.3 Searching 100BASE-T1 data

Prerequisite: A serial bus is configured for the 100BASE-T1 signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".
3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Tap "Source" and select the serial bus that is set to 100BASE-T1 (e.g. "SB1", unless already selected).

   The search dialog for 100BASE-T1 protocol analysis opens.
5. Specify search criteria according to Chapter 13.10.5, "Search on decoded 100BASE-T1 data", on page 657.
6. To acquire a waveform, press [Single].

   The R&S RTP performs an 100BASE-T1 decode according to the thresholds and protocol settings of the associated serial bus source (here in our example SB1).
7. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog.

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

# 13.11 Ethernet 1000BASE-T1 (option R&S RTP-K58)

1000BASE-T1 is a single pair Ethernet based protocol specialized for the needs of the automotive industry, such as in the automotive networking applications, enabling advanced safety, comfort and infotainment features within the automobile. Its specifications is reference to IEEE P802.3bp.


- 1000BASE-T1 basics....663
- 1000BASE-T1 configuration....663
- 1000BASE-T1 trigger....667
- 1000BASE-T1 decode results....673
- Search on decoded 1000BASE-T1 data....674


## 13.11.1 1000BASE-T1 basics

The 1000BASE-T1 has the following objectives:

- Provides a PHY that supports full duplex operating at 1000 Mbps over a pair of unshielded sheathed twisted-pair cable or better cable.
- Achieves bit error rate (BER) of less than $1e^{-10}$.
- Support for up to at least 15 meters of cable length in the automotive link segment.

To achieve the above objectives, 1000BASE-T1 uses Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer as follows:

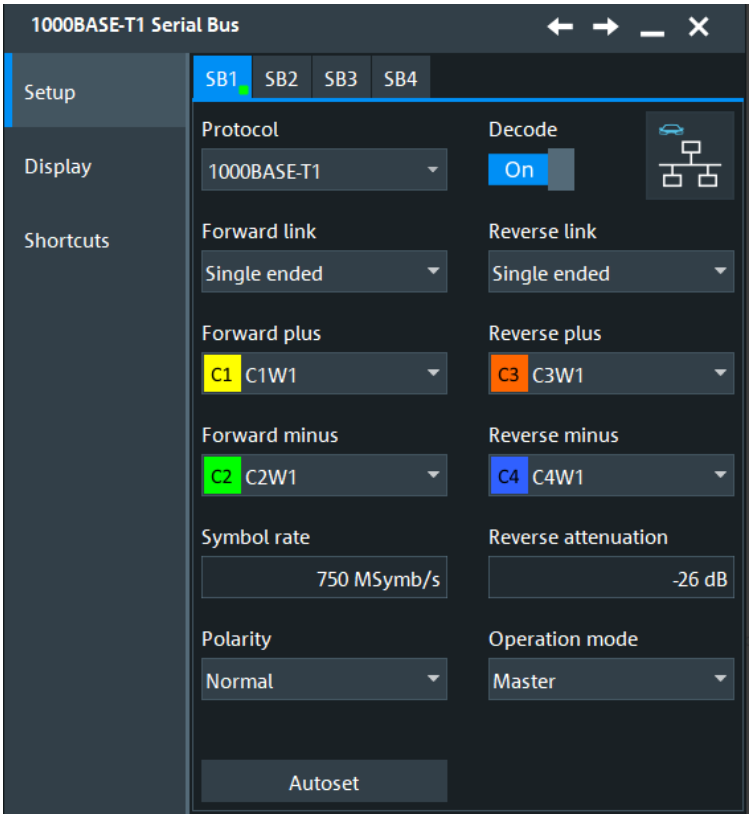
- Adopts full duplex communication of 1000BASE-T1 and therefore echo cancellation on a single twisted-pair channel. This feature reduces cabling while preserving the Ethernet MAC compatibility
- Adopts Pulse Amplitude Modulation (PAM-3) encoding scheme with the following encoding techniques:
  - PAM-3 is transmitted at 750 MB data rate.
  - It uses 15-bit scrambler to improve the EMC performance.
  - Data is encoded in 80B/81B blocks.
  - 396 bits of Reed-Solomon forward error correction (FEC) parity is added to each group of 45 80B/81B blocks.
  - 9-bit OAM symbol is added, then 396 bits Reed-Solomon. Together, they shape a PHY frame of 4050 bits, or 450 forward error correction (FEC) symbols of 9 bits. FEC allows for correction of up to 22 FEC symbols per PHY frame.

## 13.11.2 1000BASE-T1 configuration

### 13.11.2.1 1000BASE-T1 configuration settings

Access: [Bus] > "1000BASE-T1" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Forward**
Defines the forward signal settings.

You can choose between a differential and single-ended signal and set the source channel for the waveform. Only analog channels are available for selection.

Remote command:
`BUS<m>:TBTO:FTYP` on page 1885
`BUS<m>:TBTO:FDIF` on page 1884
`BUS<m>:TBTO:FDAP` on page 1884
`BUS<m>:TBTO:FDAM` on page 1884

**Reverse**
Defines the reversed signal settings. The reverse direction is optional and can be disabled.

You can choose between a differential and single-ended signal and set the source channel for the waveform. Only analog channels are available for selection.

Remote command:
`BUS<m>:TBTO:RTYP` on page 1886
`BUS<m>:TBTO:RDIF` on page 1885

BUS<m>:TBTO:RDAP on page 1885
BUS<m>:TBTO:RDAM on page 1885

**Reverse attenuation**
Sets the attenuation factor. It is used to de-amplify the reverse signal before subtracting it from the forward signal.

The effective signal amplitude passed to the decoder is:

$$Amp = (Fwd_+ - Fwd_-) - 10^{\frac{Attn}{20}}(Rev_+ - Rev_-)$$

Remote command:
BUS<m>:TBTO:ATTN on page 1883

**Polarity**
Selects the polarity of the data signal. You can select between "Normal" and "Inverted".

For "Normal" polarity, active high is defined as +1 V and active low is defined as -1 V.
For "Inverted" polarity, active high is defined as -1 V and active low is defined as +1 V.

Remote command:
BUS<m>:TBTO:POLarity on page 1886

**Operation mode**
Selects the desired direction of the full-duplex signal for analysis. If set to "Auto", the decoder automatically detects the "Master" or "Slave" mode.

Remote command:
BUS<m>:TBTO:MODE on page 1886

**Autoset**
Starts software algorithms for determining the signal threshold levels and bitrate.

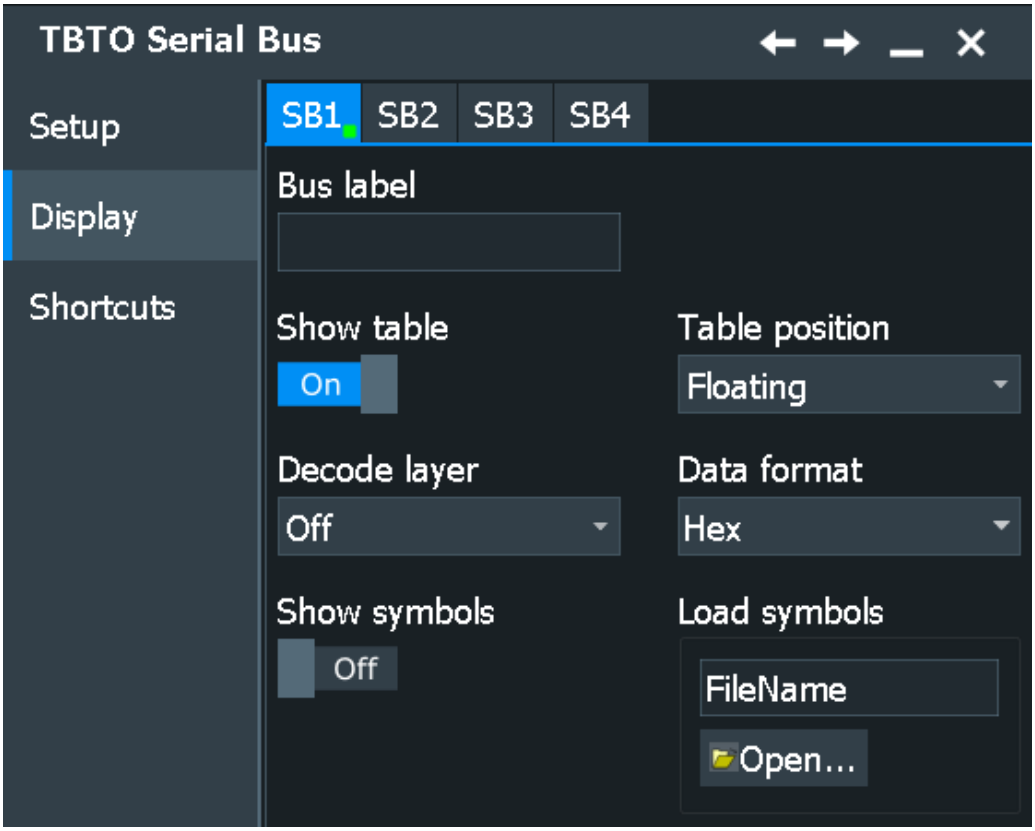
Remote command:
BUS<m>:FAUToset on page 1638

#### 13.11.2.2 1000BASE-T1 display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

To enhance the decode possibilities of the 1000BASE-T1 protocol, you can use an additional setting in the "Display" tab: "Decode layer".




**1000BASE-T1 Specific Label Settings**

Label lists are protocol-specific. Label lists for 1000BASE-T1 are available in CSV and PTT format.

A 1000BASE-T1 label file contains two values for each ethernet header:

- Ethernet header display
- Symbolic label: name of the address, specifiing its function in the bus network.

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the end results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

| | |
|---|---|
| "Off" | No decode layer is displayed. |
| "Ternary symbols" | Decoded ternary symbols. |
| "Scrambled bits" | Scrambled ternary bits. |
| "Descrambled bits" | Descrambled ternary bits. |
| "Corrected RS-FEC symbols" | Displays RS-FEC symbols that have been corrected. |

#### 13.11.2.3 Configuring 1000BASE-T1 signals

For configuration, assign the lines to the input channels and define the active states and the logical thresholds.

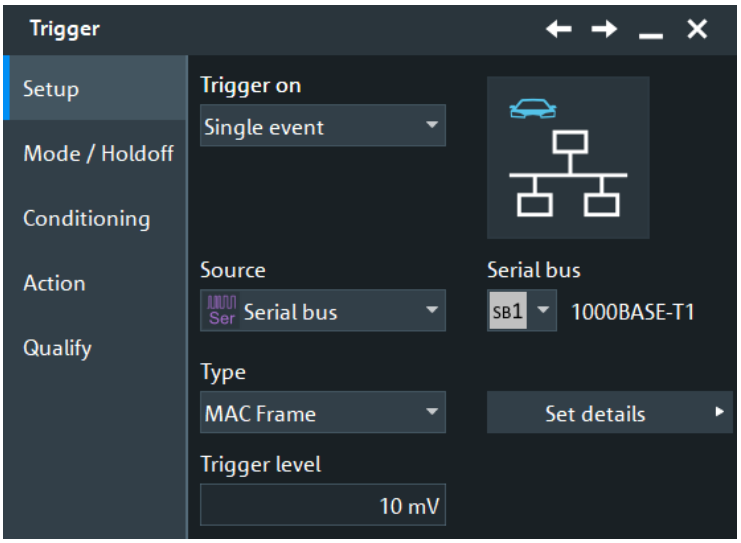
**Serial bus setup**

1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, select the protocol: "1000BASE-T1".
4. Select the type and source for the reversed and forward signal.
5. Select the "Polarity" ("Normal" or "Inverted") of the signal.
6. Enable "Decode".

### 13.11.3 1000BASE-T1 trigger

#### 13.11.3.1 1000BASE-T1 trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"



Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Type**
Selects the trigger type for 1000BASE-T1 analysis.

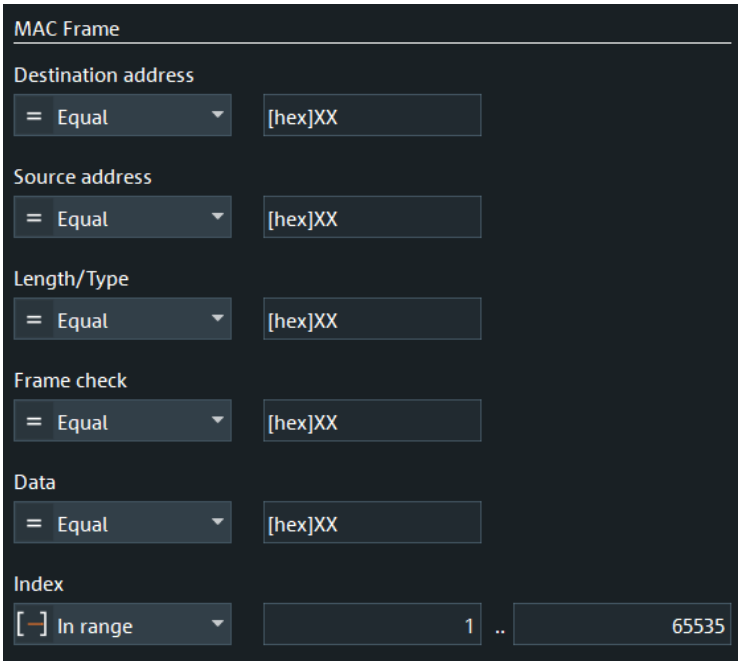
Remote command:
`TRIGger<m>:TBTO:TYPE` on page 1894

**Frame Start ← Type**
Triggers on the start of frame.

**MAC Frame ← Type**
Triggers on a media access control (MAC) frame. You can specify:

- Destination address
- Source address
- Length/Type
- Frame check
- Data
- Index

The frame contains addresses of the devices and MAC control information that define how to go about transmitting and receiving frames.




**IDLE ← Type**
Triggers on an idle frame. The frame is used for clock synchronization.

**Error ← Type**
Triggers on the specified error frame.

**Trigger level**
Sets the trigger level.

Remote command:
`TRIGger<m>:TBTO:TRGLevel` on page 1894

**Destination address**
Sets the specified destination address to be triggered on. The destination address setup consists of the condition and one or two data patterns.

The destination address corresponds to the address of the interface in the device that receives the frame. If the destination address does not match the interface's own Ethernet address, then the interface is free to ignore the rest of the frame.

"Condition" Defines the operator to set a specific destination address, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the destination address pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
TRIGger<m>:TBTO:DADDress:CONDition on page 1895
TRIGger<m>:TBTO:DADDress:MIN on page 1896
TRIGger<m>:TBTO:DADDress:MAX on page 1896

**Source address**
Sets the specified source address to be triggered on. The source address setup consists of the condition and one or two data patterns.

The source address is the physical address of the device that sends the frame.

"Condition" Defines the operator to set a specific source address, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the source address pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
TRIGger<m>:TBTO:SADDress:CONDition on page 1901
TRIGger<m>:TBTO:SADDress:MIN on page 1901
TRIGger<m>:TBTO:SADDress:MAX on page 1901

**Length/Type**
Sets the specified length/type to be triggered on. The length/type setup consists of the condition and one or two data patterns.

The value in this field indicates the manner in which the field is being used, either as a length or type field. As a length field, the value in the field indicates the number of logical link control (LLC) data octets that follow in the data field of the frame. As a type field, the value in the field is used to indicate the type of protocol data being carried in the data field of the frame.

"Condition" Defines the operator to set a specific Length/Type, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the Length/Type pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
TRIGger<m>:TBTO:LENGth:CONDition on page 1899
TRIGger<m>:TBTO:LENGth:MIN on page 1900
TRIGger<m>:TBTO:LENGth:MAX on page 1900

**Frame check**
Sets the specified frame check to be triggered on. The frame check setup consists of the condition and one or two data patterns.

This field contains a value that is used to check the integrity of the various bits in the frame fields (excluding the preamble / SFD).

"Condition" Defines the operator to set a specific frame check, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the frame check pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:TBTO:CRC:CONDition` on page 1894
`TRIGger<m>:TBTO:CRC:MIN` on page 1895
`TRIGger<m>:TBTO:CRC:MAX` on page 1895

**Data**
Sets the specified data to be triggered on. The data setup consists of the condition and one or two data patterns.

"Condition" Defines the operator to set a specific data, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the data pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:TBTO:DATA:DCONdition` on page 1896
`TRIGger<m>:TBTO:DATA:DMIN` on page 1897
`TRIGger<m>:TBTO:DATA:DMAX` on page 1897

**Index**
Sets the specified value or range within this series of data that is considered for the analysis. The index setup consists of the condition and one or two index values.

"Condition" Defines the operator to set a specific index, e.g. "Equal" or a range.

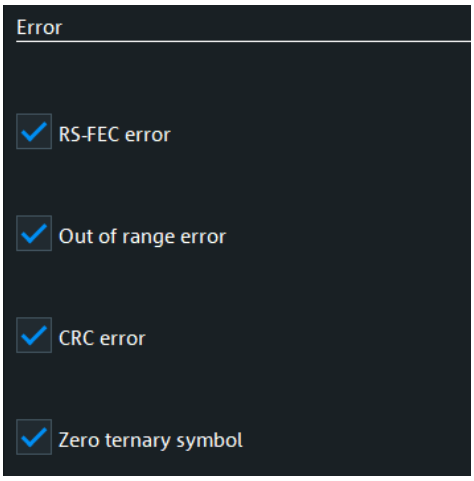
"Index Min/Index"
Defines the bit pattern of the index pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Index Max" The second index pattern is required to specify a range with conditions "In range".

Remote command:
TRIGger<m>:TBTO:DATA:ICONdition on page 1897
TRIGger<m>:TBTO:DATA:IMIN on page 1898
TRIGger<m>:TBTO:DATA:IMAX on page 1898

**Errors**
Sets the type of error events to be triggered on.




**RS-FEC error ← Errors**
Triggers, if more than 22 RS-FEC errors are detected.

Remote command:
TRIGger<m>:TBTO:ERRor:FEC on page 1899

**CRC error ← Errors**
Triggers on a frame that has a mismatch of the cyclic redundancy check (CRC) value between the transmitting and receiving device.

Remote command:
TRIGger<m>:TBTO:ERRor:CRC on page 1898

**Out of range error ← Errors**
Triggers if values are out of range. This can be, for example a preamble error, or invalid symbols in control blocks.

Remote command:
TRIGger<m>:TBTO:ERRor:OOR on page 1899

**Zero terrnary symbol ← Errors**
Triggers, if a zero ternary symbol pair (0-0) is detected.

Remote command:
TRIGger<m>:TBTO:ERRor:ZERO on page 1899

#### 13.11.3.2 Triggering on 1000BASE-T1

Prerequisite: A bus is configured for the 1000BASE-T1 signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Select the serial bus that is set to 1000BASE-T1.
5. Tap "Type" and select the trigger type to be used for 1000BASE-T1 protocol analysis.
6. To refine the trigger settings, configure additional settings, which are available for some trigger types.

For details, see Chapter 13.11.3, "1000BASE-T1 trigger", on page 667.

### 13.11.4 1000BASE-T1 decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Serial Bus" dialog > "Setup" tab, enable "Decode".
2. In the "Serial Bus" dialog > "Display" tab, select additional result display settings: "Show table" and "Show binary".
3. If required, enable "Zoom coupling"

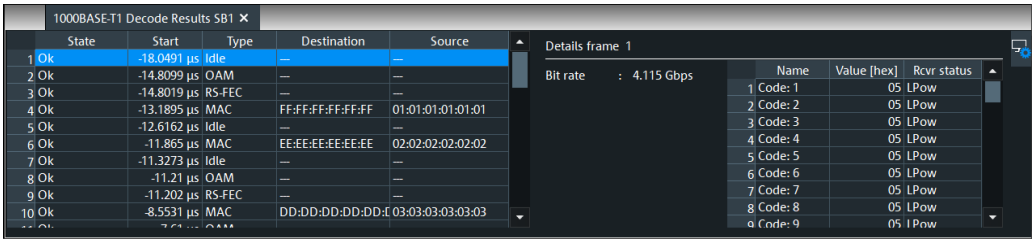
For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

#### Decode Results Table

The example shows decoded signals in 1000BASE-T1.



The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

***Table 13-15: Content of the Decode results table***

| Column | Description |
|---|---|
| Frame | Frame count |
| State | Overall state of the frame: either OK or the relevant error condition (preamble, CRC or SFD) |
| Start | Start time of the frame |
| Type | Type of frame (e.g. Idle, MAC or data) |
| Destination | Destination address of the frame |
| Source | Source address of the frame |

***Table 13-16: Content of the Details frame table***

| Column | Description |
|---|---|
| Bit rate | Value of the bit rate |
| Name | State of the field |
| Value [hex] | Hexadecimal value of the field |
| Rcvr status | Receiver status |

**Tools**

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

**Remote commands**

Remote commands are described in Chapter 24.17.12, "Ethernet 1000BASE-T1 (option R&S RTP-K58)", on page 1883.

## 13.11.5 Search on decoded 1000BASE-T1 data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

Before you can start the search, you have to configure the bus correctly and acquire decoded data.

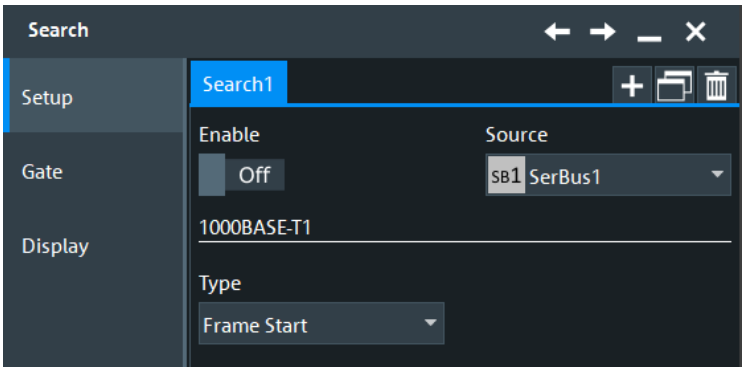
To search on decoded data, set the search source to "SerBus" for the configured protocol.

For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

#### 13.11.5.1 1000BASE-T1 search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"

**Search criteria**
Use the "Search criteria" dialog to define the event types to be searched.



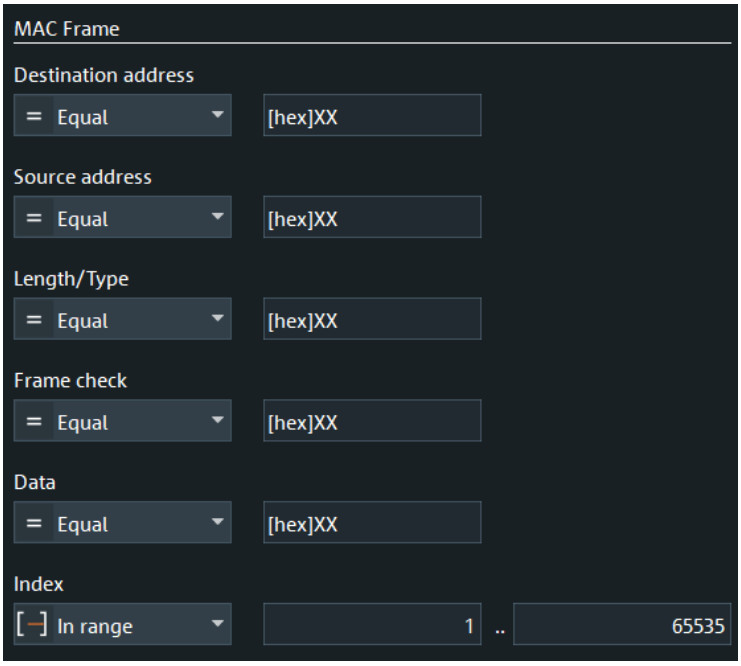

**Type**
Selects the search type for 1000BASE-T1 analysis.

Remote command:
`SEARch:TRIGger:TBTO:TYPE` on page 1910

**Frame Start ← Type**
Searches for the start of frame.

**MAC Frame ← Type**
Searches for a Media Access Control (MAC) frame that matches the specified Destination address, Source address, Length/Type, Frame check, Data or Index condition.

The frame contains addresses of the devices and MAC control information that define how to go about transmitting and receiving frames.




**IDLE ← Type**
Searches for an idle frame. The frame is used for clock synchronization.

**Error ← Type**
Searches for the specified error frame.

**Destination address**
Sets the specified destination address to be searched for. The destination address setup consists of the condition and one or two data patterns.

The destination address corresponds to the address of the interface in the device that receives the frame. If the destination address does not match the interface's own Ethernet address, then the interface is free to ignore the rest of the frame.

"Condition" Defines the operator to set a specific destination address, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the destination address pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`SEARch:TRIGger:TBTO:DADDress:CONDition` on page 1903
`SEARch:TRIGger:TBTO:DADDress:MIN` on page 1904
`SEARch:TRIGger:TBTO:DADDress:MAX` on page 1904

**Source address**
Sets the specified source address to be searched for. The source address setup consists of the condition and one or two data patterns.

The source address is the physical address of the device that sends the frame.

"Condition" Defines the operator to set a specific source address, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the source address pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`SEARch:TRIGger:TBTO:SADDress:CONDition` on page 1909
`SEARch:TRIGger:TBTO:SADDress:MIN` on page 1909
`SEARch:TRIGger:TBTO:SADDress:MAX` on page 1909

**Length/Type**
Sets the specified Length/Type to be searched for. The Length/Type setup consists of the condition and one or two data patterns.

The value in this field indicates the manner in which the field is being used, either as a length or type field. As a length field, the value in the field indicates the number of logical link control (LLC) data octets that follow in the data field of the frame. As a type field, the value in the field is used to indicate the type of protocol data being carried in the data field of the frame.

"Condition" Defines the operator to set a specific Length/Type, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the Length/Type pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`SEARch:TRIGger:TBTO:LENGth:CONDition` on page 1908
`SEARch:TRIGger:TBTO:LENGth:MIN` on page 1908
`SEARch:TRIGger:TBTO:LENGth:MAX` on page 1908

**Frame check**
Sets the specified frame check to be searched for. The frame check setup consists of the condition and one or two data patterns.

This field contains a value that is used to check the integrity of the various bits in the frame fields (excluding the preamble / SFD).

"Condition" Defines the operator to set a specific frame check, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the frame check pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`SEARch:TRIGger:TBTO:CRC:CONDition` on page 1902
`SEARch:TRIGger:TBTO:CRC:MIN` on page 1903
`SEARch:TRIGger:TBTO:CRC:MAX` on page 1903

**Data**
Sets the specified data to be searched for. The data setup consists of the condition and one or two data patterns.

"Condition" Defines the operator to set a specific data, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the data pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`SEARch:TRIGger:TBTO:DATA:DCONdition` on page 1904
`SEARch:TRIGger:TBTO:DATA:DMIN` on page 1905
`SEARch:TRIGger:TBTO:DATA:DMAX` on page 1905

**Index**
Sets the specified value or range within this series of data that is considered for the search. The index setup consists of the condition and one or two index values.

"Condition" Defines the operator to set a specific index, e.g. "Equal" or a range.

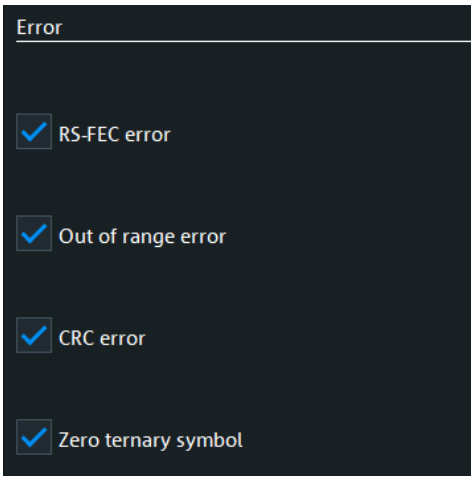
"Index Min/Index"
Defines the bit pattern of the index pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Index Max" The second index pattern is required to specify a range with conditions "In range".

Remote command:
SEARch:TRIGger:TBTO:DATA:ICONdition on page 1905
SEARch:TRIGger:TBTO:DATA:IMIN on page 1906
SEARch:TRIGger:TBTO:DATA:IMAX on page 1906

**Errors**
Sets the type of error events to be searched for.




**RS-FEC error ← Errors**
Searches for RS-FEC errors.

Remote command:
SEARch:TRIGger:TBTO:ERRor:FEC on page 1907

**CRC error ← Errors**
Searches for a frame that has a mismatch of the Cyclic Redundancy Check (CRC) value between the transmitting and receiving device.

Remote command:
SEARch:TRIGger:TBTO:ERRor:CRC on page 1906

**Out of range error ← Errors**
Searches for values that are out of range. This can be, for example a preamble error, or invalid symbols in control blocks.

Remote command:
SEARch:TRIGger:TBTO:ERRor:OOR on page 1907

**Zero terrnary symbol ← Errors**
Searches for zero ternary symbol pairs (0-0).

Remote command:
SEARch:TRIGger:TBTO:ERRor:ZERO on page 1907

#### 13.11.5.2 1000BASE-T1 search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands:

- `SEARch:RESult:TBTO:FCOunt?` on page 1911
- `SEARch:RESult:TBTO:FRAMe<m>:TYPE?` on page 1915
- `SEARch:RESult:TBTO:FRAMe<m>:STATe?` on page 1914
- `SEARch:RESult:TBTO:FRAMe<m>:STARt?` on page 1914
- `SEARch:RESult:TBTO:FRAMe<m>:STOP?` on page 1914
- `SEARch:RESult:TBTO:FRAMe<m>:DESTaddress?` on page 1911
- `SEARch:RESult:TBTO:FRAMe<m>:SRCaddress?` on page 1913
- `SEARch:RESult:TBTO:FRAMe<m>:DATA?` on page 1911
- `SEARch:RESult:TBTO:FRAMe<m>:CRC?` on page 1911
- `SEARch:RESult:TBTO:FRAMe<m>:NUMWords?` on page 1913
- `SEARch:RESult:TBTO:FRAMe<m>:FTYPe?` on page 1912
- `SEARch:RESult:TBTO:FRAMe<m>:DSYMbol?` on page 1912
- `SEARch:RESult:TBTO:FRAMe<m>:SSYMbol?` on page 1913
- `SEARch:RESult:TBTO:FRAMe<m>:WORD<n>:TYPE?` on page 1915
- `SEARch:RESult:TBTO:FRAMe<m>:WORD<n>:VALue?` on page 1915
- `SEARch:RESult:TBTO:FRAMe<m>:WORD<n>:VSTR?` on page 1916

#### 13.11.5.3 Searching 1000BASE-T1 data

Prerequisite: A serial bus is configured for the 1000BASE-T1 signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".
3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Tap "Source" and select the serial bus that is set to 1000BASE-T1 (e.g. "SB1", unless already selected).

   The search dialog for 1000BASE-T1 protocol analysis opens.

5. Specify search criteria according to Chapter 13.11.5, "Search on decoded 1000BASE-T1 data", on page 674.
6. To acquire a waveform, press [Single].

   The R&S RTP performs an 1000BASE-T1 decode according to the thresholds and protocol settings of the associated serial bus source (here in our example SB1).
7. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog.

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

## 13.12 RFFE (option R&S RTP-K40)

Radio Frequency Front-End (RFFE) control interface is a serial interface specified by the Mobile Industry Processor Interface (MIPI) alliance. The RFFE interface is designed to control RF front-end components in mobile terminals.

### 13.12.1 The RFFE protocol

The RFFE interface is specified in the "MIPI® Alliance Specification for RF Front-End Control Interface". The RFFE interface is used by the radio frequency front-end interface chips in most LTE-Advanced platforms and in smart phones in general. RFFE is a replacement for existing standards like SPI and I²C that do no meet performance requirements.

**Bus structure**

RFFE is a two-wire, serial interface that connects up to 4 master devices (Radio Frequency IC, RFIC) to up to 15 slaves (front-end modules, FEM) on a single RFFE bus. A slave device has read-write capability, or it is write only. Only one of the masters is the active master (bus owner master, BOM), which can initiate command sequences on the bus.

The interface has two lines: one clock signal (SCLK) controlled by the master, and a serial bidirectional data signal (SDATA). Furthermore, a VIO supply/reference voltage from a common source is applied to all components on the bus.

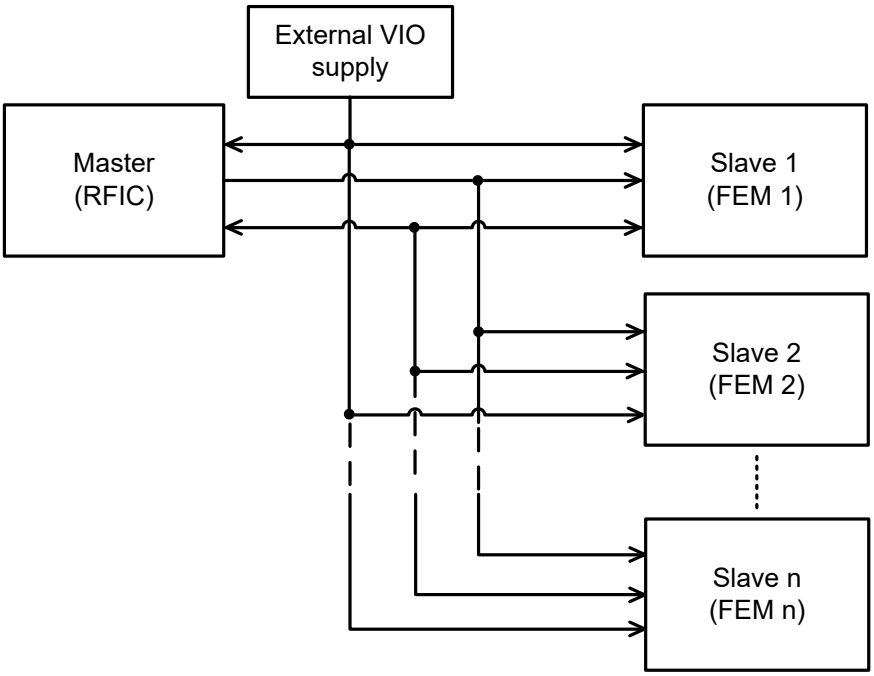



*Figure 13-15: RFFE bus structure with external VIO supply*

### Command sequences

Protocol messages are called command sequences in RFFE. The standard defines various command sequences to accomplish read and write access to slaves and to non-active masters. Command sequences are initiated by the BOM master on the SDATA line.

In general, a command sequence consists of:

- Sequence start condition (SSC)
  Two bits: 1 followed by 0 on SDATA while SCLK is at logic level zero.
- Command frame
  Consist of a 4-bit slave address field (SA), followed by 8 command payload bits and a single parity bit.
- Address and/or data frames, depending on the command sequence
  A frame consists of 8 data bits or 8 register address bits, followed by a single parity bit. The number of address and data frames varies depending on the command sequence type.
- Bus park cycle (BP)
  A BP cycle is sent at the end of a command sequence, and when the device transfers control of SDATA to another device.

Between the end of a command sequence and the beginning of a new command sequence, the bus is in idle condition at least for 10 ns.

The bits are sent MSB first.

| SSC | Command frame | | Address and data frames | | BP |
|---|---|---|---|---|---|
| | SA | Command payload | [Address frames] | [Data frames] | |
| | 4 bits | 8 bits + P | 8 bits + P | 8 bits + P | |

*Figure 13-16: General structure of a RFFE Write command sequence*

| SSC | Command frame | | Address and data frames | | | BP |
|---|---|---|---|---|---|---|
| | SA | Command payload | [Address frames] | BP | [Data frames] | |
| | 4 bits | 8 bits + P | 8 bits + P | | 8 bits + P | |

*Figure 13-17: General structure of a RFFE Read command sequence*

**Trigger**

The R&S RTP uses a hardware-based trigger to trigger on various parts of slave device messages, to trigger at maximum bus speed and on frame gaps. The data and clock lines must be connected to the input channels. Triggering on math and reference waveforms is not possible.

You can trigger on:

- Start of command sequence (SSC). In addition, you can specify a slave address.
- End of command sequence). In addition, you can specify a slave address.
- Various errors, for example, parity and bus park error
- Read and write command sequences between the BOM and the slaves.

  Within a command sequence, you can trigger on specific parts of the message:
  - Slave address
  - Byte count
  - Register address
  - Data word

**Search**

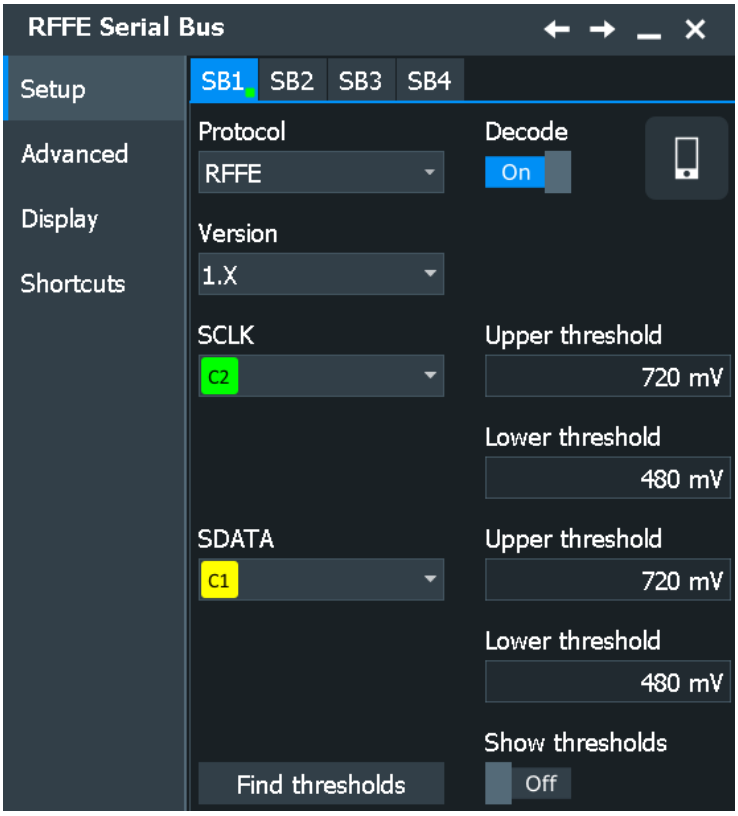
Using the search functionality, you can find various events in the acquired and decoded data. You can find the same events which you also can trigger on. In addition, you can find command sequences of master-to master communication and "interrupt summary and identification" command sequences.

## 13.12.2 RFFE configuration

### 13.12.2.1 RFFE configuration settings

Access: [Bus] > "RFFE" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Version**
Selects the RFFE version. Supported are 1.x, 2.0, and 2.1.

Remote command:
`BUS<m>:RFFE:VERSion` on page 1921

**Read mode**
Selects, if the standard or synchronous read ("sRead") mode is used.

Remote command:
`BUS<m>:RFFE:RDMD` on page 1921

**SCLK Source**
Sets the source of the clock line.

Usually, the source is one of the analog channels. Reference and math waveforms are available as source if the trigger source is one of the analog channels but not the serial bus.

Alternatively, digital channels can be used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.

For triggering on a serial bus, analog or digital channel sources are required.

Remote command:
BUS<m>:RFFE:CLOCk:SOURce on page 1917

**SDATA Source**
Sets the source of the data line.

Usually, the source is one of the analog channels. Reference and math waveforms are available as source if the trigger source is one of the analog channels but not the serial bus.

Alternatively, digital channels can be used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.

For triggering on a serial bus, analog or digital channel sources are required.

Remote command:
BUS<m>:RFFE:DATA:SOURce on page 1917

**Thresholds**
Set the threshold values for digitization of signals for each line: the positive going threshold ($V_{TP}$, upper value), the negative going threshold ($V_{TN}$, lower value), and the hysteresis. The hysteresis is the difference of $V_{TP}$ and $V_{TN}$.

If the signal value on the line is higher than the positive going threshold, the signal state is high. Otherwise, the signal state is considered low if the signal value is below the negative going threshold.

There are several ways to set the thresholds

| | |
|---|---|
| "Threshold" | Enter the value directly in the field. |
| "Find thresholds" | Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude. |

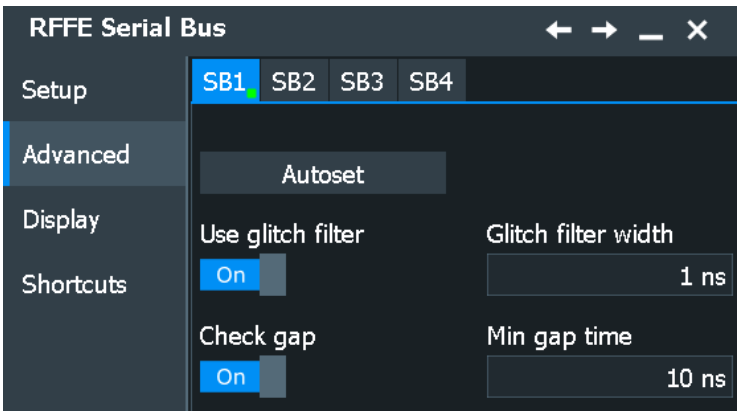
Remote command:
BUS<m>:RFFE:CLOCk:THReshold:HIGH on page 1919
BUS<m>:RFFE:CLOCk:THReshold:LOW on page 1919
BUS<m>:RFFE:CLOCk:THReshold:HYSTeresis on page 1919
BUS<m>:RFFE:DATA:THReshold:HIGH on page 1918
BUS<m>:RFFE:DATA:THReshold:LOW on page 1918
BUS<m>:RFFE:DATA:THReshold:HYSTeresis on page 1918
BUS<m>:SETReflevels on page 1638
BUS<m>:FAUToset on page 1638

**Show thresholds**
If enabled, the threshold levels are displayed in the diagram.

Remote command:
BUS<m>:THReshold on page 1638

#### 13.12.2.2 Advanced settings

Access: [Bus] > "Advanced" tab.




**Autoset**
Starts software algorithms for determining the signal threshold levels and bitrate.

Remote command:
`BUS<m>:FAUToset` on page 1638

**Glitch filter**
Enables the glitch filter on the SCLK and SDATA lines to improve decode accuracy.

The "Glitch filter width" field sets the maximum glitch width to be ignored.

Remote command:
`BUS<m>:RFFE:GFILter` on page 1920
`BUS<m>:RFFE:GFWidth` on page 1920

**Timing**
The gap defines the idle time between the Bus Park Cycle (BP) and Sequence Start Condition (SSC).

You can define a "Min gap time". The standard defines a minimum of 10 ns to separate two subsequent command sequences.

Remote command:
`BUS<m>:RFFE:MINGap:SELect` on page 1920
`BUS<m>:RFFE:MINGap:TIME` on page 1921

### 13.12.2.3 Configuring RFFE signals

For details on configuration settings, see Chapter 13.12.2.1, "RFFE configuration settings", on page 683.

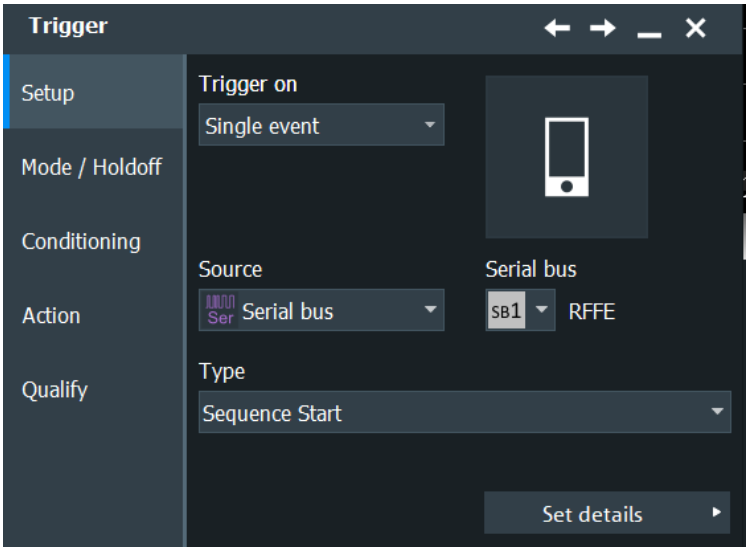
1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, select the protocol: "RFFE".
4. Select the RFFE "Version".
5. Select the "Read mode".

6. Tap "SCLK", and select the waveform of the clock line.
7. Tap "SDATA", and select the waveform of the data line.
8. Set the logical thresholds.
9. If the signal has glitches which can distort the decoding, enable the glitch filter. Set the glitch width.
10. Enable "Decode".

## 13.12.3 RFFE trigger

### 13.12.3.1 RFFE trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"




Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Version**
Selects the RFFE version. Supported are 1.x, 2.0, and 2.1.

Remote command:
BUS<m>:RFFE:VERSion on page 1921

**Type**
Selects the trigger type for RFFE analysis.

The instrument triggers always at the end of the met trigger criteria.

Remote command:
TRIGger<m>:RFFE:TYPE on page 1922

**Sequence Start ← Type**
Triggers on the beginning of a command sequence, after the slave address. The SSC is a 1 followed by 0 on SDATA line while SCLK is at logic level zero.

Optionally, you can specify a slave address pattern or a slave address range to trigger only on command sequences that are sent to these slaves.

See: "Slave Addr. /MID " on page 692.

**Sequence Stop ← Type**
Triggers on the end of a command sequence, on the bus park cycle.

Optionally, you can specify a slave address pattern or a slave address range to trigger only on command sequences that are sent to these slaves.

See: "Slave Addr. /MID " on page 692.

**Register 0 Write ← Type**
Triggers on "Register 0 Write" command sequences. This sequence sends 7 data bits in the command payload.

You can refine the trigger condition:
- Set a slave address condition to trigger only on command sequences that are sent to the specified slaves.
  See: "Slave Addr. /MID " on page 692.
- Set a data pattern condition to trigger on data patterns expected in the message.
  See: "Data" on page 693.

**Register Write ← Type**
Triggers on "Register Write" command sequences. This sequence sends the register address in the command payload, and sends one data frame.

You can refine the trigger condition:
- Set a slave address condition to trigger only on command sequences that are sent to the specified slaves.
  See: "Slave Addr. /MID " on page 692.
- Set a register address condition to trigger only on command sequences that write data to the specified register.
  See: "Address" on page 692.
- Set a data pattern condition to trigger on data patterns expected in the message.
  See: "Data" on page 693.

**Register Read ← Type**
Triggers on "Register Read" command sequences. This sequence sends the register address in the command payload, and reads back one data frame.

You can refine the trigger condition using the same settings as for the "Register Write" command sequence, see "Register Write" on page 688.

**Extended Register Write ← Type**
Triggers on "Extended Register Write" command sequences. This sequence sends the byte count of data frames in the command payload, followed by one address frame with the address of the first extended register, and up to 16 data frames.

You can refine the trigger condition:
- Set a slave address condition to trigger only on command sequences that are sent to the specified slaves.
  See: "Slave Addr. /MID " on page 692
- Set a byte count condition to trigger only on command sequences that send the specified number of data frames.
  See: "Byte count" on page 693
- Set a register address condition to trigger only on command sequences that write data to the specified register.
  See: "Address" on page 692
- Set a data pattern condition to trigger on data patterns expected in the message.
  See: "Data" on page 693 and "Index" on page 693

**Extended Register Read ← Type**
Triggers on "Extended Register Read" command sequences. This sequence sends the byte count of data frames in the command payload, followed by one address frame with the address of the first extended register, and reads back up to 16 data frames.

You can refine the trigger condition using the same settings as for the "Extended Register Write" command sequence, see "Extended Register Write" on page 689.

**Extended Register Write Long ← Type**
Triggers on "Extended Register Write" command sequences. This sequence sends the byte count of data frames in the command payload, followed by 2 address frames with the address of the first extended register, and up to 8 data frames.

You can refine the trigger condition using the same settings as for the "Extended Register Write" command sequence, see "Extended Register Write" on page 689.

**Extended Register Read Long ← Type**
Triggers on "Extended Register Write" command sequences. This sequence sends the byte count of data frames in the command payload, followed by 2 address frames with the address of the first extended register, and reads back up to 8 data frames.

You can refine the trigger condition using the same settings as for the "Extended Register Write" command sequence, see "Extended Register Write" on page 689.

**Interrupt Summary and Notification ← Type**
Available only for "Version"= "2.0"/"2.1".

Triggers on "Interrupt Summary and Notification" command sequence.

You can refine the trigger condition:
- Set a slave address condition to trigger only on command sequences that are sent to the specified MIDs.
  See: "Slave Addr. /MID " on page 692
- Set a bit count condition to trigger only on command sequences that send the specified number of data frames.
  See: "Byte count" on page 693
- Set a data pattern condition to trigger on data patterns expected in the message.
  See: "Data" on page 693

**Masked Write ← Type**
Available only for "Version"= "2.1".

Triggers on "Masked Write" command sequences. This sequence sends the register address in the command payload, and sends one mask and one data frame.

You can refine the trigger condition:
- Set a slave address condition to trigger only on command sequences that are sent to the specified MIDs.
  See: "Slave Addr. /MID " on page 692.
- Set a register address condition to trigger only on command sequences that write data to the specified register.
  See: "Address" on page 692.
- Set a mask pattern condition to trigger on a mask expected in the message.
  See: "Mask" on page 694.
- Set a data pattern condition to trigger on data patterns expected in the message.
  See: "Data" on page 693.

**Master Ownership Handover ← Type**
Available only for "Version"= "2.0"/"2.1".

Triggers on "Master Ownership Handover" command sequences.

Optionally, you can specify an MID pattern or an MID range to trigger only on command sequences that are sent to these slaves.

**Master Write ← Type**
Available only for "Version"= "2.0"/"2.1".

Triggers on "Master Write" command sequences. This sequence sends the register address in the command payload, and sends two data frames.

You can refine the trigger condition:
- Set a MID condition to trigger only on command sequences that are sent to the specified MIDs.
  See: "Slave Addr. /MID " on page 692.
- Set a register address condition to trigger only on command sequences that write data to the specified register.
  See: "Address" on page 692.
- Set a data pattern condition to trigger on data patterns expected in the message.
  See: "Data" on page 693.

**Master Read ← Type**
Available only for "Version"= "2.0"/"2.1".

Triggers on "Master Read" command sequences. This sequence sends the register address in the command payload, and reads two data frames.

You can refine the trigger condition:

- Set a MID condition to trigger only on command sequences that are sent to the specified MIDs.
  See: "Slave Addr. /MID " on page 692.
- Set a register address condition to trigger only on command sequences that write data to the specified register.
  See: "Address" on page 692.
- Set a data pattern condition to trigger on data patterns expected in the message.
  See: "Data" on page 693.

**Master Context Transfer Write ← Type**
Available only for "Version"= "2.1".

Triggers on "Master Context Transfer Write" command sequences.

You can refine the trigger condition:

- Set an MID condition to trigger only on command sequences that are sent to the specified MIDs.
  See: "Slave Addr. /MID " on page 692
- Set a byte count condition to trigger only on command sequences that send the specified number of data frames.
  See: "Byte count" on page 693
- Set a register address condition to trigger only on command sequences that write data to the specified register.
  See: "Address" on page 692
- Set a data pattern condition to trigger on data patterns expected in the message.
  See: "Data" on page 693 and "Index" on page 693

**Master Context Transfer Read ← Type**
Available only for "Version"= "2.1".

Triggers on "Master Context Transfer Read" command sequences.

You can refine the trigger condition:

- Set an MID condition to trigger only on command sequences that are sent to the specified MIDs.
  See: "Slave Addr. /MID " on page 692
- Set a byte count condition to trigger only on command sequences that send the specified number of data frames.
  See: "Byte count" on page 693
- Set a register address condition to trigger only on command sequences that write data to the specified register.
  See: "Address" on page 692
- Set a data pattern condition to trigger on data patterns expected in the message.
  See: "Data" on page 693 and "Index" on page 693

**Error conditions ← Type**
Triggers if at least one of the selected errors occurs in a frame. The instrument triggers on the first detected error.

"SSC error" Triggers when no valid SSC sequence has been found after the idle time. The instrument triggers at the invalid sequence.

"Length error" Triggers on an incorrect length of the command sequence - when additional or missing bits are detected and the sequence of bits is not as expected.

"No response" Triggers on any No Response Frame. All 9 bits of a No Response Frame, including the parity bit, are zero.

"Bus park error"
Triggers on an incorrect bus park cycle - when the instrument expects a low bit in bus park but detects high bit.

"Parity error" Triggers on incorrect parity bits.

"Unknown sequence"
Triggers when the instrument cannot detect any supported command sequence.

"Gap error" Triggers when timing gap error occurs.

"Version error" Triggers on any version errors.

Remote command:
`TRIGger<m>:RFFE:ERRor:SSC` on page 1931
`TRIGger<m>:RFFE:ERRor:LENGth` on page 1930
`TRIGger<m>:RFFE:ERRor:BP` on page 1930
`TRIGger<m>:RFFE:ERRor:PARity` on page 1930
`TRIGger<m>:RFFE:ERRor:NOResponse` on page 1930
`TRIGger<m>:RFFE:ERRor:USEQuence` on page 1931
`TRIGger<m>:RFFE:ERRor:GAP` on page 1931
`TRIGger<m>:RFFE:ERRor:VERSion` on page 1931

**Slave Addr. /MID**
Defines the address of the slave / master ID. The slave address setup consists of the condition and one or two address patterns.

"Condition" Sets the operator to trigger on a specific address pattern ("Equal" or "Not equal") or an address range.

"Slave Address (Min)" Defines the slave address pattern for all operators that require one pattern.

"Slave Adress (Max)" Defines the second address pattern that is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:RFFE:SADD:CONDition` on page 1928
`TRIGger<m>:RFFE:SADD:MIN` on page 1928
`TRIGger<m>:RFFE:SADD:MAX` on page 1929

**Address**
Defines the register address. The register address setup consists of the condition and one or two address patterns.

"Condition" Sets the operator to trigger on a specific address pattern ("Equal" or "Not equal") or an address range.

"Address (Min)" Defines the register address pattern for all operators that require one pattern.

"Adress (Max)" Defines the second address pattern that is required to specify a range with conditions "In range" and "Out of range".

Remote command:
TRIGger<m>:RFFE:ADDRess:CONDition on page 1925
TRIGger<m>:RFFE:ADDRess:MIN on page 1925
TRIGger<m>:RFFE:ADDRess:MAX on page 1925

**Byte count**
Sets the number of data frames to be read or written in the command sequence. The setting is available for all "Extended Register" command sequences, which can transfer more than one data frame.

"Condition" Sets the operator to trigger on a specific byte count ("Equal" or "Not equal") or a byte count range.

"Byte/Bit count (Min)" Defines the byte count for all operators that require one count setting.

"Byte/Bit count (Max)" Defines the second byte count value that is required to specify a range with conditions "In range" and "Out of range".

Remote command:
TRIGger<m>:RFFE:BCOunt:CONDition on page 1925
TRIGger<m>:RFFE:BCOunt:MIN on page 1926
TRIGger<m>:RFFE:BCOunt:MAX on page 1926

**Data**
Defines the data trigger condition. The data setup consists of the condition and one or two data patterns.

To define on which data frames of the sequence you want to trigger, use the Index settings.

"Condition" Sets the operator to trigger on a specific data pattern ("Equal" or "Not equal") or a data range.

"Data (Min)" Defines the data pattern for all operators that require one pattern.

"Data (Max)" Defines the second data pattern that is required to specify a range with conditions "In range" and "Out of range".

Remote command:
TRIGger<m>:RFFE:DATA:DCON on page 1926
TRIGger<m>:RFFE:DATA:DMIN on page 1927
TRIGger<m>:RFFE:DATA:DMAX on page 1927

**Index**
Defines which data frames are relevant for the trigger. The setting is available for all "Extended Register" command sequences, which can transfer more than one data frame.

The index allows you to check for a certain bit pattern at a certain position in the byte sequence. Furthermore, you can trigger on the occurrence of a certain bit pattern within a data range.

To trigger on any data byte that fulfills the data pattern settings, set the index to XX.

"Condition" Sets the operator to trigger on a specific data frame ("Equal") or a data frame range.

"Data (Min)" Defines the frame index of the data pattern for all operators that require one index.

"Data (Max)" Defines the frame index of the last data pattern that is required to specify a range with condition "In range". The instrument adjusts this value, if you enter a data pattern that is longer than the index range.

Remote command:
`TRIGger<m>:RFFE:DATA:ICONdition` on page 1927
`TRIGger<m>:RFFE:DATA:IMIN` on page 1928
`TRIGger<m>:RFFE:DATA:IMAX` on page 1928

**Mask**
Defines the mask trigger condition. The mask setup consists of the condition and one or two data patterns.The setting is available for "Masked Write" command sequences.

"Condition" Sets the operator to trigger on a specific mask frame ("Equal") or a mask frame range.

"Mask (Min)" Defines the mask pattern for all operators that require one pattern.

"Mask (Max)" Defines the second mask pattern that is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:RFFE:MASK:CONDition` on page 1929
`TRIGger<m>:RFFE:MASK:MIN` on page 1929
`TRIGger<m>:RFFE:MASK:MAX` on page 1929

### 13.12.3.2 Triggering on RFFE signals

Prerequisites: An RFFE bus is configured, see Chapter 13.12.2, "RFFE configuration", on page 683, and "Decode" is enabled.

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Tap "Serial bus" and select the serial bus that is set to RFFE.
5. Select the "Trigger type".
6. To refine the trigger settings, configure additional settings, which are available for many trigger types.
   For details, see Chapter 13.12.3.1, "RFFE trigger settings", on page 687.

### 13.12.4 RFFE label list

Label lists are protocol-specific. A label list file for RFFE contains slave addresses and/or register addresses and their symbolic names:

- "SA": slave address (SID in the labal list file)
- Address: register address.
- "Symbolic label": symbolic name as combination of the slave name (SID text) and the address name (Address text), specifying the device function

**Example: RFFE label list file**

```
# Labels for RFFE protocol
#   Column order: SID, SID text, Address, Address text
# ------------------------------------------------------------------------
@PROTOCOL_NAME = rffe
0x01,LM8335
0x01,LM8335,0x00,CMTL_REG
0x01,LM8335,0x01,GPO_PULL_DIR
0x01,LM8335,0x02,GPO_PULL_ENABLE
0x01,LM8335,0x03,GPO_OUT_HIGH_CFG
0x01,LM8335,0x04,GPO_OUT_MASK
0x01,LM8335,0x05,GPO_OUT_DATA
0x01,LM8335,0x1C,PM_TRIG
0x01,LM8335,0x1D,PROD_ID
0x01,LM8335,0x1E,MAN_ID
0x01,LM8335,0x1F,USID_REG
0x05,LM3279
0x05,LM3279,0x00,VSET_CTRL
0x05,LM3279,0x01,STATE_CTRL
0x05,LM3279,0x02,GPO_CTRL
0x05,LM3279,0x1C,PM_TRIG
0x05,LM3279,0x1D,PROD_ID
0x05,LM3279,0x1E,MAN_ID
0x05,LM3279,0x1F,USID_REG
0x0A,Test
```

See also: Chapter 13.1.4, "Content and format of the PTT file", on page 520.

Remote command:

- `BUS<m>:RFFE:SEQuence<n>:SYMBol?` on page 1935

### 13.12.5 RFFE decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".
2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode Results Table

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

***Table 13-17: Content of the Decode results table***

| Column | Description |
|---|---|
| State | Overall state of the frame.<br>"Insuffcient waveform length" indicates that the frame is not completely contained in the acquisition. Change the horizontal scale, or move the reference point to the left to get a longer acquisition. |
| Start | Time of command sequence start (SSC) |
| Type | Type of the command sequence |

***Table 13-18: Content of the Details frame table***

| Column | Description |
|---|---|
| Label | Symbolic name of the address if a label list is applied. |
| SA /MID | Address of the SA/ MID |
| Address | Register address (hex value) |
| BC | Byte count (decimal value) |
| Write bit rate | Value of the write bit rate |
| Parity control | Checks if there an error in the parity |
| Parity address 0/1 | Checks if there an error in the parity address |
| Read bit rate | Value of the read bit rate |
| Name | Type of the command sequence |

| Column | Description |
|---|---|
| State | Overall state of the frame.<br>"Insuffcient waveform length" indicates that the frame is not completely contained in the acquisition. Change the horizontal scale, or move the reference point to the left to get a longer acquisition. |
| Value | Values of the data bytes. |

**Tools**

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

**Remote commands**

Remote commands to retrieve decode results are described in Chapter 24.17.13.3, "Decode results", on page 1932.

## 13.12.6 Search on decoded RFFE data

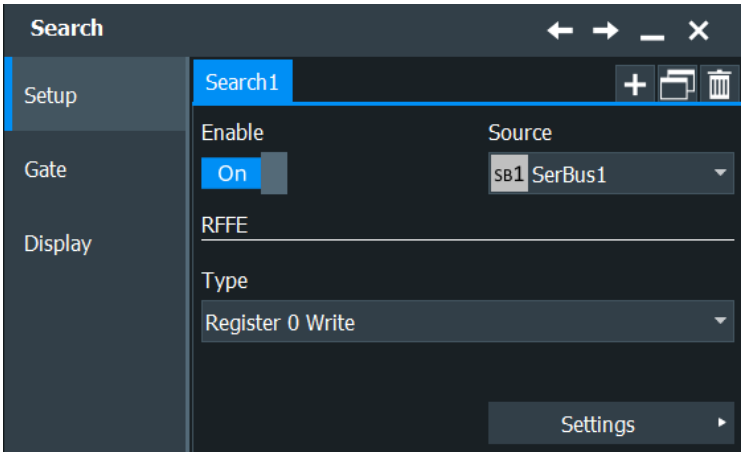
Using the search functionality, you can find various events in the decoded data, the same events which you also can trigger on. Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search "Source" to the serial bus that is configured for the protocol to be analyzed.

See also Chapter 11, "Search functions", on page 449.

### 13.12.6.1 RFFE search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"




**Type**
The search criterion is defined by "Type". All trigger types are also available for search.

Additional search parameters are provided under "Detailed Search Parameter Setup".

For details, see also "Type" on page 688.

"Sequence Start"
Searches for SSC.
Additional search parameter: "Slave Addr." on page 699

"Sequence Stop"
Searches for bus park at the end of a command sequence.
Additional search parameter: "Slave Addr." on page 699

"Register 0 Write"
Searches for Register 0 Write command sequences. Additional search parameters:
- Slave address, see "Slave Addr." on page 699.
- Data pattern, see "Data" on page 700

"Register Write"
Searches for Register Write command sequences. Additional search parameters:
- Slave address, see "Slave Addr." on page 699
- Register address, see "Address" on page 700
- Data pattern, see "Data" on page 700

"Register Read"
Searches for Register Read command sequences.
This search uses the same additional parameters as the Register Write search.

"Extended Register Write"
Searches for Extended Register Write command sequences. Additional search parameters:
- Slave address, see "Slave Addr." on page 699
- Byte count, see "Byte count" on page 699
- Register address, see "Address" on page 700
- Data pattern, see "Data" on page 700 and "Index" on page 700

"Extended Register Read"
Searches for Extended Register Read command sequences.
This search uses the same additional parameters as the Extended Register Write search.

"Extended Register Write Long"
Searches for Extended Register Write Long command sequences.
This search uses the same additional parameters as the Extended Register Write search.

"Extended Register Read Long"
Searches for Extended Register Read Long command sequences.
This search uses the same additional parameters as the Extended Register Write search.

"Master Read" Searches for Master Read command sequences. Additional search parameters:
- Address of the non-BOM master, see "Slave Addr." on page 699.
- Register address, see "Address" on page 700.
- Data pattern that is returned by the non-BOM master, see "Data" on page 700.

"Master Write" Searches for Master Write command sequences. Additional search parameters:
- Address of the non-BOM master, see "Slave Addr." on page 699
- Register address, see "Address" on page 700
- Data pattern to be written into the non-BOM, see "Data" on page 700.

"Master Ownership Handover"
Searches for Master Ownership Handover command sequences. Additional search parameters:
- Address of the BOM-to-be, see "Slave Addr." on page 699
- Data pattern of the confirmation sequence of the new BOM, see "Data" on page 700.

"Interrupt Summary and Notification"
Searches for interrupts. Additional search parameters:
- Slave address, see "Slave Addr." on page 699
- Byte count, see "Byte count" on page 699
- Data of the interrupt identification sequence, see

"Error Condition"
Searches for the selected errors, see "Error conditions" on page 700.

Remote command:
`SEARch:TRIGger:RFFE:TYPE` on page 1939

**Slave Addr.**
Searches for a slave address or a slave address range.

In command sequences transferred between masters, this is the non-BOM address.

The slave address settings are the same as in the RFFE trigger setup, see "Slave Addr. /MID " on page 692.

Remote command:
`SEARch:TRIGger:RFFE:SADD:CONDition` on page 1941
`SEARch:TRIGger:RFFE:SADD:MIN` on page 1942
`SEARch:TRIGger:RFFE:SADD:MAX` on page 1943

**Byte count**
Searches for a byte count or a byte count range.

The byte count settings are the same as in the RFFE trigger setup, see "Byte count" on page 693

Remote command:
SEARch:TRIGger:RFFE:BCOunt:CONDition on page 1941
SEARch:TRIGger:RFFE:BCOunt:MIN on page 1942
SEARch:TRIGger:RFFE:BCOunt:MAX on page 1943

**Address**
Searches for a register address or an address range.

The register address settings are the same as in the RFFE trigger setup, see "Address" on page 692.

Remote command:
SEARch:TRIGger:RFFE:ADDRess:CONDition on page 1941
SEARch:TRIGger:RFFE:ADDRess:MIN on page 1942
SEARch:TRIGger:RFFE:ADDRess:MAX on page 1943

**Data**
Searches for a data pattern or a data word range. The setting is available for all "Extended Register" command sequences, which can transfer more than one data frame.

The data settings are the same as in the RFFE trigger setup, see "Data" on page 693.

Remote command:
SEARch:TRIGger:RFFE:DATA:DCON on page 1941
SEARch:TRIGger:RFFE:DATA:DMIN on page 1942
SEARch:TRIGger:RFFE:DATA:DMAX on page 1943

**Index**
Defines which data frames are relevant for the search. The setting is available for all Extended Register command sequences, which can transfer more than one data frame. To search for any data byte that fulfills the data pattern settings, set the index to XX.

Remote command:
SEARch:TRIGger:RFFE:DATA:ICONdition on page 1942
SEARch:TRIGger:RFFE:DATA:IMIN on page 1942
SEARch:TRIGger:RFFE:DATA:IMAX on page 1943

**Data**
Defines the pattern of the interrupt identification sequence, which consists of interrupt slots 15 to 0. The setting is only available for "Interrupt Summary and Notification" command sequence.

Remote command:
SEARch:TRIGger:RFFE:INTerrupt on page 1943

**Error conditions**
Defines which error conditions are relevant for the search.

Remote command:
SEARch:TRIGger:RFFE:ERRor:LENGth on page 1943
SEARch:TRIGger:RFFE:ERRor:PARity on page 1944
SEARch:TRIGger:RFFE:ERRor:SSC on page 1944

SEARch:TRIGger:RFFE:ERRor:GAP on page 1944
SEARch:TRIGger:RFFE:ERRor:VERSion on page 1944

### 13.12.6.2 RFFE search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands:

- SEARch:RESult:RFFE:FCOunt? on page 1945
- SEARch:RESult:RFFE:SEQuence<m>:TYPE? on page 1945
- SEARch:RESult:RFFE:SEQuence<m>:STATe? on page 1946
- SEARch:RESult:RFFE:SEQuence<m>:STARt? on page 1946
- SEARch:RESult:RFFE:SEQuence<m>:STOP? on page 1947
- SEARch:RESult:RFFE:SEQuence<m>:SADD? on page 1946
- SEARch:RESult:RFFE:SEQuence<m>:ADDRess? on page 1947
- SEARch:RESult:RFFE:SEQuence<m>:BCOunt? on page 1947
- SEARch:RESult:RFFE:SEQuence<m>:DATA? on page 1948
- SEARch:RESult:RFFE:SEQuence<m>:SYMBol? on page 1948
- SEARch:RESult:RFFE:SEQuence<m>:BYTE<n>:STATe? on page 1948
- SEARch:RESult:RFFE:SEQuence<m>:BYTE<n>:VALue? on page 1948

### 13.12.6.3 Searching RFFE data

Prerequisites: An RFFE bus is configured, see Chapter 13.12.2, "RFFE configuration", on page 683, and "Decode" is enabled.

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".
3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Tap the "Source" button and select the serial bus that is set to RFFE.

   "Protocol" shows the RFFE icon.
5. Tap "Type" and select the search type.
   All trigger types are also available for search.

6. To refine the search settings, configure additional settings, which are available for many search types.
   For details, see Chapter 13.12.6.1, "RFFE search setup", on page 697.
7. Under "Control", "Enable" the search.

   The "Search Results" box opens.
8. Close the "Search" dialog box.
9. Press [Run / Stop] to start acquisition.
10. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog:

    The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

## 13.13 D-PHY (option R&S RTP-K42)

The D-PHY is a specification developed by the Mobile Industry Processor Interface (MIPI) alliance as a standard for the communication of high-speed components, like cameras and displays, within mobile devices.

- D-PHY basics....702
- D-PHY configuration....703
- D-PHY trigger....707
- D-PHY decode results....712
- Search on decoded D-PHY data....713

### 13.13.1 D-PHY basics

A D-PHY interface consists of one clock lane and up to four data lanes. The D-PHY data lanes have two operational modes:

- A high speed mode (HS): differential signal with a data rate of 80 Mbps to 1.5GHz.This mode is used for the transmission of large volumes of information.
- A low power mode (LP): single-ended signal with a data rate < 10 Mbps. This mode is used for conserving power .

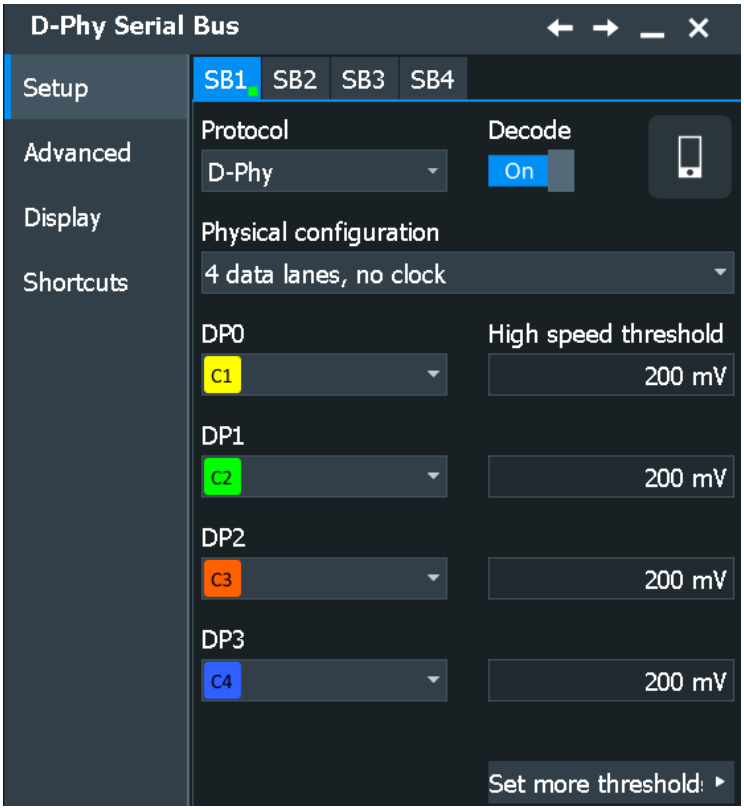
D-PHY provides a framework for other protocols such as the Display Serial Interface (DSI) and Camera Serial Interface (CSI-2).

## 13.13.2 D-PHY configuration

### 13.13.2.1 D-PHY configuration settings

Access: [Bus] > "D-PHY" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Variant**
Selects the protocol running on the interface. You can select between the Display Serial Interface (DSI) and Camera Serial Interface (CSI-2).

Remote command:
`BUS<m>:DPHY:VARiant` on page 1952

**Physical configuration**
Selects the number of data lanes and if the clock lane and low power are available.

Remote command:
`BUS<m>:DPHY:CONFig` on page 1950

**Source Connection**
Defines the source and the type of probe used for the respective lane.

Permitted source selections are none, the analog, mathematical, and reference channels.

**CP ← Source Connection**
Sets the source of the clock lane.

Remote command:
BUS<m>:DPHY:CP:SOURce on page 1950

**DP0/DN0 ← Source Connection**
Selects the source of the low-power data lanes.

The "DP0" lane has to be present all the time, hence "Source DP0">"None" is not allowed.

Remote command:
BUS<m>:DPHY:DPZero:SOURce on page 1953
BUS<m>:DPHY:DNZero:SOURce on page 1953

**DP1/DP2/DP3 ← Source Connection**
Selects the source of the high-speed data lanes.

Remote command:
BUS<m>:DPHY:DPONe:SOURce on page 1953
BUS<m>:DPHY:DPTWo:SOURce on page 1953
BUS<m>:DPHY:DPTHree:SOURce on page 1953

**Probe ← Source Connection**
Selects the type of probe used for the respective lane. The D-PHY low-power mode uses a single-ended signal, while the high-speed mode signaling is differential.

Remote command:
BUS<m>:DPHY:CP:PROBe on page 1950
BUS<m>:DPHY:DNZero:PROBe on page 1953
BUS<m>:DPHY:DPZero:PROBe on page 1953
BUS<m>:DPHY:DPONe:PROBe on page 1954
BUS<m>:DPHY:DPTWo:PROBe on page 1954
BUS<m>:DPHY:DPTHree:PROBe on page 1954

**Data Rate**
Sets a data rate.

Remote command:
BUS<m>:DPHY:DRATe on page 1951
BUS<m>:DPHY:DSPData on page 1951

**HS Threshold**
Sets the threshold value for the digitization of high-speed data line.

Remote command:
BUS<m>:DPHY:DPZero:HSPeed:THReshold on page 1954
BUS<m>:DPHY:DPONe:HSPeed:THReshold on page 1954
BUS<m>:DPHY:DPTWo:HSPeed:THReshold on page 1954
BUS<m>:DPHY:DPTHree:HSPeed:THReshold on page 1954

**Set thresholds**
Opens a dialog to set the threshold values.

**HS Hysteresis ← Set thresholds**
Sets a value for the hysteresis of the respective lane.

Remote command:
BUS<m>:DPHY:CP:HSPeed:HYSTeresis on page 1954
BUS<m>:DPHY:DPZero:HSPeed:HYSTeresis on page 1954
BUS<m>:DPHY:DPONe:HSPeed:HYSTeresis on page 1954
BUS<m>:DPHY:DPTWo:HSPeed:HYSTeresis on page 1954
BUS<m>:DPHY:DPTHree:HSPeed:HYSTeresis on page 1954

**Low power thresholds ← Set thresholds**
Sets the thresholds for the low-power mode.

**DP0 High / DN0 High ← Low power thresholds ← Set thresholds**
Sets the high-power threshold value for the respective lane.

Remote command:
BUS<m>:DPHY:DNZero:LPOWer:THUPper on page 1952
BUS<m>:DPHY:DPZero:LPOWer:THUPper on page 1952

**DP0 Low / DN0 Low ← Low power thresholds ← Set thresholds**
Sets the low-power threshold value for the respective lane.

Remote command:
BUS<m>:DPHY:DNZero:LPOWer:THLower on page 1953
BUS<m>:DPHY:DPZero:LPOWer:THLower on page 1953

**Couple thresholds ← Set thresholds**
Enables coupling, i.e. the same threshold and hysteresis value is used for all lanes.

Remote command:
BUS<m>:DPHY:THCoupling on page 1951

### 13.13.2.2 Display settings

Access: [Bus] > "Configuration" tab > "Protocol = D-PHY" > "Display" tab

To enhance the decode possibilities of the D-PHY protocol, you can use an additional setting in the "Display" tab: "Decode layer".

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the end results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

"HS Edges" The high-speed edges of each lane, showing the transitions in high-speed mode.

"HS Binary" All high-speed bits of each lane, whether they are part of a burst or not.

"HS Burst Bits" The filtered high-speed bits of each lane , including only bits that are part of a burst.

"HS Burst Words"
8 bits per word in a high-speed burst of each lane.

"HS Merged Bytes" The words from the previous layer after they have been merged into one honeycomb.

"HS Merged Words"
The bytes that are combined/split into words.

"LP Edges" The combined edge display of DP0 and DN0.

"LP States" The low-power states LP00, LP01, LP10, LP11 of DP0 and DN0.

**Show multiple lanes**
Enables the display of the results in multiple lanes.

Remote command:
`BUS<m>:DPHY:SMLanes` on page 1955

### 13.13.2.3 Configuring the D-PHY signals

For configuration, assign the lines to the input channels and define the logical thresholds and the hysteresis.

1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, select the protocol: "D-PHY".
4. Tap the "Variant" button, and select the protocol.
5. Enter the number of "Data Lanes".
6. Select the source and type of "Probe", for each lane.
7. Enter the "HS threshold" and the "Hysteresis" for each data lane.
8. Enter the low-power thresholds.
9. Enable "Decode".

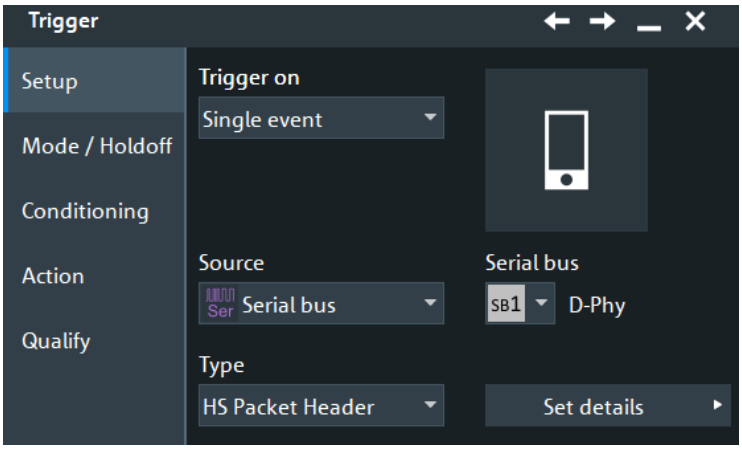
## 13.13.3 D-PHY trigger

### 13.13.3.1 D-PHY trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"

In this section, all trigger settings are described. Their availability on the instrument depends on the selected USB protocol type and trigger type. The user interface of the instrument displays only appropriate settings and guides you through the trigger setup.

For a list of supported trigger conditions, refer to data sheet.




Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Serial bus**
Selects the serial bus to be triggered on. Make sure to select the correct bus before you enter the settings.

To trigger on a serial bus, the signals sources must be channel signals. If the data or clock source is a math or reference waveform, you cannot trigger on that bus.

Remote command:
`TRIGger<m>:SOURce:SBSelect` on page 1642

**Protocol**
Defines the protocol type of the selected serial bus.

Remote command:
BUS<m>:TYPE on page 1637

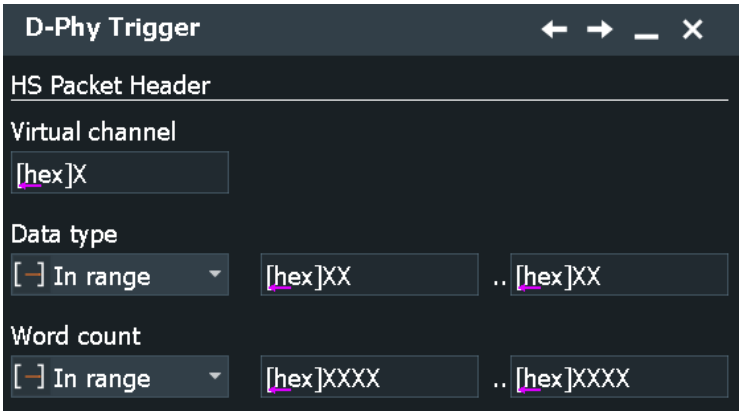
**Type**
Selects the trigger type for the D-PHY analysis.

Remote command:
TRIGger<m>:DPHY:TYPE on page 1956

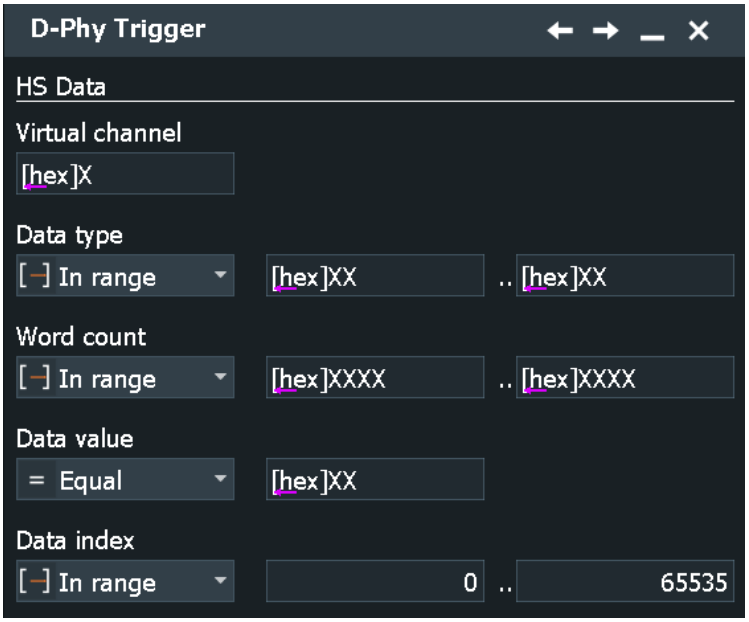
**HS Start of Packet ← Type**
Sets the trigger to the transmission start of a high-speed packet.

**HS End of Packet ← Type**
Sets the trigger to the transmission end of a high-speed packet.

**HS Packet Header ← Type**
Sets the trigger to a packet header of a high-speed package. The header consists of a data identifier (containing the Virtual Channel and the Data Type) and a Word Count. You can specify the values each part of the packet header.




**HS Data ← Type**
Sets the trigger to a specified high-speed data.




**LP Escape Mode ← Type**
Sets the trigger to an escape mode event.

**LP Lane Turnaround ← Type**
Sets the trigger to a low-power turnaround, a reversion of the transmission direction.

**LP HS Request ← Type**
Sets the trigger to a high-speed request.

**Virtual Channel**
Triggers on a specific virtual channel, an independent data stream for up to four peripherals.

Remote command:
`TRIGger<m>:DPHY:HSVC` on page 1961

**Data Type**
Sets the specified data type to be triggered on. The data type setup consists of the condition and one or two data patterns.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"Data Min/ Data" Specifies the value or sets the start value of a range.

"Data Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger<m>:DPHY:DTYPe:CONDition` on page 1958
`TRIGger<m>:DPHY:DTYPe:MAX` on page 1958
`TRIGger<m>:DPHY:DTYPe:MIN` on page 1959

**Word Count**
Sets the specified specific word count to be triggered on. The word count setup consists of the condition and one or two data patterns.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"Data Min/ Data" Specifies the value or sets the start value of a range.

"Data Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger<m>:DPHY:WCOunt:CONDition` on page 1962
`TRIGger<m>:DPHY:WCOunt:MAX` on page 1962
`TRIGger<m>:DPHY:WCOunt:MIN` on page 1962

**Data Value**
Set the specified data value to be triggered on. The data value setup consists of the condition and one or two data value patterns.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"Data Min/ Data" Specifies the value or sets the start value of a range.

"Data Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
For "Type">"HS Data":
`TRIGger<m>:DPHY:DATA:CONDition` on page 1956
`TRIGger<m>:DPHY:DATA:MAX` on page 1957
`TRIGger<m>:DPHY:DATA:MIN` on page 1957
For "Type">"LP Escape Mode":
`TRIGger<m>:DPHY:ESDTa:CONDition` on page 1960
`TRIGger<m>:DPHY:ESDTa:MAX` on page 1960
`TRIGger<m>:DPHY:ESDTa:MIN` on page 1960

**Data Index**
Data index sets the range within this series of the data value that is considered for the analysis. The data index setup consists of the condition and one or two data index values.

"Condition" Sets the comparison condition to a specific value or a range.

"Index Min/Index" Specifies the value or sets the start value of a range.

"Index Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
For "Type">"HS Data":
`TRIGger<m>:DPHY:DIDX:CONDition` on page 1957
`TRIGger<m>:DPHY:DIDX:MAX` on page 1957
`TRIGger<m>:DPHY:DIDX:MIN` on page 1958
For "Type">"LP Escape Mode":
`TRIGger<m>:DPHY:ESINdex:CONDition` on page 1961
`TRIGger<m>:DPHY:ESINdex:MAX` on page 1961
`TRIGger<m>:DPHY:ESINdex:MIN` on page 1961

**Escape mode**
The escape mode setup consists of the condition and one or two data patterns.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"Data Min/ Data" Specifies the value or sets the start value of a range.

"Data Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
TRIGger<m>:DPHY:ESCMode:CONDition on page 1959
TRIGger<m>:DPHY:ESCMode:MAX on page 1959
TRIGger<m>:DPHY:ESCMode:MIN on page 1959

#### 13.13.3.2 Triggering on D-PHY

Prerequisite: A bus is configured for the D-PHY signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Tap "Serial bus" and select the serial bus that is set to D-PHY.
5. Select the "Trigger Type" to be used for D-PHY protocol analysis.
6. To refine the trigger settings, configure additional settings, which are available for some trigger types.
   For details, see Chapter 13.13.3.1, "D-PHY trigger settings", on page 707.

### 13.13.4 D-PHY decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Serial Bus" dialog > "Setup" tab, enable "Decode".
2. In the "Serial Bus" dialog > "Display" tab, select additional result display settings: "Show table" and "Show binary".
3. If required, enable "Zoom coupling"

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

If a frame is marked red, some kind of error occurred. To analyze the error, see the indication of the corresponding frame in the result table.

ECC checks are supported. ECC can correct a certain number of bit errors in the header. If the errors can be fixed, the status is an ECC warning. If there are too many errors, an ECC error is shown.

### Decode Results Table

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

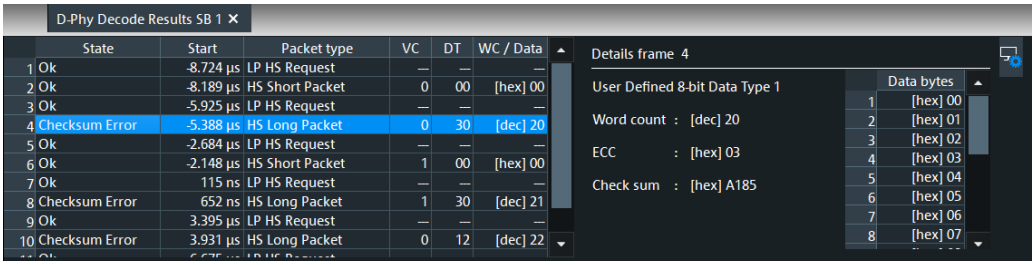



*Table 13-19: Content of the Decode results table*

| Column | Description |
|---|---|
| State | Overall state of the frame |
| Start | Time of frame start in relation to the trigger point |
| Packet Type | Packet type |
| VC | Number of virtual channels |
| DT | Data type |
| WC/ Data | Word count or data value |
| Number of Packets | Number of packets in the frame |

### Tools

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

### Remote commands

Remote commands are described in Chapter 24.17.14.3, "D-PHY decode results", on page 1962.

## 13.13.5 Search on decoded D-PHY data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

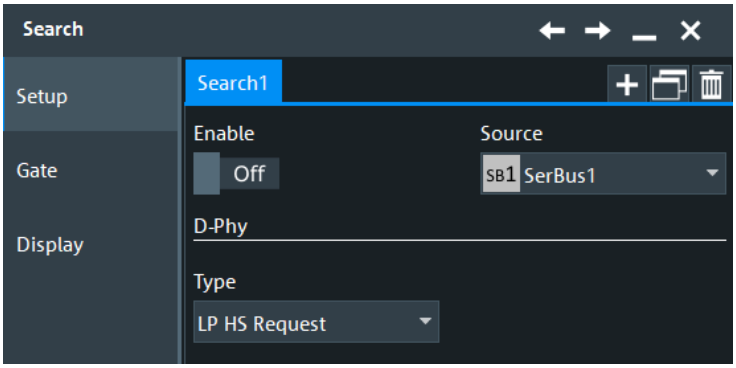
Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search source to "SerBus" for the configured protocol.

For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

#### 13.13.5.1 D-PHY search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"




**Type**
Searches for the selected D-PHY type.

Remote command:
`SEARch:TRIGger:DPHY:TYPE` on page 1969

**HS Start of Packet ← Type**
Searches for a transmission start of a high-speed packet.

**HS End of Packet ← Type**
Searches for a transmission end of a high-speed packet.

**HS Packet Header ← Type**
Searches for a packet header of a high-speed package. The header consists of a data identifier (containing the Virtual Channel and the Data Type) and a Word Count/ Data. You can specify the values each part of the packet header.

**HS Data ← Type**
Searches for a specified high-speed data.

**LP Escape Mode ← Type**
Searches for an escape mode event.

**LP Lane Turnaround ← Type**
Searches for a low-power turnaround, a reversion of the transmission direction.

**LP HS Request ← Type**
Searches for a high-speed request.

**Virtual Channel**
Searches for a specific virtual channel, an independent data stream for up to four peripherals.

Remote command:
SEARch:TRIGger:DPHY:HSVC on page 1976

**Data Type**
Searches for a specific data type. The data type setup consists of the condition and one or two data patterns.

"Condition" Defines the operator to set a specific data type ("Equal" or "Not equal") or a data type range.

"Data Min/Data"
Defines the bit pattern of the data pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
SEARch:TRIGger:DPHY:DTYPe:CONDition on page 1972
SEARch:TRIGger:DPHY:DTYPe:MAX on page 1972
SEARch:TRIGger:DPHY:DTYPe:MIN on page 1972

**Word Count/ Data**
Searches for a specific word count /data. The word count setup consists of the condition and one or two data patterns.

"Condition" Defines the operator to set a specific word count ("Equal" or "Not equal") or a word count range.

"Data Min/Data"
Defines the bit pattern of the word pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second word pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
SEARch:TRIGger:DPHY:WCOunt:CONDition on page 1976
SEARch:TRIGger:DPHY:WCOunt:MAX on page 1977
SEARch:TRIGger:DPHY:WCOunt:MIN on page 1977

**Data Value**
Searches for a specific data value. The data value setup consists of the condition and one or two data value patterns.

"Condition" Defines the operator to set a specific data value ("Equal" or "Not equal") or a data value range.

"Data Min/Data"
Defines the bit pattern of the data value pattern.
In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data value pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
For "Type">"HS Data":
`SEARch:TRIGger:DPHY:DATA:CONDition` on page 1969
`SEARch:TRIGger:DPHY:DATA:MAX` on page 1970
`SEARch:TRIGger:DPHY:DATA:MIN` on page 1970
For "Type">"LP Escape Mode":
`SEARch:TRIGger:DPHY:ESDTa:CONDition` on page 1974
`SEARch:TRIGger:DPHY:ESDTa:MAX` on page 1974
`SEARch:TRIGger:DPHY:ESDTa:MIN` on page 1975

**Data Index**
Data index sets the range within this series of the data value to be searched for. The data index setup consists of the condition and one or two data index values.

"Condition" Defines the operator to set a specific data ("Equal") or a data range.

"Index Min/Index"
Defines the minimum index.

"Index Max" The second index pattern is required to specify a range with conditions "In range".

Remote command:
For "Type">"HS Data":
`SEARch:TRIGger:DPHY:DIDX:CONDition` on page 1971
`SEARch:TRIGger:DPHY:DIDX:MAX` on page 1971
`SEARch:TRIGger:DPHY:DIDX:MIN` on page 1971
For "Type">"LP Escape Mode":
`SEARch:TRIGger:DPHY:ESINdex:CONDition` on page 1975
`SEARch:TRIGger:DPHY:ESINdex:MAX` on page 1975
`SEARch:TRIGger:DPHY:ESINdex:MIN` on page 1976

**Escape mode**
Searches for an escape mode event. The escape mode setup consists of the condition and one or two data patterns.

"Condition" Defines the operator to set a specific word count ("Equal" or "Not equal") or a word count range.

"Data Min/Data"
Defines the bit pattern of the data pattern.
In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
SEARch:TRIGger:DPHY:ESCMode:CONDition on page 1973
SEARch:TRIGger:DPHY:ESCMode:MAX on page 1973
SEARch:TRIGger:DPHY:ESCMode:MIN on page 1973

#### 13.13.5.2 D-PHY search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands:

- SEARch:RESult:DPHY:FCOunt? on page 1978
- SEARch:RESult:DPHY:FRAMe<m>:CS? on page 1978
- SEARch:RESult:DPHY:FRAMe<m>:DATA? on page 1978
- SEARch:RESult:DPHY:FRAMe<m>:DTName? on page 1978
- SEARch:RESult:DPHY:FRAMe<m>:DTYPe? on page 1979
- SEARch:RESult:DPHY:FRAMe<m>:ECC? on page 1979
- SEARch:RESult:DPHY:FRAMe<m>:PACKet<n>:IDX? on page 1979
- SEARch:RESult:DPHY:FRAMe<m>:PACKet<n>:VALue? on page 1980
- SEARch:RESult:DPHY:FRAMe<m>:STARt? on page 1980
- SEARch:RESult:DPHY:FRAMe<m>:STATe? on page 1980
- SEARch:RESult:DPHY:FRAMe<m>:STOP? on page 1981
- SEARch:RESult:DPHY:FRAMe<m>:TYPE? on page 1981
- SEARch:RESult:DPHY:FRAMe<m>:VCHannel? on page 1982

#### 13.13.5.3 Searching D-PHY

Prerequisite: A serial bus is configured for the D-PHY signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".
3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.

4. Tap "Source" and select the serial bus that is set to D-PHY (e.g. "SB1", unless already selected).
5. Specify search criteria according to Chapter 13.13.5.1, "D-PHY search setup", on page 714.
6. To acquire a waveform, press [Single].

   The R&S RTP performs a D-PHY decode according to the thresholds and protocol settings of the associated serial bus source.
7. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog:

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

## 13.14 M-PHY and USB SSIC (option R&S RTP-K44 and K64)

The M-PHY® is a serial communication protocol. Its specifications were developed by the Mobile Industry Processor Interface (MIPI) alliance as standards for the communication of high-speed components, like smartphones and tablets, within mobile devices. For more details on the specifications, refer to http://mipi.org/specifications/physical-layer#M-PHY Specification.

This protocol is the foundation for several upper layer protocols, such as UniPro℠ (Unified Protocol), which manage complex data transfer functions. Each of these protocols is optimized for its particular purpose, such as data storage, data transfer, display, camera, memory sharing and radio interface.

When implemented together with the MIPI UniPro, defined as UniPort-M, they deliver high speed, low power, scalable and flexible data transport solutions.

The super speed inter-chip (SSIC) was developed by the USB standards committee based on USB3.1 G1 (SuperSpeed). It is developed to support the chip to chip communication of USB3 components. The lower communication layers are using M-PHY as the encoder platform and use a multilane distribution similar to UniPro. The higher communication layers (above 8b/10b) are almost identical to USB3.1 G1.

- M-PHY basic ..... 719
- M-PHY configuration ..... 720
- M-PHY trigger ..... 723
- M-PHY decode results ..... 738
- Search on decoded M-PHY data ..... 739

## 13.14.1 M-PHY basic

This chapter provides an overview of the characteristics of the M-PHY and UniPro protocols.

### 13.14.1.1 Data transfer

M-PHY words are always 10 bits long (both data and control words). For UniPro, they have variable lengths that depend on the frame type. For example, control words have 8 bits, data words have 16 bits and reserved words have 3 bits.

To achieve power efficiency and high-performance transmissions, multiple power-saving states and recovery times are utilized. Scalability and flexibility are achieved through the various transmission speed ranges and rates.

Different modes of operations:

- Disable mode: lowest power mode entered into once the power supply is turned on
- Hibernate (Hibern8): ultra low-power state, which can be used without configuration loss
- High-speed mode (HS): supports three gears with predefined data rates. This mode is used during high-speed transmission for transporting large volume of data. This mode utilizes the power-saving Stall state to reduce power consumption while offering a fast state transition in the range of nanoseconds
- Low-power mode (PWM): supports seven gears with predefined frequency ranges. This mode is used during low-speed transmission for power conservation. This mode utilizes the power-saving Sleep state with a state transition time typically in the range of microseconds

The R&S RTP supports all operating speed modes: high-speed and low-power mode. The gear settings of these modes are auto detected by the instrument.

### 13.14.1.2 Data analysis

The M-PHY and UniPro decoding process involves several stages, similar to D-PHY.

The stages are as the following:

- Stage 1: involves two substages:
  - Stage 1a: converts the three stage cross points into bit and attempts a HS-Sync. This stage involves the Hibern8-Detection/Filter, NRZ Unclocked Decoder and Burst Detection.
  - Stage 1b is only involved if Stage 1a burst detection fails, assuming using PWM. Input is from the bits from stage 1a. This stage involves the PWM Decoder, Line CFG Decoder and Burst Detection.
- Stage 2: performs the 8b/10b decoding of bits from stage 1 into Bytes and identifies the 8b/10b control words. This stage involves the 8b/10b Decoder and Shift-decoder to Bytes.
- Stage 3: starts decoding the Bytes from stage 2. This stage involves the Descrambler and 17-bit Shift Decoder.

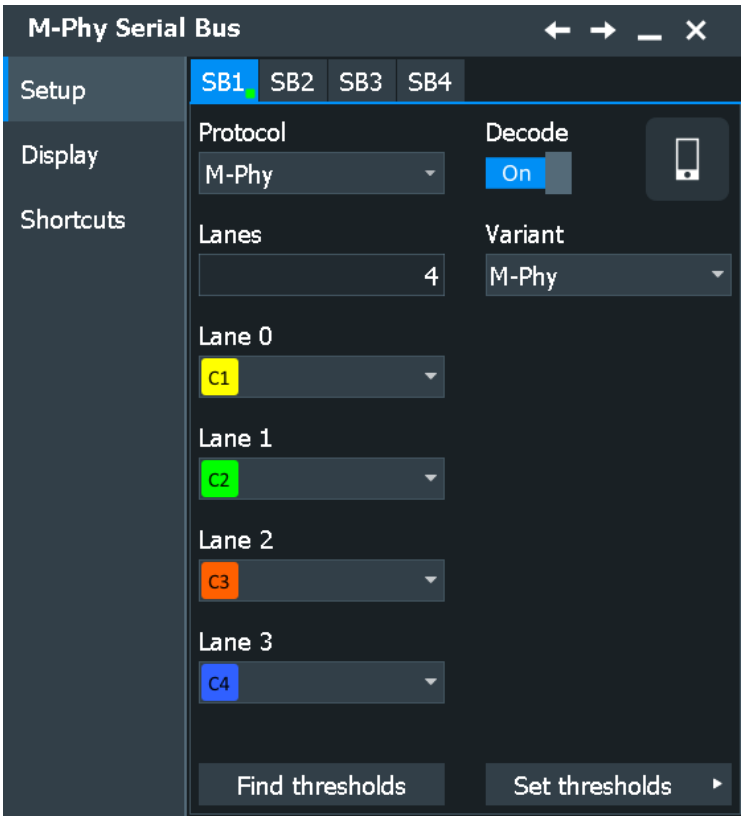
- Stage 4: merges the data lanes and finalizes the decoding to the UniPro data frames. This stage involves the Lane Merger, PDU Sync Detector, Shift Decode and Dynamic Data unit Decoder.

## 13.14.2 M-PHY configuration

### 13.14.2.1 M-PHY configuration settings

Access: [Bus] > "M-PHY" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Variant**
Selects the protocol running on the interface. You can select between the M-PHY, UniPro and SSIC.

Remote command:
`BUS<m>:MPHY:VARiant` on page 1983

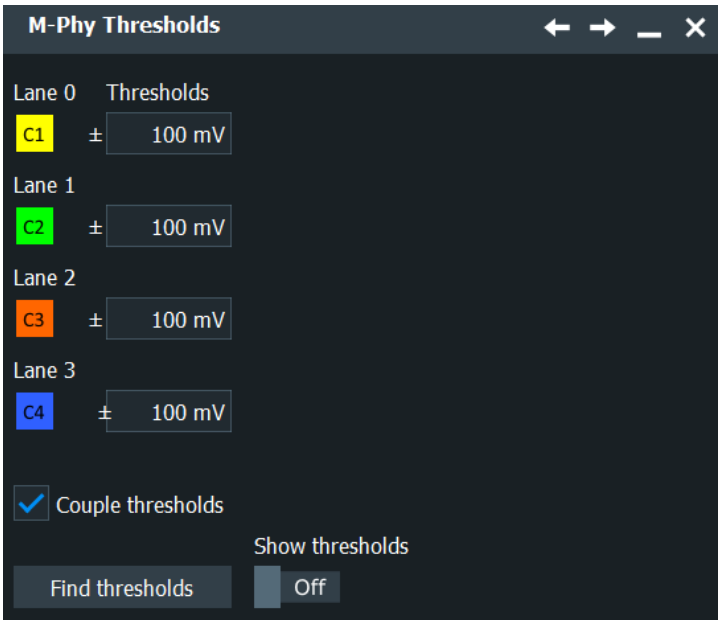
**Lanes**
Sets the number of logical lanes that are mapped to the physical lines. You can select up to four lanes.

Remote command:
BUS<m>:MPHY:DLANes on page 1983

**Lane 0 / Lane 1 / Lane 2 / Lane 3: Channel**
Select the signal sources for the logical lanes.

Remote command:
BUS<m>:MPHY:DZERo:SOURce on page 1983
BUS<m>:MPHY:DONE:SOURce on page 1984
BUS<m>:MPHY:DTWO:SOURce on page 1984
BUS<m>:MPHY:DTHRee:SOURce on page 1984

**Threshold setup**
Opens a dialog, where you can set the tresholds for digitization of signals for each lane.




"Thresholds" Enter individual values for each line directly in the fields.

"Couple thresholds" Enables the same threshold value for all lanes.

"Find thresholds" Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

"Show thresholds" If enabled, the threshold levels are displayed in the diagram.

Remote command:
BUS<m>:MPHY:DZERo:THReshold on page 1984
BUS<m>:MPHY:DONE:THReshold on page 1984
BUS<m>:MPHY:DTWO:THReshold on page 1984

BUS<m>:MPHY:DTHRee:THReshold on page 1984
BUS<m>:MPHY:THCoupling on page 1984

**Scrambled Mode**
Available, if "Variant" = "SSIC".

Selects, if the SSIC data is scrambled or descrambled. Set this parameter according to your signal parameters, to ensure a proper decoding.

Remote command:
BUS<m>:MPHY:SCRMode on page 1985

### 13.14.2.2 Display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

To enhance the decode possibilities of the M-PHY or UniPro protocol, you can use an additional setting in the "Display" tab: "Decode layer".

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the end results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

"Final" Final decoded frames of each lane.

"Edges" All edges of each lane in stage 1 decoding.

"Bits" All bits of each lane in stage 1 decoding.

"8b/10b Symbols"
8b/10b symbols in stage 2 decoding.

"LCC bits" LCC bits in stage 3 decoding.

"Filter/descrambler"
UniPro filter/descrambler in stage 3 decoding.

"Lane merge"
UniPro lane merge in stage 4 decoding.

"Bytes" UniPro bytes in stage 2 decoding.

**Show multiple lanes**
Enables the display of the results in multiple lanes.

Remote command:
BUS<m>:MPHY:SMLanes on page 1985

### 13.14.2.3 Configuring the M-PHY signals

Access: [Bus] > "Setup" tab > "Protocol" = *M-PHY*

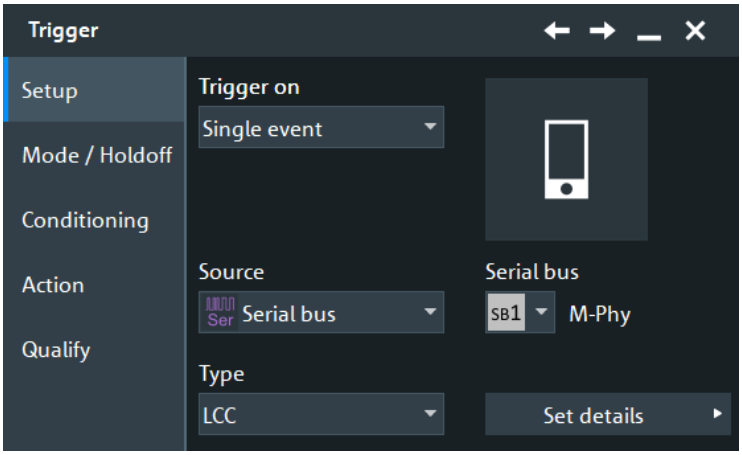
Assign the lanes to the channels and define the thresholds:

1. Press [Bus] on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, select the protocol: M-PHY.
4. Tap "Variant" and select the protocol.
5. Enter the number of "Lanes".
6. Select the source "Channel" for each lane.
7. Enter the "Thresholds" for each lane.
8. Enable "Decode".

## 13.14.3 M-PHY trigger

### 13.14.3.1 M-PHY trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"




Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Type**
Selects the trigger type for the analysis of the selected protocol.

Remote command:
TRIGger<m>:MPHY:TYPE on page 1987

**Start of frame ← Type**
Triggers on the start of an M-PHY or a UniPro frame.

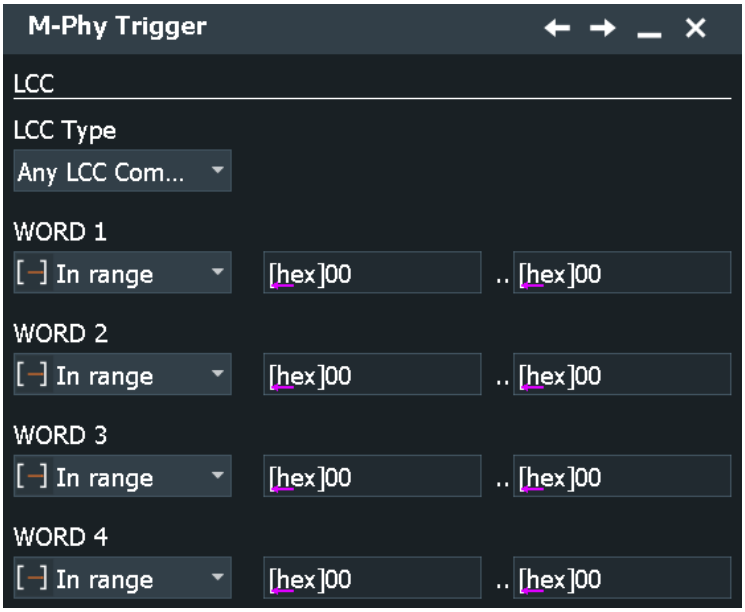
**Burst ← Type**
Triggers on an M-PHY burst frame.

**Adapt ← Type**
Triggers on an M-PHY Adapt frame. This control frame is used for clock/bit synchronization.

**LCC ← Type**
Triggers on an M-PHY Line Control Command (LCC) frame that matches the specified:

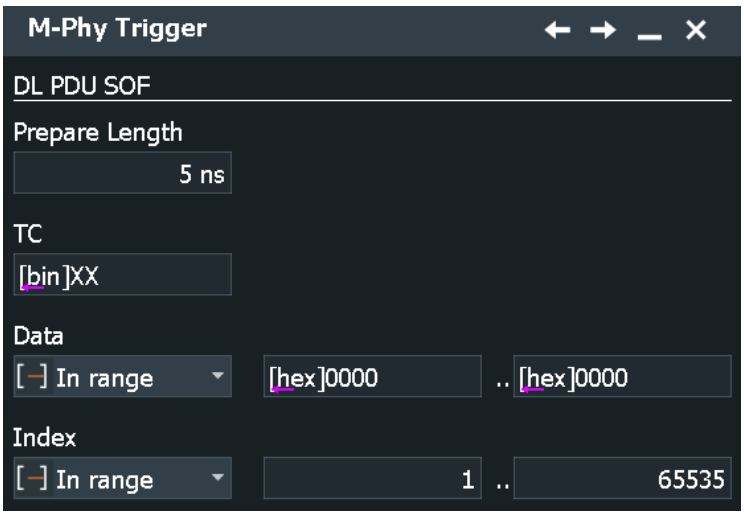
- LCC Type
- WORD

This control frame is used for configuring the line to a different state or mode depending on the LCC type.



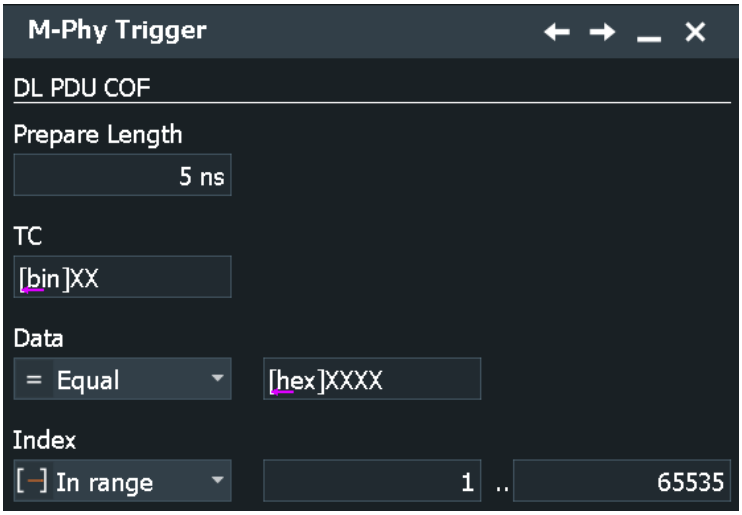

**DL PDU SOF ← Type**
Triggers on a UniPro Downlink Protocol Data Unit (DL PDU) Start Of Frame (SOF) that matches the specified:

- Prepare length
- TC
- Data
- Index



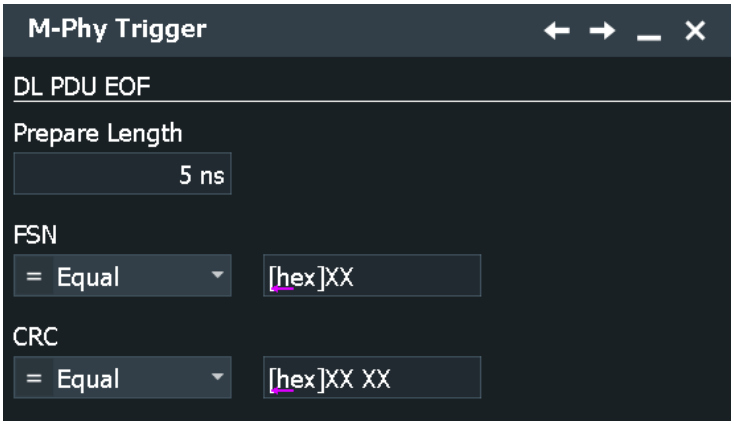

**DL PDU COF ← Type**
Triggers on a UniPro DL PDU Continuation Of Pre-empted Frame (COF) that matches the specified:

- Prepare length
- TC
- Data
- Index



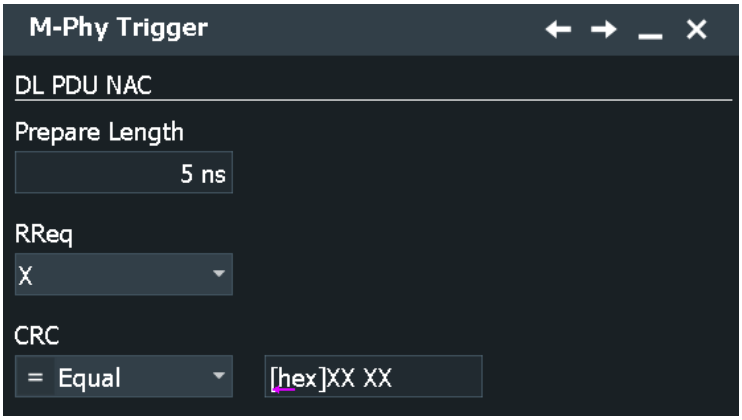

**DL PDU EOF ← Type**
Triggers on a UniPro DL PDU End Of Frame (EOF) frame that matches the specified conditions:

- Prepare length
- FSN
- CRC



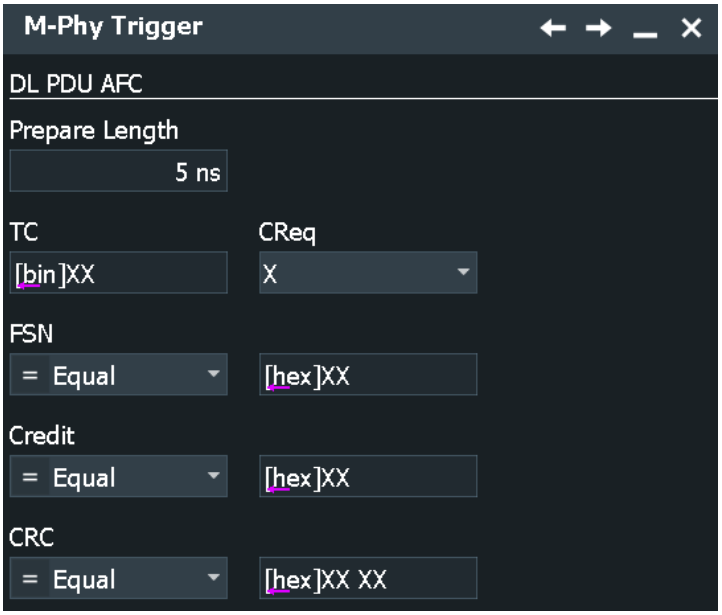

**DL PDU NAC ← Type**
Triggers on a UniPro DL PDU Negative Acknowledgment Control (NAC) frame that matches the specified conditions:

- Prepare length
- RReq
- CRC




**DL PDU AFC ← Type**
Triggers on a UniPro DL PDU Acknowledgement and L2 Flow Control (AFC) frame that matches the specified conditions:

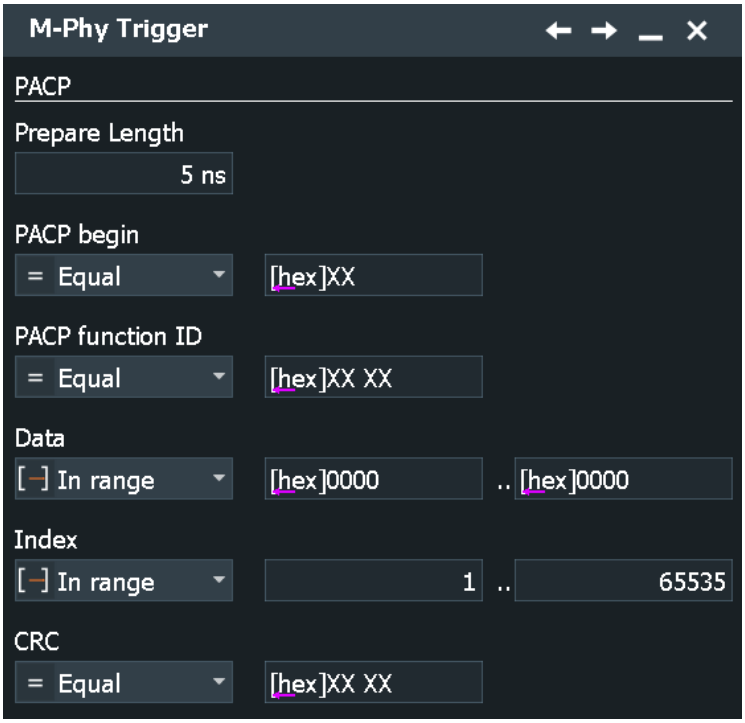
- Prepare length
- TC
- CReq
- FSN
- Credit
- CRC




**PACP ← Type**
Triggers on a UniPro PHY Adapter Control Protocol (PACP) frame that matches the specified condition:

- Prepare length
- PACP Begin
- PACP Function ID
- Data
- Index
- CRC

This control frame is used mainly for power mode change and L1.5 link management. It is also used in scrambling request.

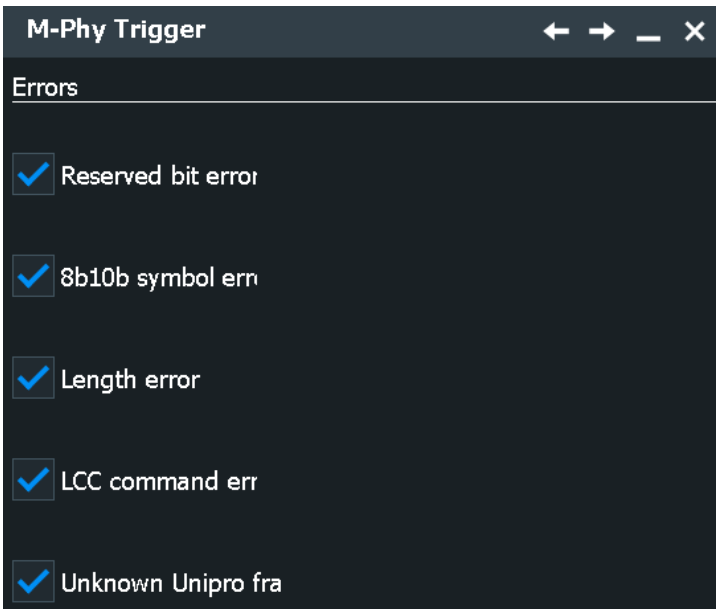



**Trigger Upr0 ← Type**
Triggers on a UniPro Trigger Upper0 frame. This control frame is used for link startup sequence.

**Trigger Upr1 ← Type**
Triggers on a UniPro Trigger Upper1 frame. This control frame is used for link startup sequence.

**Trigger Upr2 ← Type**
Triggers on a UniPro Trigger Upper2 frame. This control frame is used for link startup sequence.

**Errors ← Type**
Triggers on the specified M-PHY or UniPro error frame.




**Prepare length**
Sets the prepare length.

**LCC Type**
Selects the type of LCC command to be triggered on.

"Any LCC Command"
All the available LCC commands

"HIBERN8-SLEEP"
Switches the power-saving state to ultra-low power without configuration loss. This state saves up to 90% power and is used with PWM burst.

"HIBERN8-STALL"
Switches the power-saving state to ultra-low power without configuration loss. This state saves up to 75% power and is used with HS burst.

"READ-CAPABILITY"
Recovers data about the Optical Media Converter (OMC) capabilities

"READ-MFG-INFO"
Retrieves manufacturing ID and vendor-specific information

"READ-VEND-INFO"
Retrieves the additional four delimited bytes containing vendor-specific information

"WRITE-ATTRIBUTE"
Sets the configuration parameters required for lane operation

"PWM-G0/PWM-G1/PWM-G2/PWM-G3/PWM-G4/PWM-G5/PWM-G6/PWM-G7"
Switches the transmission mode to the selected low-power gear

"HS-G1A/HS-G2A/HS-G3A/HS-G4A/HS-G1B/HS-G2B/HS-G3B/HS-G4B"
Switches the transmission mode to the selected high-speed gear

"Reserved"
Reserved bit command. These bits are reserved for future use.

Remote command:
TRIGger<m>:MPHY:LCCType on page 1987

**WORD 1/WORD 2/WORD 3/WORD 4**
Sets the specified words to be triggered on. The setup for a word consists of the condition and one or two word patterns.

"Condition" Sets the comparison condition to a specific value or a range.

"Data Min/Data"
Specifies the value or sets the start value of a range.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
For " WORD 1":
TRIGger<m>:MPHY:LWONe:CONDition on page 1988
TRIGger<m>:MPHY:LWONe:MIN on page 1989
TRIGger<m>:MPHY:LWONe:MAX on page 1989
For " WORD 2":
TRIGger<m>:MPHY:LWTWo:CONDition on page 1989
TRIGger<m>:MPHY:LWTWo:MIN on page 1990
TRIGger<m>:MPHY:LWTWo:MAX on page 1990
For " WORD 3":
TRIGger<m>:MPHY:LWTHree:CONDition on page 1990
TRIGger<m>:MPHY:LWTHree:MIN on page 1991
TRIGger<m>:MPHY:LWTHree:MAX on page 1991
For " WORD 4":
TRIGger<m>:MPHY:LWFour:CONDition on page 1991
TRIGger<m>:MPHY:LWFour:MIN on page 1992
TRIGger<m>:MPHY:LWFour:MAX on page 1992

**TC**
Sets the specified Traffic Class (TC) to be triggered on.

TC is used for classifying the traffic flow based on protocol and port number, e.g. TC0 and TC1 are two priority classes with guaranteed link reliability defined and used in L2.

"Data" Defines the bit pattern of the TC pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

Remote command:
TRIGger<m>:MPHY:TC on page 1992

**Data**
Sets the data to be triggered on. The trigger data setup consists of a comparison condition and one or two data patterns.

"Condition" Sets the comparison condition to a specific value or a range.

"Data Min/Data"
Specifies the value or sets the start value of a range.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger<m>:MPHY:DATA:DCON` on page 1993
`TRIGger<m>:MPHY:DATA:DMIN` on page 1993
`TRIGger<m>:MPHY:DATA:DMAX` on page 1993

**Index**
The trigger index setup consists of a comparison condition and one or two address patterns.

"Condition" Sets the comparison condition to a specific value or a range.

"Index Min/Index"
Specifies the value or sets the start value of a range.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Index Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger<m>:MPHY:DATA:ICONdition` on page 1994
`TRIGger<m>:MPHY:DATA:IMIN` on page 1994
`TRIGger<m>:MPHY:DATA:IMAX` on page 1994

**FSN**
Sets the specified Frame Sequence Number (FSN) to be triggered on. The FSN setup consists of the condition and one or two FSN patterns.

"Condition" Defines the operator to set a specific FSN, e.g. "Equal" or "Not Equal" or a range.

"Data Min/Data"
Defines the bit pattern of the FSN pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second FSN pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
TRIGger<m>:MPHY:FSNumber:CONDition on page 1995
TRIGger<m>:MPHY:FSNumber:MIN on page 1995
TRIGger<m>:MPHY:FSNumber:MAX on page 1995

**CRC**
Sets the specified Cyclic Redundancy Check (CRC) to be triggered on. The CRC setup consists of the condition and one or two CRC patterns.

CRC is an error detecting code to detect accidental changes to raw data.

"Condition" Defines the operator to set a specific CRC, e.g. "Equal" or "Not Equal" or a range.

"Data Min/Data"
Defines the bit pattern of the CRC pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
TRIGger<m>:MPHY:CRC:CONDition on page 1996
TRIGger<m>:MPHY:CRC:MIN on page 1996
TRIGger<m>:MPHY:CRC:MAX on page 1996

**Req**
Sets the specified Reset Link Request (RReq) or Credit Transmit Request (CReq) to be triggered on.

RReq is used for requesting the remote end to reinitialize its Transmit (TX) Physical Layer (PHY) while CReq is used for requesting flow control information for the corresponding TC from the remote end.

"Data" Defines the bit pattern of the RReq pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

Remote command:
TRIGger<m>:MPHY:RREQ on page 1997
TRIGger<m>:MPHY:CREQ on page 1997

**Credit**
Sets the specified credit to be triggered on. The credit setup consists of the condition and one or two credit patterns.

Credit represents the total number of credits available since boot time.

"Condition" Defines the operator to set a specific credit, e.g. "Equal" or "Not Equal" or a range.

"Data Min/Data"
Defines the bit pattern of the credit pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second credit pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:MPHY:CREDit:CONDition` on page 1997
`TRIGger<m>:MPHY:CREDit:MIN` on page 1998
`TRIGger<m>:MPHY:CREDit:MAX` on page 1998

**PACP Begin**
Sets the specified PACP Begin to be triggered on. The PACP Begin setup consists of the condition and one or two PACP Begin patterns.

"Condition" Defines the operator to set a specific PACP Begin, e.g. "Equal" or "Not Equal" or a range.

"Data Min/Data"
Defines the bit pattern of the data value pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second PACP Begin pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:MPHY:PBEGin:CONDition` on page 1998
`TRIGger<m>:MPHY:PBEGin:MIN` on page 1999
`TRIGger<m>:MPHY:PBEGin:MAX` on page 1999

**PACP Function ID**
Sets the specified PACP Function ID to be triggered on. The PACP Function ID setup consists of the condition and one or two PACP Function ID patterns.

"Condition" Defines the operator to set a specific PACP Function ID, e.g. "Equal" or "Not Equal" or a range.

"Data Min/Data"
Defines the bit pattern of the PACP Function ID pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second PACP Function ID pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`TRIGger<m>:MPHY:PFID:CONDition` on page 1999
`TRIGger<m>:MPHY:PFID:MIN` on page 2000
`TRIGger<m>:MPHY:PFID:MAX` on page 2000

**Errors**
Sets the type of error events to be triggered on.

**Reserved Bit Error ← Errors**
Triggers on reserved bit error.

This error means that the waveform violates the reserved field according to the specification. For example, a DL_PDU_AFC frame requires that the two bits before and three bits after the FSN field to be reserved and set as 0. If the waveform carries 1 in these positions instead, it is marked as this error.

Remote command:
`TRIGger<m>:MPHY:ERRor:REServed` on page 2001

**LCC Command Error ← Errors**
Triggers on LCC command error.

This error is marked when the Command field in the LCC packet is not a known command defined in the specification.

Remote command:
`TRIGger<m>:MPHY:ERRor:LCMD` on page 2001

**8b10b Symbol Error ← Errors**
Triggers on 8b10b symbol error.

For example, some of the 10-bit combinations that do not map to 8-bit according to the specification are invalid 10-bit and therefore marked as this error.

Remote command:
`TRIGger<m>:MPHY:ERRor:SYMBol` on page 2001

**Unknown Unipro Frame ← Errors**
Triggers on unidentified UniPro frame error. This error is marked when the end mark on UniPro packets is undetected.

Remote command:
`TRIGger<m>:MPHY:ERRor:UNKNown` on page 2002

**Length Error ← Errors**
Triggers on frames that violate the length according to the specification.

Remote command:
`TRIGger<m>:MPHY:ERRor:LENGth` on page 2001

#### 13.14.3.2 Triggering on M-PHY

Prerequisite: A bus is configured for the M-PHY signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and select the "Trigger" tab.
2. Press the "Select" button. The "Source" button is shown.
3. Tap the "Source" button and select the "Serial bus" trigger source.

4. Tap "Serial bus" and select the serial bus that is set to M-PHY, e.g.:

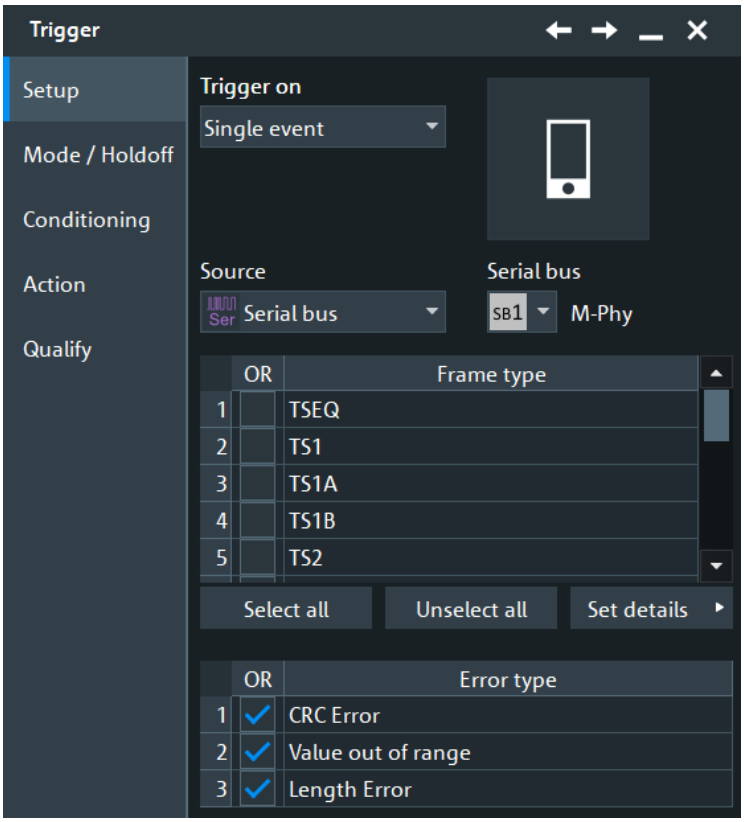
   The "Protocol" selection is then automatically set to "M-PHY".

5. Select the "Type" to use in the M-PHY protocol analysis.

6. To refine the trigger settings, configure additional settings, which are available for some trigger types.
   For details, see Chapter 13.14.3.1, "M-PHY trigger settings", on page 723.

### 13.14.3.3 SSIC trigger settings

The setting is available for SSIC option R&S RTP-K64.



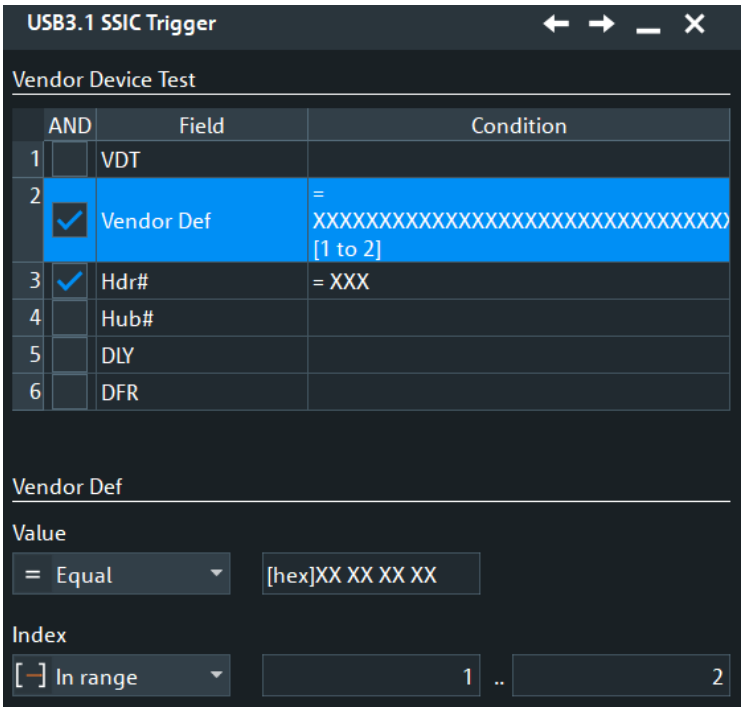

**Frame type**
Selects the frame type for the M-PHY SSIC.

The SSIC trigger frames and fields are the same as for USB3.1, see "Frame types" on page 837.

**Error types, Enable**
The table lists the error types you can trigger on. Select the required errors in the "OR" column.

Remote command:
TRIGger<m>:MPHY:SSIC:ERENable on page 2032
TRIGger<m>:MPHY:SSIC:ERRor<n>:ENABle on page 2032

**Set details**
Opens a dialog to define frame specific settings.




**AND / Field name / Condition ← Set details**
This table lists the field numbers and names in the selected frame together with a summary of the user settings of checking conditions for each field and a checkbox to enable the checking.

Select a field in the table to specify the checking conditions for this field in the "Set details" dialog.

The SSIC trigger frames and fields are the same as for USB3.1, see "Frame types" on page 837.

Remote command:
TRIGger<m>:MPHY:SSIC:FRENable on page 2032
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:ENABle on page 2033

**Data**
Defines for the selected field, how a data check is executed.

"Condition" Defining specific data or a data range requires to set the operator to one of the following conditions: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies data or sets the start value of a data range.

"Max" Sets the end value of a data range, if "Condition" is set to "In range" or "Out of range".

Remote command:
TRIGger<m>:MPHY:SSIC:DOPerator on page 2034
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DOPerator on page 2034
TRIGger<m>:MPHY:SSIC:DMIN on page 2034
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DMIN on page 2034
TRIGger<m>:MPHY:SSIC:DMAX on page 2033
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DMAX on page 2033

**BitState**
Defines the bit state to be checked for the selected field. Permitted bit states are "1", "0" or "X" (do not care).

Remote command:
TRIGger<m>:MPHY:SSIC:BIT on page 2033
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:BIT on page 2033

**Index**
Defines for the selected field, how an index check is executed.

| | |
|---|---|
| "Condition" | Defining a specific index or an index range requires to set the operator to one of the following conditions: equal, in range. |
| "Min" | Specifies the index or sets the start value of an index range. |
| "Max" | Sets the end value of an index range, if "Condition" is set to "In range". |

Remote command:
TRIGger<m>:MPHY:SSIC:IOPerator on page 2036
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IOPerator on page 2036
TRIGger<m>:MPHY:SSIC:IMIN on page 2035
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IMIN on page 2035
TRIGger<m>:MPHY:SSIC:IMAX on page 2035
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IMAX on page 2035

#### 13.14.3.4 Triggering on SSIC

Prerequisite: A bus is configured for the "M-PHY" > "Variant" = "SSIC" signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Tap "Serial bus" and select the serial bus that is set to SSIC.
5. If you trigger on errors, select the error types you want to find.
6. Select the frame type to be used for SSIC protocol analysis.
7. For some frame types, you can define the frame fields.

For details, see Chapter 13.14.3.3, "SSIC trigger settings", on page 735.

## 13.14.4 M-PHY decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".
2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode Results Table

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

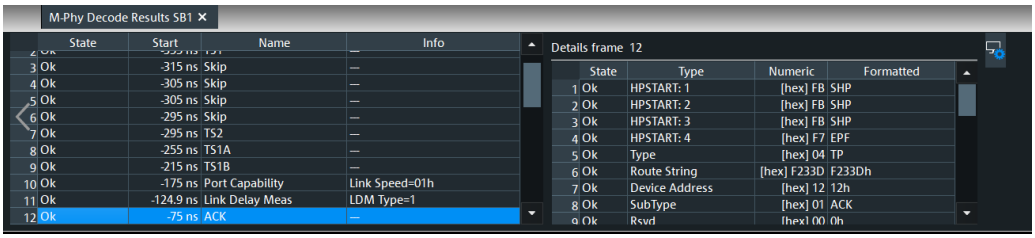



*Table 13-20: Content of the Decode results table*

| Column | Description |
|---|---|
| Frame | Frame count |
| State | Overall state of the frame indicating, for example, if the frame is valid or invalid |
| Start | Time of frame start in relation to the trigger point |
| Info | Label on top of the frame |

*Table 13-21: Content of the Details frame table*

| Column | Description |
|---|---|
| State | State of the field |
| Type | Frame identifier specifying the data or control frame name |

| Column | Description |
|---|---|
| Numeric | Hexadecimal value of the field |
| Formatted | Formatted content of the field |

**Tools**

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

**Remote commands**

Remote commands are described in Chapter 24.17.15.3, "Decode results M-PHY", on page 2002.

## 13.14.5 Search on decoded M-PHY data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

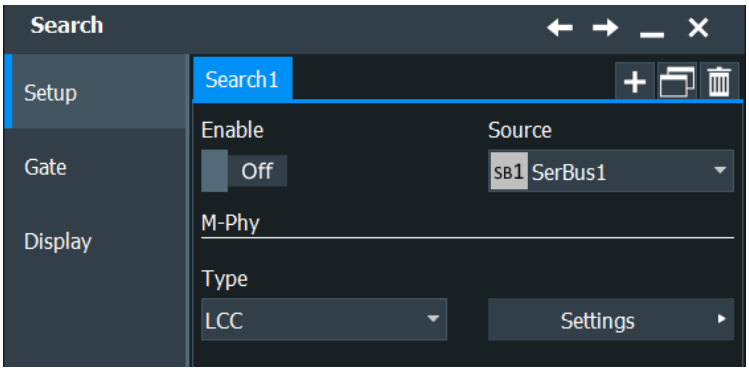
Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search source to "SerBus" for the configured protocol.

For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

### 13.14.5.1 M-PHY search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"




**Type**
Selects the search type for the selected protocol.

Remote command:
SEARch:TRIGger:MPHY:TYPE on page 2010

**Start of frame ← Type**
Searches for the start of an M-PHY or a UniPro frame.

**Burst ← Type**
Searches for an M-PHY burst.

**Adapt ← Type**
Searches for an M-PHY Adapt frame. This control frame is used for clock/bit synchronization.

**LCC ← Type**
Searches for an M-PHY Line Control Command (LCC) frame that matches the specified LCC Type or any WORD condition.

This control frame is used for configuring the line to a different state or mode depending on the LCC type.

**DL PDU SOF ← Type**
Searches for a UniPro Downlink Protocol Data Unit (DL PDU) Start Of Frame (SOF) that matches the specified TC, Data or Index condition.

**DL PDU COF ← Type**
Searches for a UniPro DL PDU Continuation Of Pre-empted Frame (COF) that matches the specified TC, Data or Index condition.

**DL PDU EOF ← Type**
Searches for a UniPro DL PDU End Of Frame (EOF) that matches the specified FSN or CRC condition.

**DL PDU NAC ← Type**
Searches for a UniPro DL PDU Negative Acknowledgment Control (NAC) frame that matches the specified RReq or CRC condition.

**DL PDU AFC ← Type**
Searches for a UniPro DL PDU Acknowledgement and L2 Flow Control (AFC) frame that matches the specified TC, CReq, FSN, Credit or CRC condition.

**PACP ← Type**
Searches for a UniPro PHY Adapter Control Protocol (PACP) frame that matches the specified PACP Begin, PACP Function ID, Data, Index or CRC condition.

This control frame is used mainly for power mode change and L1.5 link management. It is also used in scrambling request.

**Trigger Upper0 ← Type**
Searches for a Trigger Upper0 frame. This control frame is used for link startup sequence.

**Trigger Upper1 ← Type**
Searches for a Trigger Upper1 frame. This control frame is used for link startup sequence.

**Trigger Upper2 ← Type**
Searches for a Trigger Upper2 frame. This control frame is used for link startup sequence.

**Errors ← Type**
Searches for the specified error frame.

**LCC Type**
Selects the type of LCC command to be searched for.

"Any LCC Command"
All the available LCC commands

"HIBERN8-SLEEP"
Switches the power-saving state to ultra-low power without configuration loss. This state saves up to 90% power and is used with PWM burst.

"HIBERN8-STALL"
Switches the power-saving state to ultra-low power without configuration loss. This state saves up to 75% power and is used with HS burst.

"READ-CAPABILITY"
Recovers data about the Optical Media Converter (OMC) capabilities

"READ-MFG-INFO"
Retrieves manufacturing ID and vendor-specific information

"READ-VEND-INFO"
Retrieves the additional four delimited bytes containing vendor-specific information

"WRITE-ATTRIBUTE"
Sets the configuration parameters required for lane operation

"PWM-G0/PWM-G1/PWM-G2/PWM-G3/PWM-G4/PWM-G5/PWM-G6/PWM-G7"
Switches the transmission mode to the selected low-power gear

"HS-G1A/HS-G2A/HS-G3A/HS-G4A/HS-G1B/HS-G2B/HS-G3B/HS-G4B"
Switches the transmission mode to the selected high-speed gear

"Reserved"
Reserved bit command. These bits are reserved for future use.

Remote command:
`SEARch:TRIGger:MPHY:LCCType` on page 2010

**WORD 1/WORD 2/WORD 3/WORD 4**
Sets the specified words to be searched for. The setup for a word consists of the condition and one or two word patterns.

"Condition" Defines the operator to set a specific word, e.g. "Equal" or " Not Equal" or a range.

"Data Min/Data"
Defines the bit pattern of the word pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second word pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
For " WORD 1":
`SEARch:TRIGger:MPHY:LWONe:CONDition` on page 2011
`SEARch:TRIGger:MPHY:LWONe:MIN` on page 2012
`SEARch:TRIGger:MPHY:LWONe:MAX` on page 2012
For " WORD 2":
`SEARch:TRIGger:MPHY:LWTWo:CONDition` on page 2012
`SEARch:TRIGger:MPHY:LWTWo:MIN` on page 2013
`SEARch:TRIGger:MPHY:LWTWo:MAX` on page 2013
For " WORD 3":
`SEARch:TRIGger:MPHY:LWTHree:CONDition` on page 2014
`SEARch:TRIGger:MPHY:LWTHree:MIN` on page 2014
`SEARch:TRIGger:MPHY:LWTHree:MAX` on page 2014
For " WORD 4":
`SEARch:TRIGger:MPHY:LWFour:CONDition` on page 2015
`SEARch:TRIGger:MPHY:LWFour:MIN` on page 2015
`SEARch:TRIGger:MPHY:LWFour:MAX` on page 2015

**TC**
Sets the specified Traffic Class (TC) to be searched for.

TC is used for classifying the traffic flow based on protocol and port number, e.g. TC0 and TC1 are two priority classes with guaranteed link reliability defined and used in L2.

"Data" Defines the bit pattern of the TC pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

Remote command:
`SEARch:TRIGger:MPHY:TC` on page 2016

**Data**
Sets the specified data to be searched for. The data setup consists of the condition and one or two data patterns.

"Condition" Defines the operator to set a specific data, e.g. "Equal" or "Not Equal") or a range.

"Data Min/Data"
Defines the bit pattern of the data pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
SEARch:TRIGger:MPHY:DATA:DCONdition on page 2016
SEARch:TRIGger:MPHY:DATA:DMIN on page 2016
SEARch:TRIGger:MPHY:DATA:DMAX on page 2017

**Index**
Index sets the range within this series of data that is considered for the search. The index setup consists of the condition and one or two data index values.

"Condition" Defines the operator to set a specific index, e.g. "Equal" or a range.

"Index Min/Index"
Defines the bit pattern of the index pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Index Max" The second index pattern is required to specify a range with conditions "In range".

Remote command:
SEARch:TRIGger:MPHY:DATA:ICONdition on page 2017
SEARch:TRIGger:MPHY:DATA:IMIN on page 2018
SEARch:TRIGger:MPHY:DATA:IMAX on page 2018

**FSN**
Sets the specified Frame Sequence Number (FSN) to be searched for. The FSN setup consists of the condition and one or two FSN patterns.

"Condition" Defines the operator to set a specific FSN, e.g. "Equal" or "Not Equal" or a range.

"Data Min/Data"
Defines the bit pattern of the FSN pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second FSN pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
SEARch:TRIGger:MPHY:FSNumber:CONDition on page 2018
SEARch:TRIGger:MPHY:FSNumber:MIN on page 2019
SEARch:TRIGger:MPHY:FSNumber:MAX on page 2019

**CRC**
Sets the specified Cyclic Redundancy Check (CRC) to be searched for. The CRC setup consists of the condition and one or two CRC patterns.

CRC is an error detecting code to detect accidental changes to raw data.

"Condition" Defines the operator to set a specific CRC, e.g. "Equal" or "Not Equal" or a range.

"Data Min/Data"
Defines the bit pattern of the CRC pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second CRC pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
`SEARch:TRIGger:MPHY:CRC:CONDition` on page 2019
`SEARch:TRIGger:MPHY:CRC:MIN` on page 2020
`SEARch:TRIGger:MPHY:CRC:MAX` on page 2020

**Req**
Sets the specified Reset Link Request (RReq) or Credit Transmit Request (CReq) to be searched for.

RReq is used for requesting the remote end to reinitialize its Transmit (TX) Physical Layer (PHY) while CReq is used for requesting flow control information for the corresponding `TC` from the remote end.

"Data" Defines the bit pattern of the RReq pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

Remote command:
For "RReq":
`SEARch:TRIGger:MPHY:RREQ` on page 2021
For "CReq"
`SEARch:TRIGger:MPHY:CREQ` on page 2020

**Credit**
Sets the specified credit to be searched for. The credit setup consists of the condition and one or two credit patterns.

Credit represents the total number of credits available since boot time.

"Condition" Defines the operator to set a specific credit, e.g. "Equal" or "Not Equal" or a range.

"Data Min/Data"
Defines the bit pattern of the credit pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second credit pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
SEARch:TRIGger:MPHY:CREDit:CONDition on page 2021
SEARch:TRIGger:MPHY:CREDit:MIN on page 2022
SEARch:TRIGger:MPHY:CREDit:MAX on page 2022

**PACP Begin**
Sets the specified PACP Begin to be searched for. The PACP Begin setup consists of the condition and one or two PACP Begin patterns.

"Condition" Defines the operator to set a specific PACP Begin, e.g. "Equal" or "Not Equal" or a range.

"Data Min/Data"
Defines the bit pattern of the data value pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second PACP Begin pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
SEARch:TRIGger:MPHY:PBEGin:CONDition on page 2022
SEARch:TRIGger:MPHY:PBEGin:MIN on page 2023
SEARch:TRIGger:MPHY:PBEGin:MAX on page 2023

**PACP Function ID**
Sets the specified PACP Function ID to be searched for. The PACP Function ID setup consists of the condition and one or two PACP Function ID patterns.

"Condition" Defines the operator to set a specific PACP Function ID, e.g. "Equal" or "Not Equal" or a range.

"Data Min/Data"
Defines the bit pattern of the PACP Function ID pattern.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second PACP Function ID pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
SEARch:TRIGger:MPHY:PFID:CONDition on page 2023
SEARch:TRIGger:MPHY:PFID:MIN on page 2024
SEARch:TRIGger:MPHY:PFID:MAX on page 2024

**Errors**
Sets the type of error events to be searched for.

**Reserved Bit Error ← Errors**
Searches for reserved bit error.

This error means that the waveform violates the reserved field according to the specification. For example, a DL_PDU_AFC frame requires that the two bits before and three bits after the FSN field to be reserved and set as 0. If the waveform carries 1 in these positions instead, it is marked as this error.

Remote command:
`SEARch:TRIGger:MPHY:ERRor:REServed` on page 2024

**LCC Command Error ← Errors**
Searches for LCC command error.

This error is marked when the Command field in the LCC packet is not a known command defined in the specification.

Remote command:
`SEARch:TRIGger:MPHY:ERRor:LCMD` on page 2025

**8b10b Symbol Error ← Errors**
Searches for 8b10b symbol error.

For example, some of the 10-bit combinations that do not map to 8-bit according to the specification are invalid 10-bit and therefore marked as this error.

Remote command:
`SEARch:TRIGger:MPHY:ERRor:SYMBol` on page 2025

**Unknown Unipro Frame ← Errors**
Searches for unidentified UniPro frame error. This error is marked when the end mark on Unipro packets is undetected.

Remote command:
`SEARch:TRIGger:MPHY:ERRor:UNKNown` on page 2025

**Length Error ← Errors**
Searches for frames that violate the length according to the specification.

Remote command:
`SEARch:TRIGger:MPHY:ERRor:LENGth` on page 2025

### 13.14.5.2 M-PHY search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands:

- `SEARch:RESult:MPHY:FCOunt?` on page 2026
- `SEARch:RESult:MPHY:FRAMe<m>:CCOunt?` on page 2026

- SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:DATA? on page 2027
- SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:TYPE? on page 2027
- SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:STARt? on page 2027
- SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:STOP? on page 2028
- SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:STATe? on page 2028
- SEARch:RESult:MPHY:FRAMe<m>:DATA? on page 2028
- SEARch:RESult:MPHY:FRAMe<m>:FTYPe? on page 2029
- SEARch:RESult:MPHY:FRAMe<m>:STARt? on page 2030
- SEARch:RESult:MPHY:FRAMe<m>:STATe? on page 2029
- SEARch:RESult:MPHY:FRAMe<m>:STOP? on page 2031

#### 13.14.5.3 Searching M-PHY

Prerequisite: A serial bus is configured for the M-PHY signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".
3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Specify the search criteria according to Chapter 13.14.5.1, "M-PHY search setup", on page 739.
5. To acquire a waveform, press [Single].

   The R&S RTP performs an M-PHY decode according to the thresholds and protocol settings of the associated serial bus source.
6. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog:

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

#### 13.14.5.4 SSIC search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"

**Search criteria**
Use the "Search criteria" dialog to define the search type. If you search for frame content, define the frame type in which data is to be searched.

You can define individual search parameters for the fields of the selected frame in the table below the "Search criteria". To specify these parameters, select a field in the table and define the data and/or index operators and values, or the bit state.

For a description of how to set the search conditions, see Chapter 13.14.3.3, "SSIC trigger settings", on page 735.

Remote command:
SEARch:TRIGger:MPHY:SSIC:ERRor<m>:ENABle on page 2037
SEARch:TRIGger:MPHY:SSIC:FIENable on page 2038
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:ENABle on page 2037
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:BIT on page 2039
SEARch:TRIGger:MPHY:SSIC:BIT on page 2039
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DMAX on page 2039
SEARch:TRIGger:MPHY:SSIC:DMAX on page 2039
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DMIN on page 2040
SEARch:TRIGger:MPHY:SSIC:DMIN on page 2040
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DOPerator on page 2040
SEARch:TRIGger:MPHY:SSIC:DOPerator on page 2040
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:ENABle on page 2038
SEARch:TRIGger:MPHY:SSIC:FRENable on page 2037
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IMAX on page 2041
SEARch:TRIGger:MPHY:SSIC:IMAX on page 2041
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IMIN on page 2042
SEARch:TRIGger:MPHY:SSIC:IMIN on page 2042
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IOPerator on page 2042
SEARch:TRIGger:MPHY:SSIC:IOPerator on page 2042

## 13.15 Custom: Manchester / NRZ (option R&S RTP-K50)

R&S RTP-K50 is a firmware option that enables the R&S RTP to analyze customizable serial bus signals encoded by the following coding standards:

- Manchester
- Manchester II
- NRZ Clocked
- NRZ Unclocked

For analysis, signals encoded in any of these protocols can be triggered and decoded.

Due to the free format description, no search within the decoded events is available.

This chapter describes:

- Custom: Manchester / NRZ protocols ..... 749
- Custom: Manchester / NRZ configuration ..... 750
- Custom: Manchester / NRZ trigger ..... 770
- Custom filter ..... 775
- Custom: Manchester / NRZ decode results ..... 778
- Search on decoded custom Manchester / NRZ data ..... 779

## 13.15.1 Custom: Manchester / NRZ protocols

"Manchester" coding is a self-clocked coding scheme also known as phase-shift keying (or phase encoding, PE). It is used in protocols such as ProfiBus (IEC 61158), DALI (Digital Addressable Lighting Interface, IEC 60929 and IEC 62386), MVB (Multifunction Vehicle Bus, part of IEC 61375 for Train Communication Networks, TCN), and Ethernet 10BASE-T (10 Mbit/s, IEEE 802.3i). In terms of a logical Boolean operation, the Manchester value of each bit (as per G. E. Thomas) is the exclusive disjunction (XOR) of the original data value and the clock value. A "0" is expressed by a high-to-low transition, a "1" by a low-to-high transition. These transitions, which occur at the middle of each bit period, make the signal self-clocked.

"Manchester II" coding (as per IEEE 802.3) is represented by inverted Manchester values: a "0" is expressed by a low-to-high transition, a "1" by a high-to-low transition.

NRZ stands for "non-return-to-zero" coding: Typically a "1" is represented by a positive voltage and a "0" is represented by a negative voltage, with no "zero" voltage state. NRZ code requires only half the bandwidth of Manchester code, and it can either be clocked or unclocked. NRZ unclocked signals require a user-defined bit rate and gap time setting for triggering and decoding.

### 13.15.1.1 Special features of Manchester coding

In practical protocols, Manchester coding appears in many variations, often employing deliberate coding violations to encode special waveform features, such as unambiguous synchronization and termination patterns. To adapt to these specific Manchester implementations and handle ambiguous signals, the option R&S RTP-K50 for Custom Serial Bus uses a combination of automatic algorithms and user configurable parameters.

#### Quaternary Symbols

The software supports not just traditional binary symbols "0" and "1", but also arbitrary violation waveforms that use two additional symbols, yielding a total of four valid "quaternary bit" values. The two additional violation symbols are "H" (high) and "L" (low). Values of "H" correspond to a waveform lacking a transition in the center of the bit, with a physical high voltage state. Similarly, "L" violations also lack a center transition, but have a physical low voltage state. Most Manchester synchronization and termination conventions, even those containing violations, may be expressed as sequences of these four symbols. R&S RTP-K50 uses the quaternary notation to support Manchester patterns in the honeycomb display and to describe synchronization and termination patterns in the frame description table.

#### Idle Conditions

The state of the signal line in between messages is the idle condition. Manchester appears in practical standards with varying idle conditions: it can idle at the high, low, or middle voltage state. High and low idle states correspond to "biphase" Manchester, while the middle voltage (often ground) adds a third state to become "ternary" Manchester. Using ternary Manchester, option R&S RTP-K50 can usually establish the gaps between messages automatically. Using binary Manchester, the software has no

way to automatically discriminate an idling bus from monotonic sequences of "H" or "L" violations. For these biphase situations, R&S RTP-K50 offers a "Gap Time" detection feature, which allows to distinguish long intervals of non-transitions between bus idling and sequences of violations. Other differences between biphase and ternary Manchester are managed automatically by the software, with no user input required.

**Edge Conventions**

Most Manchester encodings establish the beginning of the first bit by a first transition, hence an "overhead" edge. The center of the bit is then marked by a second transition, which is a "sampling" edge. Some Manchester implementations, however, sample the first bit on the first edge. The option R&S RTP-K50 attempts to automatically detect this situation. Unfortunately, it is possible to trick the algorithm with waveforms that contain many (legitimate) violations. In these situations, the user can force a "First Edge" or "Second Edge" convention for handling edges. Edge sampling according to the "First Edge" convention is more likely to appear in biphase Manchester, but the software also supports this setting for ternary Manchester situations.

**Bit Rate**

Typically, a single bit rate is clearly specified in Manchester protocols; however, some implementations use a variable bit rate. By default, R&S RTP-K50 automatically determines the bit rate with no user input required. However, there are fundamental ambiguities possible in Manchester, if the bitrate is unknown. In particular, sequences like "0000", "1111", "0101", "1010", and many situations involving "H" and "L" violations, cannot be decoded without a known bit rate. The situation becomes even less defined with eventual Manchester coding violations. In these situations, a fixed "Bit Rate" setting has to be provided by the user to bypass the software's estimation algorithm.

## 13.15.2 Custom: Manchester / NRZ configuration

If you need information on how to get started with configuring the custom serial bus setup, see Chapter 13.15.2.7, "Configuring custom Manchester / NRZ signals", on page 769. Otherwise proceed with the configuration settings.

### 13.15.2.1 Custom: Manchester / NRZ configuration settings

Access: [Bus] > "Custom" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.

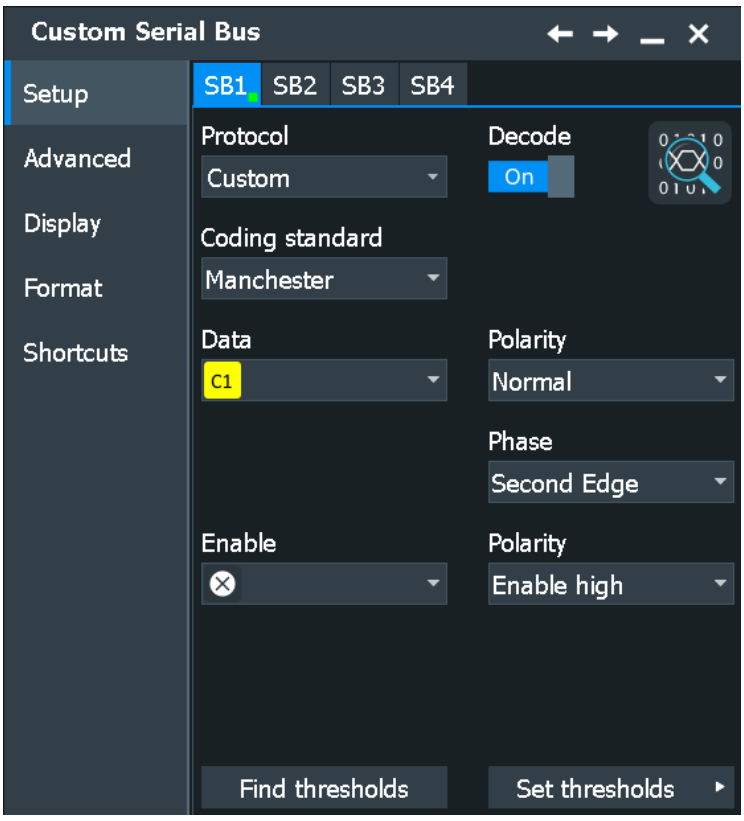



*Figure 13-18: Coding standard selection in the serial bus protocol configuration dialog*

See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Coding Standard**
To define the coding of the custom serial bus to be analyzed, select one of the following standards:

"Manchester" Selects the coding standard Manchester.
Optional "Timing settings" are "Bit Rate" (default: disabled, 10 Mbit/s) and "Gap time" (default: disabled, 10 µs).

"Manchester II" Selects the coding standard Manchester II, which is the inverted signal of the coding standard Manchester.
Optional "Timing settings" are "Bit Rate" (default: disabled, 10 Mbit/s) and "Gap time" (default: disabled, 10 µs).

"NRZ Clocked" Selects the coding standard NRZ Clocked.

"NRZ Unclocked" Selects the coding standard NRZ unclocked.

Remote command:
`BUS<m>:CMSB:CODing` on page 2044

**Data Source**
Defines the input source for the custom serial bus data signal.

The data source for Manchester and NRZ coding standards is selected separately, independent of each other. The data source is set to default upon switching the coding standard.

Permitted source selections are:

- For "Manchester"/ "Manchester II":
  - Decoding: the analog, mathematical, and reference channels
  - Triggering: the analog channels
- For "NRZ Clocked":
  - Decoding: the analog, mathematical, reference and digital channels
    Digital channels can be only used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.
  - Triggering: the analog and digital channels
- For "NRZ Unclocked":
  - Decoding: the analog, mathematical, reference and digital channels
    Digital channels can be only used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.
  - Triggering: the analog channels

Remote command:
`BUS<m>:CMSB:MANChester:DATA` on page 2045
`BUS<m>:CMSB:NRZ:DATA` on page 2048

**Clock Source**
Defines the source for the custom serial bus clock signal (only available for the coding standard "NRZ Clocked").

Permitted source selections are the analog, mathematical, reference and digital channels.

Digital channels can be only used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.

For triggering on the serial bus, when the NRZ clocked coding standard is selected, analog or digital channel sources are required.

Remote command:
`BUS<m>:CMSB:NRZ:CLCK` on page 2047

**Data Polarity (Manchester)**
Defines the polarity of the custom serial bus data signal in Manchester coding standards. The available settings are "Normal" or "Inverted".

Remote command:
`BUS<m>:CMSB:MANChester:POLarity` on page 2045

**Data Polarity (NRZ)**
Defines the polarity of the custom serial bus data signal in NRZ coding standards. The available settings are:

"Active high" The value "1" is represented by a voltage above the threshold.

"Active low" The value "1" is represented by a voltage below the threshold.

Remote command:
`BUS<m>:CMSB:NRZ:POLarity` on page 2050

**Data Idle Polarity (NRZ Unclocked)**
Defines the idle polarity of the custom serial bus data signal (only available for the coding standard "NRZ Unclocked"). The available settings are:

"Idle low" The base value of the clock is "0"; after an idle period, the data signal starts with a low-to-high transition.

"Idle high" The base value of the clock is "1"; after an idle period, the data signal starts with a high-to-low transition.

Remote command:
`BUS<m>:CMSB:NRZ:IDLPolarity` on page 2048

**Clock Polarity (NRZ Clocked)**
Defines the polarity of the custom serial bus clock signal (only available for the coding standard "NRZ Clocked"). The available settings are:

"Idle low" The base value of the clock is "0".

"Idle high" The base value of the clock is "1".

Remote command:
`BUS<m>:CMSB:NRZ:CPOLarity` on page 2049

**Clock Phase (Manchester)**
Defines the phase of the custom serial bus clock signal for the Manchester coding standards. The available settings are:

"Auto Detect" Lets the decoder automatically select the method ("First Edge" or "Second Edge") for detecting the clock phase.

"First Edge"
- At "Idle" = "low": data are captured on the clock's rising edge (low-to-high transition) and propagated on a falling edge.
- At "Idle" = "high": data are captured on the clock's falling edge (high-to-low transition) and propagated on a rising edge.

"Second Edge"
- At "Idle" = "low": data are captured on the clock's falling edge (high-to-low transition) and propagated on a rising edge.
- At "Idle" = "high": data are captured on the clock's rising edge (low-to-high transition) and propagated on a falling edge.

**Note:** The requirement to specify "First Edge" or "Second Edge" (or let the decoder decide) has the following background: In Manchester coding, an edge is always a transition from high to low (0) or from low to high (1). However, if the signal comes from the idle state, this implies that right before the first valid edge, there is always an overhead transition from idle to high or from idle to low. Some standards may regard this as a valid transition. To avoid a potentially ambiguous situation, a decision has to be made if the first edge is indeed only some overhead transition - or a transition that needs to be sampled. For more details on edge conditions, see Chapter 13.15.1.1, "Special features of Manchester coding", on page 749.

Remote command:
`BUS<m>:CMSB:MANChester:CPHase` on page 2047

**Clock Phase (NRZ Clocked)**
Defines the phase of the custom serial bus clock signal for the coding standard "NRZ Clocked", depending on "Clock Polarity". The available settings are:

"First Edge"
- At "Idle" = "low": data are captured on the clock's rising edge (low-to-high transition) and propagated on a falling edge
- At "Idle" = "high": data are captured on the clock's falling edge (high-to-low transition) and propagated on a rising edge

"Second Edge"
- At "Idle" = "low": data are captured on the clock's falling edge (high-to-low transition) and propagated on a rising edge
- At "Idle" = "high": data are captured on the clock's rising edge (low-to-high transition) and propagated on a falling edge

Remote command:
`BUS<m>:CMSB:NRZ:CPHase` on page 2049

**Enable Source (NRZ)**
Defines the input source for the custom serial bus enable signal.

If an input is chosen, signals are only decoded when this channel is in the enabled state. This allows you to mark a time when the signal on the selected source is active and when not.

Permitted source selections are the analog, mathematical, and reference channels.

When the serial bus trigger has been selected, the only permitted source selections are the analog channels "C1" – "C4", which are required for triggering.

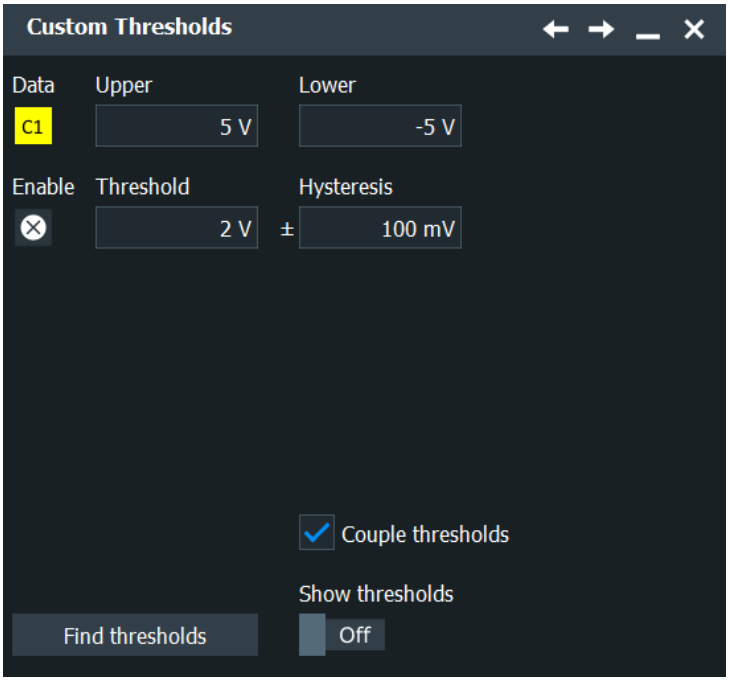
Math and Reference channels can only be selected, if no serial bus trigger is selected.

Remote command:
`BUS<m>:CMSB:NRZ:ENBLe` on page 2050

**Enable Polarity (NRZ)**
Selects whether the transmitted enable signal is active when the voltage is below the Set Thresholds ("Enable low") or higher than it ("Enable high").

Remote command:
`BUS<m>:CMSB:NRZ:ENAPolarity` on page 2050

**Set Thresholds**
Sets the threshold value for the digitization of each signal line and the hysteresis. If the signal voltage on the line is higher than the upper threshold, the signal state is high. Otherwise, if the signal voltage is below the lower threshold, the signal state is considered low.




- Manchester coding standards use 3-state signals with an upper and a lower voltage threshold in the range of -25 V to +25 V. A low-to-high transition requires the signal to exceed the upper threshold; a high-to-low transition requires the signal to fall below the lower threshold.
  You can set a hysteresis only for the enable signal.
- NRZ coding standards use a single voltage threshold for the data line. The value is in the range of -25 V to +25 V. It is entered into the middle of three available threshold input fields, or into the upper available threshold input field for NRZ Unclocked.
  You can set the hysteresis for all signal lines.
- In the NRZ Clocked coding standard, there is an additional clock voltage threshold available. This value in the range of -25 V to +25 V is entered into the upper threshold input field.
  You can set the hysteresis for the clock line.

There are four ways to set the threshold:

"Threshold" Directly sets the threshold values.

- For Manchester: upper threshold in the upper field, lower threshold in the lower field.
- For NRZ Clocked: clock threshold in the upper field, data threshold in the middle field and enable threshold in the lower field. Hysteresis values are in the right column.
- For NRZ Unclocked: data threshold in the upper field and enable threshold in the lower field. Hysteresis values are in the right column.

Remote command:
BUS<m>:CMSB:MANChester:THReshold:HIGH on page 2045
BUS<m>:CMSB:MANChester:THReshold:LOW on page 2046
BUS<m>:CMSB:NRZ:THReshold:CLCK on page 2051
BUS<m>:CMSB:NRZ:THReshold:DATA on page 2051
BUS<m>:CMSB:NRZ:THReshold:ENBLe on page 2052
BUS<m>:CMSB:NRZ:HYSTeresis:CLCK on page 2051
BUS<m>:CMSB:NRZ:HYSTeresis:DATA on page 2051
BUS<m>:CMSB:NRZ:HYSTeresis:ENBLe on page 2052

"Find thresholds" Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

Remote command:
BUS<m>:SETReflevels on page 1638

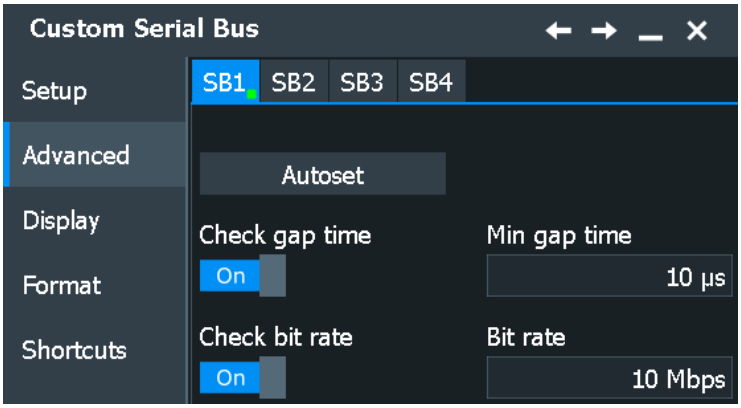
"Couple thresholds"

- For Manchester and Manchester II coding, the upper and lower thresholds are coupled to voltage values with the same magnitude but opposite sign (positive for the upper threshold and negative for the lower threshold). However, if the upper threshold is set to a negative voltage or the lower threshold is set to a positive voltage, coupling is disabled, and the other voltage (the one that was not actively set) is automatically adjusted, to avoid an upper threshold below the lower one, or a lower threshold above the upper one.
- For NRZ Clocked coding, the clock and data threshold values are coupled to the same voltage.

Remote command:
BUS<m>:CMSB:MANChester:THReshold:COUPling on page 2046
BUS<m>:CMSB:NRZ:THReshold:COUPling on page 2053

#### 13.15.2.2 Advanced

Access: [Bus] > "Advanced" tab.




**Autoset**
Starts software algorithms for determining the signal threshold levels and bitrate.

Remote command:
`BUS<m>:FAUToset` on page 1638

**Check bit rate**
Enables the bit rate setting. In most cases, bit rate is recovered from the signal automatically. You can check the value by enabling this setting.

This setting is not available for "NRZ Clocked".

Remote command:
`BUS<m>:CMSB:BITRate:ENABle` on page 2053

**Bit Rate**
Defines the transmission speed setting for the data signal. The bit rate is recovered from the signal automatically. However, the parameter can be used as a correction if the signal doesn't have single bit transitions, e.g. a compliance pattern that looks like "110011001100".

A bit rate definition is not available for "NRZ Clocked".

For more information on the bit rate, see Chapter 13.15.1.1, "Special features of Manchester coding", on page 749.

Remote command:
`BUS<m>:CMSB:BITRate:VALue` on page 2054

**Check gap time**
Enables the gap time settings (always enabled for the coding standard "NRZ Unclocked", and also for triggering on signals in any coding standard).

Remote command:
`BUS<m>:CMSB:GAPTime:ENABle` on page 2054

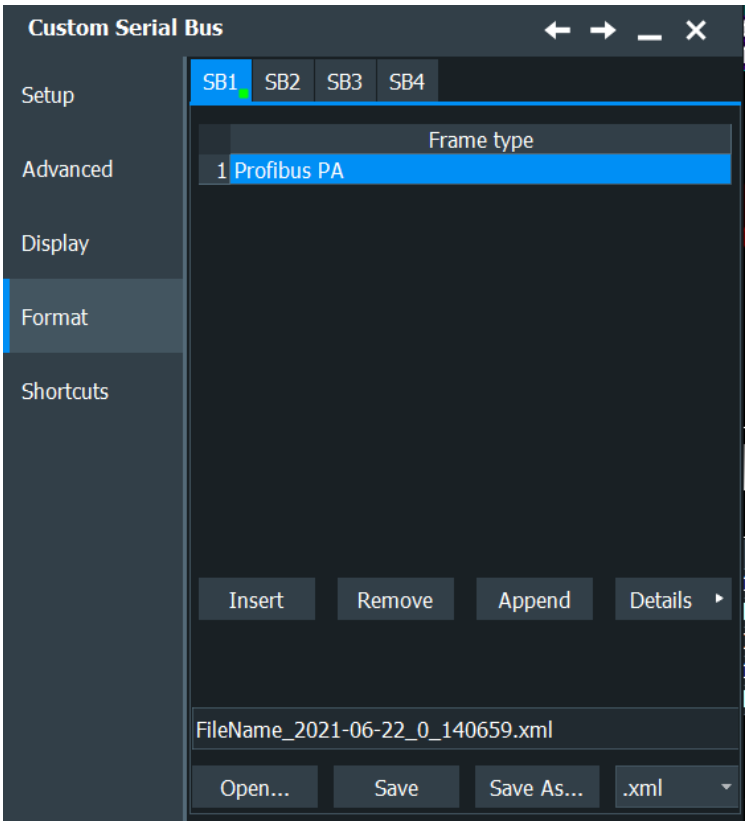
**Min gap time**
Specifies a minimum gap time (idle time or timeout) between two frames. A gap time definition is optional for the coding standards "Manchester", "Manchester II" and "NRZ Clocked", but obligatory for "NRZ Unclocked" (and also for triggering on signals in any coding standard). Default gap time is 10 µs, permitted gap times range from 1 ns to 1 s.

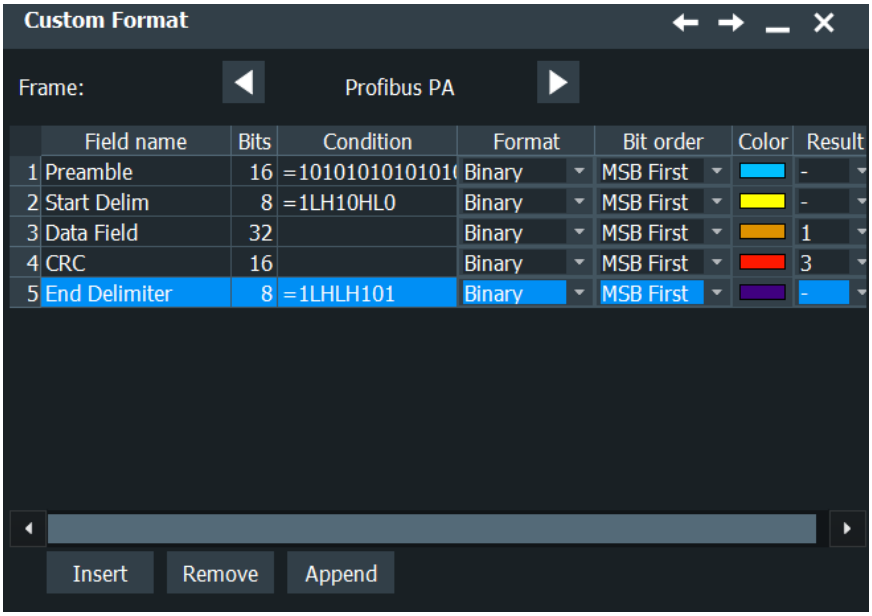
For more details on gap time and idle conditions, see Chapter 13.15.1.1, "Special features of Manchester coding", on page 749.

Remote command:
`BUS<m>:CMSB:GAPTime:VALue` on page 2054

### 13.15.2.3 Frame format configuration

This dialog enables you to describe the generic format and logical structure of typical protocols by creating customized frame descriptions of various structures and lengths.






**Frames**

A frame format description (or *frame description*, for short) is represented by one "page" in this dialog. It can be created by clicking on "Add". If one or several frame descriptions already exist, the new frame description is then created at the end of the frame format description list (or *frame list*, for short). Describing a frame format requires assigning it a name as well as creating Frame format configuration and specifying cell descriptions. The number of frame descriptions is limited to 100. The frame description that is on display can be deleted from the frame list by clicking "Remove".

The "Frame type" string is intended for the user to label the frame description (typically according to the specifications of the applicable protocol standard). For example, MDIO (Management Data Input/Output) specifies the frames "READ", "WRITE", "ADDRESS".

The frame format dialog also provides the features "Open", "Save", "Save As...", and "Explore...", to store created sets of frame descriptions into files (in ".xml" format), or load existing files of this kind.

The frame identification is executed top down, in the order in which the frame formats are described in the frame list. This provides a hierarchy of criteria for identifying frames.

In case it is required to change the order of previously created frame descriptions, it is recommended to save the set of frame descriptions, and then edit the XML file with any suitable editor. This also allows - with due care - additional editing features, if required.

If no user-defined frame description should be suitable to identify an incoming frame, per default such a "missed" frame is reproduced as "Undescribed Bits" in the honeycomb display. These bits are not shown in the results table.

**Note:** There may be frame descriptions that positively identifies each kind of frame, e.g. if no **equal** operator is defined for any of the fields. This "catches" every frame, even if there are other frame descriptions to follow in the frame list. Therefore, if a "catch all" frame description is used, it should be placed at the end of the frame list, or

it overwrites any subsequent frame description. However, instead of using a "catch all" frame description, the built-in "undescribed bits" display as mentioned above may be the better approach to create frame descriptions.

It is in the responsibility of the user to define unambiguous settings for each frame type. For a description of these conditions in XML file format and the required XML grammar and syntax, see Chapter 13.15.2.5, "XML syntax", on page 766.

Remote command:
BUS<m>:CMSB:FRAMe<n>:TYPE on page 2055
BUS<m>:CMSB:ADDFrame on page 2054
BUS<m>:CMSB:CLR on page 2055
BUS<m>:CMSB:FCOunt? on page 2055

**Field name**
A field description (which is represented by one row in one frame description) can be created at any position of a frame description by clicking on "Insert". This brings up a new field description in the active frame description, on top of the selected position. The "Append" button adds a cell description at the end of a frame description, below the lowest existing cell description. The number of Field descriptions is not limited. The "Remove" button deletes a selected field description from the active frame description.

The field result can be selected to be displayed in a specified result column of the decode table (see "Result" on page 765).

**Note:** The cell descriptions must be sequential and complete. No gaps are allowed, since the "Bis" are used to calculate the start position of the next cell.

A frame type is identified when all user-defined cell conditions are met, which can be regarded as related by the Boolean AND operator.

This can also locate a synchronization pattern, specified by the equal operator in the field. For example, if you define a "Preamble" cell with the condition *=FFFFFFFF*, the decoder scans the data for this pattern, and then synchronize to it.

The field in a frame are described by:

- Field Name
- "Bits" on page 761
- "Condition" on page 761
- "Format" on page 764
- "Color" on page 765
- "Result" on page 765

Remote command:
BUS<m>:CMSB:FRAMe<n>:CCOunt? on page 2056
BUS<m>:CMSB:FRAMe<n>:APPend on page 2055

**Field Name**
The strings in the column title describe cell names. They do not have to be unique; cell names are just for user support.

Remote command:
BUS<m>:CMSB:FRAMe<n>:CELL<o>:NAME on page 2056

**Bits**
This crucial information defines the length of the cell and - based on the previous cells - also the cell end position and the next cell start position within a frame.

If, for a given bit count, the Condition value is longer, it is truncated. If the condition value is shorter, it is padded with 0. Both truncation and padding occur at the left side of the condition value.

Examples:
- if "Condition" is "=111000" and the "Bit order" is "MSB", then
  - if "Bits" is 4, the truncated condition is "=1000"
  - if "Bits" is 8, the padded condition is "=00111000"
- if "Condition" is "=111000" and the bit order is "LSB" (accordingly, the condition in "MSB" format would be "=000111"), then
  - if "Bit s" is 4, the truncated condition is "=1000" for LSB and "=0001" for MSB
  - if "Bits" is 8, the padded condition is "=00111000" for LSB and "=00011100" for MSB

Remote command:
`BUS<m>:CMSB:FRAMe<n>:CELL<o>:BITCount` on page 2056

**Condition**
This text field is used to apply various conditions and functionalities for a cell. Among others, it can be used to identify mandatory values (such as CRC checksum or ID) that help to identify a frame. The numeric format and bit order of the condition value has to match up with what is defined in the fields "Format" and "Bit order".

The following conditions are implemented:

"= (equal)"

The **equal** operator (represented by the "=" sign) defines a pattern for the field to match. Valid condition entries are characters that match the cell's defined "Format", "Bit order", and "Bits". In binary format, for example, valid characters are "1", "0", "H" (high), and "L" (low).

**Three cases** have to be distinguished (cases A, B1, and B2), depending on the presence of a Variable Length Array: [] in the same frame description:

- **Case A**: If there is **no** "Variable Length Array" field, then each field marked with the equal operator acts as a key to identify a frame type. Only if all these fields match up with the expected value, the frame type is identified.
- **Cases B1 and B2**: If there **is** a "Variable Length Array" cell, then the equal operator has two different functionalities, depending on the position of the equal-operator cell within the frame description:
  - **B1**: If the cell is located *anywhere before* the "Variable Length Array" cell, the condition acts as a key to identify a frame type (as in case A).
  - **B2**: If the cell is located *immediately after* the "Variable Length Array" cell, the condition acts as an array delimiter.
    (Note: If the cell, which is marked with the equal operator, is located after the "Variable Length Array" cell, but *not* immediately after it, the decode result is unpredictable.)

Typically, Manchester protocols use code violations for synchronization. The states "H" and "L", supported by the equal operator in binary Format, mark that a transition is expected at this bit, but only a high or low signal is found.
Examples for the MVB protocol:
Master - Delimiter: "=*1LH0LH000*"
Slave - Delimiter: "=*0000LH0LH*"
For more details on the violation symbols "H" and "L", see Chapter 13.15.1.1, "Special features of Manchester coding", on page 749.
Also, the length of the pattern must correspond to the bit field length (or the results are unpredictable).

"[ ] (array)" The **array** operator (represented by the "[" and "]" bracket signs) defines the number of permissible repetitions of the cell.

**Example: Fixed-Length Array: [n]**

The length parameter "n" is a decimal number > 0, which determines that the cell is repeated n times within the frame. If, for example, the Bits is 8, then the array operator identifies n cells of 8-bit length, and present them in the results table and honeycomb display with the specified name and color.

A fixed-length array is treated the same as other cells, except the real length of such an array is n · bit count.

**Example: Variable Length Array: []**

This array operator with empty "[" and "]" bracket signs does not determine a fixed size array. The cell could be repeated any number of times, including 0 times. As a result, the cell and the frame are of unspecified length (a situation that covers typical use cases).

The length of the frame is then determined by the end of frame condition, which can be an operator or a gap.

For processing reasons, only one (1) variable length array is supported in a frame, and a delimiter must follow immediately in the next cell after it. This is required to enable the software to correctly terminate the array. Otherwise the bits could not be assigned correctly, and it would not be possible to determine where a repletion starts and where it ends. With an end of frame condition, the software can calculate the length of one single array within a frame. But if there were more arrays, it would be impossible to know which array was how long.

The variable length array can also be the last cell of a frame. In this case, no delimiter is required. If decoded successfully, the detailed view in the results table shows the elements of the array. If the cell name of the array is "Data", then the detail view of result table displays the elements with an array index as "Data: 1", "Data: 2" ... etc.

If the variable length array cell is selected, it is shown as array [n], where n is the actual size detected in the waveform.

As an exception to the rule, it is permissible to assign the variable length array to consecutive fields. In this case, the fields are treated as a structure which is repeated. For example, if two consecutive fields are defined as A[] and B[], the decoder creates a sequence of ABABAB until the end condition has been detected.

**Note:** It is possible to combine check functions in a dynamic array. In the example above, if B[] is extended by odd(1), with "1" being the index of A[], then B checks the parity for each index of A.

| | |
|---|---|
| "crc5usb(n-m)" | The **crc 5-bit** operator performs a check for a 5\-bit CRC function using the polynomial as defined by the USB standard. n and m define the index range for the CRC check.<br>For example, if the CRC shall check fields 1 to 4, the function shall be written "crc5usb(1-5)".<br>If the range of the CRC check includes an array, all elements in the array is included in the CRC check.<br>If the check fails, the CRC field is marked as "CRC error" in the result details and displayed in the color red in the honeycomb display. The frame that contains the field is marked in the same way, except if another higher priority error is found within this frame. |
| "odd(n-m), even(n-m)" | The "**parity**" operators perform checks on odd or even parity in the given index range n to m.<br>Odd parity is fulfilled if the count of "1" bits in the range including the parity bit is odd. Even parity is fulfilled if the count of "1" bits in the range including the parity bit is even.<br>If the parity check fails, the parity field is marked as "CRC error" in the result details and displayed in red color in the honeycomb display. The frame that contains the field is marked in the same way, except if another higher priority error is found within this frame. |
| "{Block}" | The **block** operator is represented by the "{" and "}" bracket signs. Consecutive fields marked with "{Block}" and using the same name are displayed in the honeycomb display as a consecutive packet of name "Block" with the first field's color. This feature is a visual effect in the honeycomb only. |

Remote command:
`BUS<m>:CMSB:FRAMe<n>:CELL<o>:CONDition` on page 2057

**Format**
Selects from the following numeric data formats for the Condition value:

- Decimal
- Hexadecimal
- Octal
- Binary

The following rules apply:

- If the condition value contains at least one "H", "h", "L", "l", "X", "x", and the remaining characters only contain "1" and/or "0", the numeric format is automatically interpreted as binary, regardless of its definition.
- The wild-card characters "x" and "X" are only supported in binary format.
  Examples: If the numeric format is set to be "HEX", then:
  - "=1HL111000" is valid (read as binary)
  - "=0x10101" is valid (read as binary, the "x" is interpreted here as a wild card)
  - "=1010" is valid (read as HEX, with a total of 16 bits)
  - "=0x5A" is valid (read as HEX "5A", since "0x" is a valid HEX prefix; nevertheless, it is recommended to enter "5A" instead)
  - "=5X12" is invalid
  - "=1H33" is invalid

Remote command:
BUS<m>:CMSB:FRAMe<n>:CELL<o>:FORMat on page 2057

**Bit Order**
This defines, in which order the bits of a cell's Condition value are evaluated: either the most significant bit (MSB) or the least significant bit (LSB) first. Since the bit order is considered for the interpretation of the condition, you should specify MSB or LSB correctly.

Remote command:
BUS<m>:CMSB:FRAMe<n>:CELL<o>:BITorder on page 2057

**Color**
Opens a dialog to select the color representation of different cells in the honeycomb display. Assigning user-selected colors helps to interpret the decode results more easily.

In the "Adjust Colors" dialog, you can either select one of the predefined colors or define a new one.

Remote command:
BUS<m>:CMSB:FRAMe<n>:CELL<o>:CRGB on page 2058

**Result**
This determines which cells shall be displayed in which result columns of the decode table. No index means that the result is not displayed. The decode table supports three result columns, which have to be unique for each frame type. For different frame types, though, the user can define different result columns, to display unrelated information.

**Note:** To see more than the three selected results, bring up a full list of the states and values of all cells by activating "Show details" in the decode table dialog.

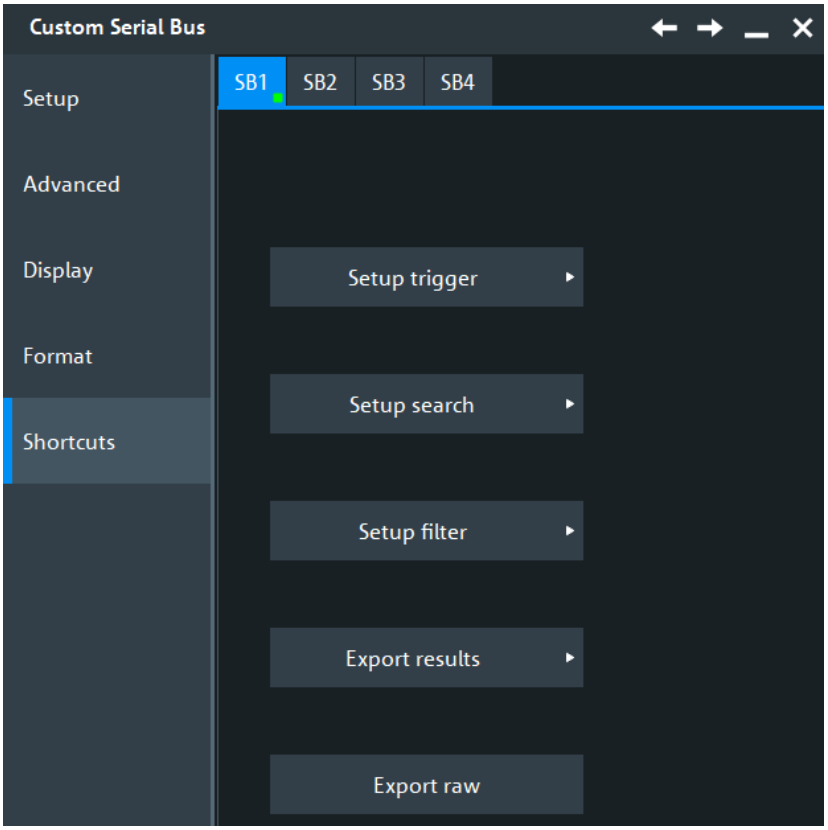
Remote command:
BUS<m>:CMSB:FRAMe<n>:CELL<o>:CLMN on page 2058

**Open or Save XML File**
For efficient working and for convenient exchange of frame descriptions, they can both be loaded ("Open") or saved ("Save" / "Save As...") in XML file format. "Explore..." opens the SaveXML folder, which is the "Default Path" for saving frame descriptions.

Remote command:
BUS<m>:CMSB:LOAD on page 2059
BUS<m>:CMSB:SAVE on page 2059

#### 13.15.2.4 Shortcuts

The "Shortcuts" tab gives quick access to other dialogs with protocol relevant settings.




**Export Raw**
Decodes the data and exports the raw data to an excel file. The file is saved under:

`c:\\Users\\Public\\Documents\\Rohde-Schwarz\\RTx\\ResultBoxExports\\CustomSBRawBits.csv`

### 13.15.2.5 XML syntax

This chapter explains the required grammar and syntax of `XML` files, which contain custom frames and can be loaded or saved. Below is a typical example of such an `XML` file:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameDescription xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Version="1" xsi:noNamespaceSchemaLocation=".\Schema\FrameDescription.xsd">
  <Frame Type="00">
    <Format Name="Start Delim" BitCount="16" Condition="=1010101010HL" NumericFormat="Binary" BitOrder="MSB First" ColorRGB="ffff00" Column="-"/>
    <Format Name="OP-00" BitCount="8" Condition="=00000000" NumericFormat="Binary" BitOrder="MSB First" ColorRGB="0080ff" Column="1"/>
    <Format Name="data" BitCount="8" Condition="[]" NumericFormat="Hex" BitOrder="MSB First" ColorRGB="10a500" Column="2"/>
    <Format Name="End Delim" BitCount="8" Condition="=1HLHL101" NumericFormat="Binary" BitOrder="MSB First" ColorRGB="ffff00" Column="-"/>
  </Frame>
  <Frame Type="01">
    <Format Name="Start Delim" BitCount="16" Condition="=1010101010HL" NumericFormat="Binary" BitOrder="MSB First" ColorRGB="ffff00" Column="-"/>
    <Format Name="OP-01" BitCount="8" Condition="=00000001" NumericFormat="Binary" BitOrder="MSB First" ColorRGB="0080ff" Column="1"/>
    <Format Name="data" BitCount="8" Condition="[]" NumericFormat="Hex" BitOrder="MSB First" ColorRGB="10a500" Column="2"/>
    <Format Name="CRC" BitCount="16" Condition="" NumericFormat="Hex" BitOrder="MSB First" ColorRGB="ff808080" Column="2"/>
    <Format Name="End Delim" BitCount="8" Condition="=1HLHL101" NumericFormat="Binary" BitOrder="MSB First" ColorRGB="ffff00"" Column="-"/>
  </Frame>   <Frame Type="ff">
    <Format Name="Start Delim" BitCount="16" Condition="=1010101010HL" NumericFormat="Binary" BitOrder="MSB First" ColorRGB="ffff00" Column="-"/>
    <Format Name="OP-ff" BitCount="8" Condition="=11111111" NumericFormat="Binary" BitOrder="MSB First" ColorRGB="0080ff" Column="1"/>
    <Format Name="data" BitCount="8" Condition="[]" NumericFormat="Hex" BitOrder="MSB First" ColorRGB="ffb233af" Column="2"/>
    <Format Name="CRC" BitCount="16" Condition="" NumericFormat="Hex" BitOrder="MSB First" ColorRGB="ff808080" Column="2"/>
    <Format Name="End Delim" BitCount="8" Condition="=1HLHL101" NumericFormat="Binary" BitOrder="MSB First" ColorRGB="ffff00" Column="-"/>
  </Frame>
</FrameDescription>
```

***Figure 13-19: Example of XML file syntax with three custom frame format descriptions***

The first out of three XML frames in Figure 13-19 is interpreted by the software in the following way:

A suitable XML file as shown in Figure 13-19 is composed as follows:

**Header:**

```
<?xml version="1.0" encoding="utf-8"?>
```

**Root Element:**

```
<FrameDescription xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 Version="1" xsi:noNamespaceSchemaLocation=".\Schema\FrameDescription.xsd">
```

The root element contains the "Frame Description" attributes, including a link for the file`FrameDescription.xsd`. This schema file, which is installed in the system, enables the software to validate an XML file before opening it.

**Frame:**

A frame description must include between *0* and *n* tags of the following kind:

```
<Frame> </Frame>
```

**Frame Type:**

Each "<Frame>" tag requires a "Type" attribute in string format:

```
<Frame Type = "string">
```

Tells the software the name of each frame.

**Format:**

Each frame must include between *1* and *n* tags of the following kind:

```
<Format> </Format>
```

Together with the attributes, this is written in short form, as in Figure 13-19:

```
<Format attribute... attribute... attribute... />
```

The format describes the fields in each frame. It can have the following attributes:

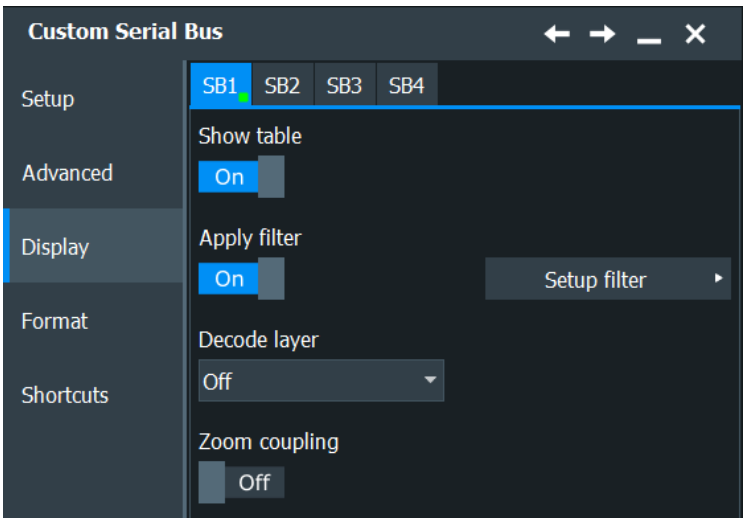
- "Field Name", in string format.
- "Bit count" specifies the length of bits, in numerical format.
- "Condition" (optional) identifies the bit pattern to match, in string format.
- "Format " allows the following choices:
  - "Decimal"
  - "Hex"
  - "Octal"
  - "Binary"
- "Bit order" allows two alternatives:
  - "MSB First" (most significant bit first)
  - "LSB First" (least significant bit first)
- "Color" allows you to set a user defined ARGB hexadecimal color value.
- The "Result Column" has four options:
  - "-" (none, which is the default)
  - "1"
  - "2"
  - "3"

### 13.15.2.6 Custom: Manchester / NRZ display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

To enhance the decode possibilities of the custom serial protocol, you can use an additional setting in the "Display" tab: "Decode layer".




**Apply Filter**
Enables filtering on a custom serial bus. For details see: Chapter 13.15.4, "Custom filter", on page 775.

Remote command:
`BUS<m>:CMSB:FILTer:ENABle` on page 2066

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the end results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

"Final" ...

"Edges" ...

"Binary" ...

"Synchronization" ...

### 13.15.2.7 Configuring custom Manchester / NRZ signals

For configuration, assign the lines to the input channels, define the active states and the logical thresholds, and specify frame format descriptions.

**Serial bus setup**

1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, select the protocol: "Custom".
4. Switch to the "Trigger" tab, tap "Source" and select "Serial bus".

   This prevents using digital waveforms (Math and Reference) as channel signals.

   **Note:** For triggering on a custom serial bus, analog input channels are required.
5. Switch back to the "Decode" tab.
6. Tap "Coding Standard" and select the coding ("Manchester", "Manchester II", "NRZ Clocked", or "NRZ Unclocked") you want to set up.
7. Select the polarity and phase of the data signal (and potentially of the clock signal).
8. Set the logical thresholds. See "Set Thresholds" on page 754.
9. Enable "Decode".
10. Switch to the "Format" tab and open or create frame format descriptions.

For details on configuration settings, see Chapter 13.15.2.1, "Custom: Manchester / NRZ configuration settings", on page 750.

## 13.15.3 Custom: Manchester / NRZ trigger

If you need information on how to get started with triggering on Custom serial bus signals, see Chapter 13.15.3.2, "Triggering on custom Manchester / NRZ serial bus", on page 775. Otherwise proceed with the Custom serial bus trigger settings.

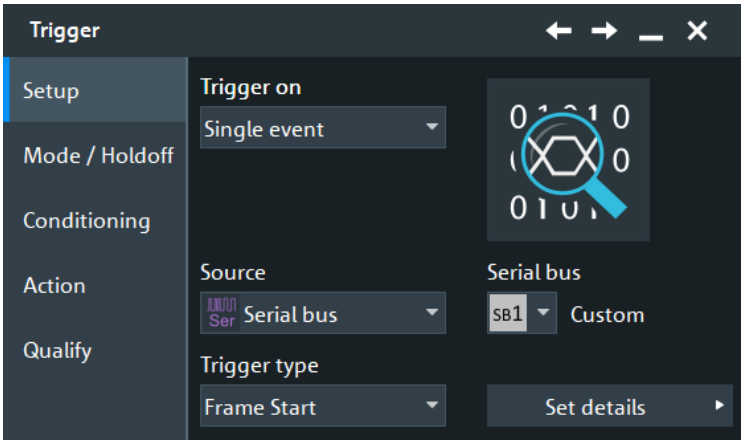
### 13.15.3.1 Custom: Manchester / NRZ trigger settings

Access: [Bus] > "Trigger" tab > "Source = Serial Bus" > select "Protocol = Custom"

In this section, all trigger settings are described. Their availability on the instrument depends on the selected coding standard and trigger type. The user interface of the instrument displays only appropriate settings and guides you through the trigger setup.

You can adjust the "Gap time" and "Bit rate" also in the "Trigger" tab if necessary.

For a list of supported trigger conditions, refer to the data sheet.




Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

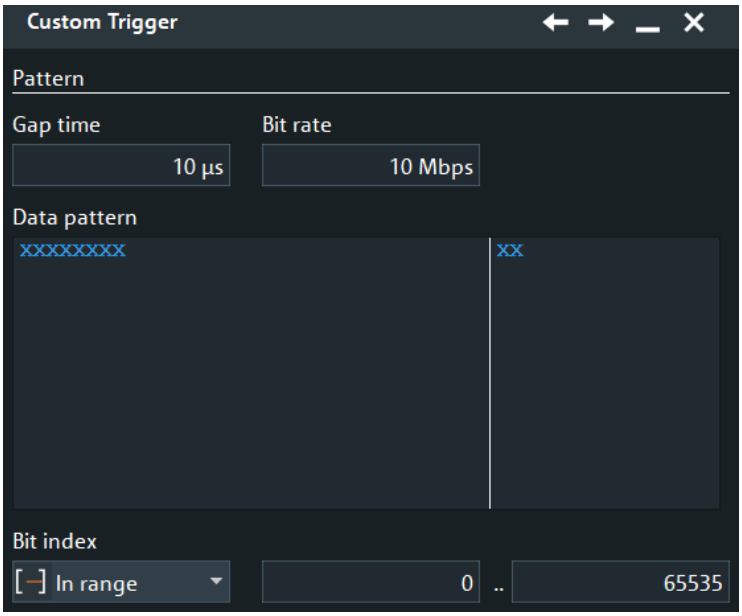
**Type**
Defines the trigger type for custom serial bus analysis. The available trigger types are "Frame Start", "Pattern" and "Advanced".

Remote command:
`TRIGger<m>:CMSB:TYPE` on page 2060

**Frame Start ← Type**
For Manchester and NRZ Clocked coding standards, the frame start trigger is set to the end of the gap time. The start of frame (SOF) condition is the first bit after the gap (timeout).

For the NRZ Unclocked coding standard, the trigger requires that the signal contains a start bit. The frame start trigger follows the gap time and is set to the end of the start bit.

**Pattern ← Type**
Specifies the pattern match conditions for a payload data check. The trigger is set to the first occurrence of a matching data bit pattern (which can be freely specified), starting after the minimum gap time, and after the detected start of the data frame. The trigger instant is after the last bit of the specified data pattern.




"Data Pattern" Specifies the data pattern that is to be found and triggered. The pattern can be entered in binary or hexadecimal format, maximum pattern length is 256 binary characters or 64 hexadecimal characters.

"Bit index operator" Sets the operator ("Equal", "Greater or equal", or "In range") to set a specific bit index (data position).

"Bit index" Sets the bit index (data position), or the start value of a bit index range. Default bit index value is 0, permitted values range from 0 to 65535.

"Bit index to" Sets the end value of a bit index range (data position range). Available only, if the "Bit index operator" is set to "In range". Default bit index end value is 65535, permitted values range from 0 to 65535.

"Word Length" Sets the number of bits in an NRZ Unclocked word (hence, the size of the data frame). Default word length is 8 bits, permitted lengths range from 0 bits to 31 bits .

**Note:** The NRZ Unclocked coding standard requires a signal that contains both a start bit and a stop bit:

- The start bit should be opposite in polarity to the idle state of the signal, and it is the first transition detected following the gap time.
- The stop bit should be the same polarity of the idle state, and it is the last bit in a data frame.

The end of the stop bit and the detection of the next frame's start bit constitutes the maximum gap time.

In order for the trigger to operate correctly, **you have to specify** the correct word length in the trigger menu. The trigger then counts the number of bits it decodes, and when the count matches the word length, the next bit is treated as the stop bit.

"Gap Time" Sets the minimum gap time for synchronization. The trigger is set to a position after the gap time, when the other trigger conditions are met. Default gap time is 10 µs, permitted gap times range from 1 ns to 1 s.

"Bit Rate" Sets the transmission speed for the data signal. Default bit rate is 10 Mbps, permitted bit rates range from 300 bps to 50 Mbps.

Remote command:

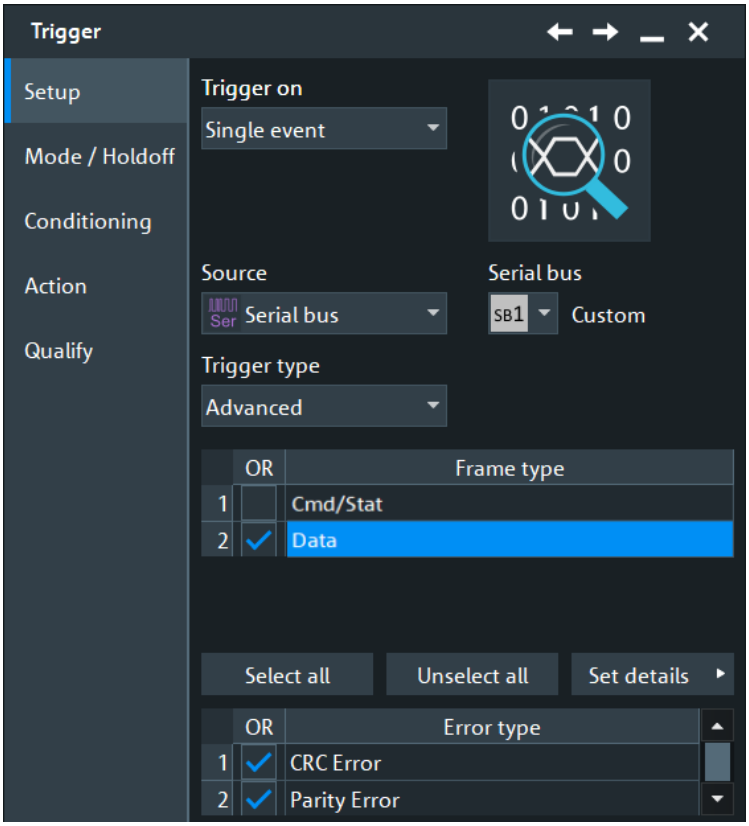
`TRIGger<m>:CMSB:PATTern` on page 2061
`TRIGger<m>:CMSB:ICONdition` on page 2061
`TRIGger<m>:CMSB:IMIN` on page 2061
`TRIGger<m>:CMSB:IMAX` on page 2062
`BUS<m>:CMSB:GAPTime:VALue` on page 2054
`TRIGger<m>:CMSB:NRZ:WRDLength` on page 2062

**Advanced**

Trigger on various frame types, fields in the frames, and data patterns.

Which settings are available, depends on what you have defined in the "Format" tab.

You can further refine the trigger criteria, selecting which frames you want to select for the triggering and what conditions their field values must fulfill.




**Error types: Item name, Enable ← Advanced**
The table lists the error types you can trigger on. Select the required errors in the "Enable" column. Available are "CRC" error and "Parity" error.

Remote command:
`TRIGger<m>:CMSB:ADVanced:ERENable` on page 2065
`TRIGger<m>:CMSB:ADVanced:ERRor<n>:ENABle` on page 2065

**Frame type ← Advanced**
Selects the frame type for the custom bus trigger analysis.

You can define individual checking parameters for the fields listed in the "Trigger type dependent settings".

To specify these parameters, select a field from this list and define the data and/or index operators and values, or the bit state.

The trigger instant is the last criterion that is fulfilled.

Remote command:
`TRIGger<m>:CMSB:ADVanced:FRAMe<n>:ENABle` on page 2062

**Field name / Condition / Enable ← Advanced**
This table lists the field numbers and names in the selected frame together with a summary of the user settings of checking conditions for each field and a checkbox to enable the checking.

Select a field in the table to specify the checking conditions for this field in the "Data", "BitState" and/or "Index" dialog (whichever applies). The condition is only applied, and the "Condition" is only shown in the table, if "Enable" is checked.

Remote command:
`TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:ENABle` on page 2063
`TRIGger<m>:CMSB:ADVanced:FIENable` on page 2063

**BitState ← Advanced**
Defines the bit state to be checked for the selected field. Permitted bit states are "1", "0" or "X" (do not care).

Remote command:
`TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:BIT` on page 2063
`TRIGger<m>:CMSB:ADVanced:BIT` on page 2063

**Data ← Advanced**
Defines for the selected field, how a data check is executed.

"Condition" Defining specific data or a data range requires to set the operator to one of the following conditions: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies data or sets the start value of a data range.

"Max" Sets the end value of a data range, if "Condition" is set to "In range" or "Out of range".

Remote command:
`TRIGger<m>:CMSB:ADVanced:DMAX` on page 2064
`TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:DMAX` on page 2064
`TRIGger<m>:CMSB:ADVanced:DMIN` on page 2064
`TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:DMIN` on page 2064
`TRIGger<m>:CMSB:ADVanced:DOPerator` on page 2064
`TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:DOPerator` on page 2064

**Index ← Advanced**
Defines for the selected field, how an index check is executed.

"Condition" Defining a specific index or an index range requires to set the operator to one of the following conditions: equal, in range.

"Min" Specifies the index or sets the start value of an index range.

"Max" Sets the end value of an index range, if "Condition" is set to "In range".

Remote command:
`TRIGger<m>:CMSB:ADVanced:IMAX` on page 2065
`TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:IMAX` on page 2065
`TRIGger<m>:CMSB:ADVanced:IMIN` on page 2066
`TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:IMIN` on page 2066
`TRIGger<m>:CMSB:ADVanced:IOPerator` on page 2066
`TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:IOPerator` on page 2066

#### 13.15.3.2 Triggering on custom Manchester / NRZ serial bus

Prerequisite: A bus is configured for the custom serial bus signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and select the "Trigger" tab.
2. Press the "Select" button. The "Source" button is shown.
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Select the serial bus that is set to Custom.
5. Tap "Type" and select the trigger type to be used for custom serial bus protocol analysis.
6. Depending on the selected custom serial bus coding standard, more setup conditions have to be specified.

For information on how to proceed with the configuration settings, see Chapter 13.15.3.1, "Custom: Manchester / NRZ trigger settings", on page 770.

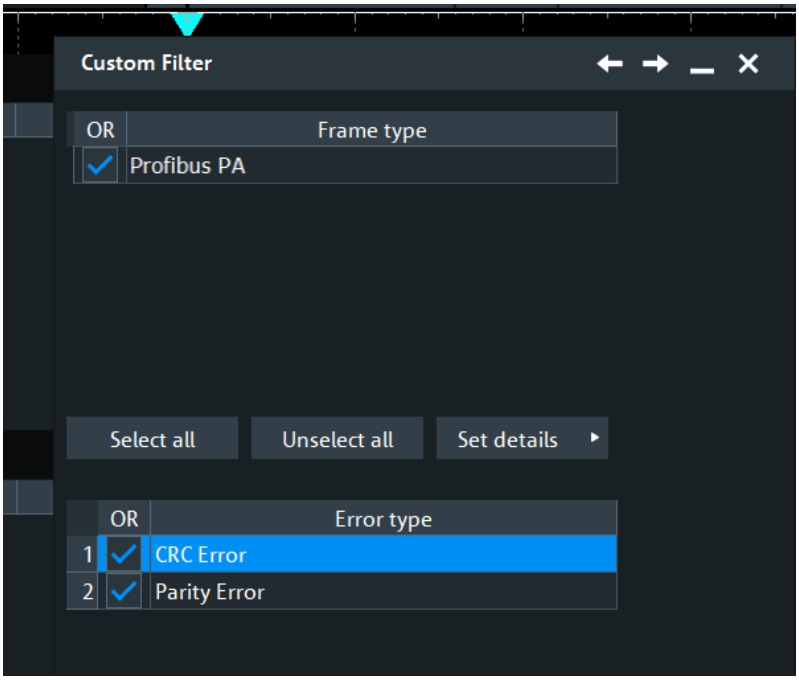
### 13.15.4 Custom filter

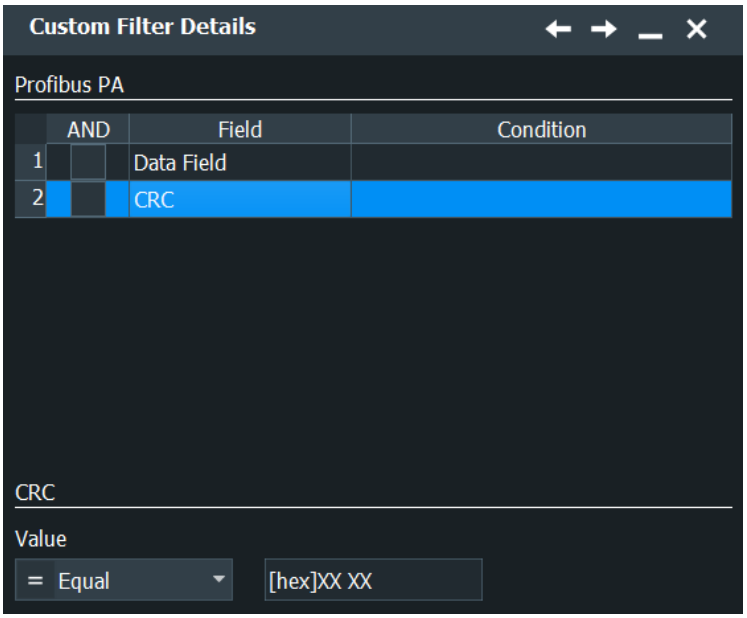
With the filter function you can display only selected events in the acquired data. Only one single filter can be enabled on the protocol at a time.

#### 13.15.4.1 Custom filter settings

Access:

1. Press [Bus].
2. Set "Protocol = Custom".
3. In the "Display" tab, enable "Apply filter".
4. Open "Setup filter".




**Frame type**
Selects the frame type for the custom bus filtering.

You can define individual checking parameters for the fields listed in the "Trigger type dependent settings".

To specify these parameters, select a field from this list and define the data and/or index operators and values, or the bit state.

Remote command:
`BUS<m>:CMSB:FILTer:FRAMe<n>:ENABle` on page 2067

**Error types: Item name, Enable**
The table lists the error types you can filter on. Select the required errors in the "Enable" column. Available are "CRC" error and "Parity" error.

Remote command:
BUS<m>:CMSB:FILTer:ERRor<n>:ENABle on page 2067

**Field name / Condition / Enable**
This table lists the field numbers and names in the selected frame together with a summary of the user settings of checking conditions for each field and a checkbox to enable the checking.

Select a field in the table to specify the checking conditions for this field in the "Data", "BitState" and/or "Index" dialog (whichever applies). The condition is only applied, and the "Condition" is only shown in the table, if "Enable" is checked.

Remote command:
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:ENABle on page 2069

**Bit State**
Defines the bit state to be checked for the selected field. Permitted bit states are "1", "0" or "X" (do not care).

Remote command:
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:BIT on page 2068

**Data**
Defines for the selected field, how a data check is executed.

"Condition" Defining specific data or a data range requires to set the operator to one of the following conditions: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies data or sets the start value of a data range.

"Max" Sets the end value of a data range, if "Condition" is set to "In range" or "Out of range".

Remote command:
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:DMAX on page 2068
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:DMIN on page 2069
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:DOPerator on page 2069

**Index**
Defines for the selected field, how an index check is executed.

This function is avalible only for array fields.

"Condition" Defining a specific index or an index range requires to set the operator to one of the following conditions: equal, in range.

"Min" Specifies the index or sets the start value of an index range.

"Max" Sets the end value of an index range, if "Condition" is set to "In range".

Remote command:
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:IMAX on page 2070
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:IMIN on page 2070
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:IOPerator on page 2070

### 13.15.5 Custom: Manchester / NRZ decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Serial Bus" dialog > "Setup" tab, enable "Decode".
2. In the "Serial Bus" dialog > "Display" tab, select additional result display settings: "Show table" and "Show binary".
3. If required, enable "Zoom coupling"

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

In contrast to other protocols, the instrument captures and decodes customizable serial bus signals according to the individual frame configuration.

#### Decode Results Table

The "Decode results" table contains information about all decoded frames.

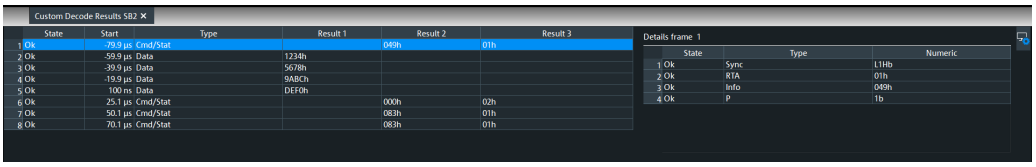



***Table 13-22: Content of the "Decode results" table***

| Column | Description |
|---|---|
| State | Overall state of the frame: either OK or the relevant error condition (e.g. preamble, length) |
| Start | Start time of the frame |
| Type | Frame type as specified in the "Frame type" field of the "Frame Format" description dialog |
| Result 1 | 1st field content as specified in the "Result" of the "Frame Format" description dialog |
| Result 2 | 2nd field content |
| Result 3 | 3rd field content |

*Table 13-23: Content of the "Decode results details" table*

| Column | Description |
|---|---|
| State | Overall state of the cell: either OK or the relevant error condition (e.g. length error) |
| Type | The type of frame, as specified with the "Frame type" |
| Numeric | Data content of the cell (e.g. 0x1, 1000LL00L) |

**Tools**

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

**Remote commands**

Remote commands are described in Chapter 24.17.16.4, "Decode results", on page 2071.

## 13.15.6 Search on decoded custom Manchester / NRZ data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search source to "SerBus" for the configured protocol.

For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

If you need information on how to get started with searching custom serial bus data, see Chapter 13.15.6.3, "Searching custom serial bus data", on page 780. Otherwise proceed with the custom serial bus search setup.

### 13.15.6.1 Custom Manchester / NRZ search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"

**Search criteria**
Use the "Search criteria" dialog to define the event types to be searched. Available event types are "Frame" and "Error".

Individual search parameters, which do not depend on the coding standard and trigger settings of the custom serial bus, can be specified in the tabs below the "Search criteria" dialog.

**Frame**
Searches for four different frame conditions.

Remote command:
SEARch:TRIGger:CMSB:ERRor<m>:ENABle on page 2077
SEARch:TRIGger:CMSB:FIENable on page 2078
SEARch:TRIGger:CMSB:FRAMe<m>:ENABle on page 2080
SEARch:TRIGger:CMSB:FRENable on page 2078
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:BIT on page 2078
SEARch:TRIGger:CMSB:BIT on page 2078
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMAX on page 2079
SEARch:TRIGger:CMSB:DMAX on page 2079
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMIN on page 2079
SEARch:TRIGger:CMSB:DMIN on page 2079
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DOPerator on page 2080
SEARch:TRIGger:CMSB:DOPerator on page 2080
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMAX on page 2081
SEARch:TRIGger:CMSB:IMAX on page 2081
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMIN on page 2081
SEARch:TRIGger:CMSB:IMIN on page 2081
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IOPerator on page 2081
SEARch:TRIGger:CMSB:IOPerator on page 2081

**Error Condition**
Searches for two error conditions: "CRC Error" or "Parity Error".

Remote command:
SEARch:TRIGger:CMSB:ERRor<m>:ENABle on page 2077

### 13.15.6.2 Custom serial bus search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands are listed in Chapter 24.17.16.6, "Search results", on page 2082

### 13.15.6.3 Searching custom serial bus data

Prerequisite: A serial bus is configured for the custom serial bus signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".

3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Specify search criteria according to Chapter 13.15.6.1, "Custom Manchester / NRZ search setup", on page 779.
5. To acquire a waveform, press [Single].

   The R&S RTP performs a custom serial bus decode according to the thresholds and protocol settings of the associated serial bus source (here in our example SB1).
6. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog:

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

## 13.16 8b/10b (option R&S RTP-K52)

The 8b/10b coding scheme is used for high-speed serial data transmission, used by different standards, like fibre channel, PCIe, DVI and USB 3.0. Also the MIPI UniPro M-PHY interface in smartphones and display interfaces such as HDMI use 8b/10b encoding, as do base stations based on the Common Public Radio Interface (CPRI).

The R&S RTP can decode differential and single-ended 8b/10b signals. Furthermore, you can search the decoded data to isolate data contents and error states such as disparity violations.

You can also trigger on defined data contents and on error states.

- The 8b/10b code ..... 781
- 8b/10b configuration ..... 782
- 8b/10b trigger ..... 787
- 8b/10b decode results ..... 790
- Search on decoded 8b/10b data ..... 791

### 13.16.1 The 8b/10b code

The 8b10b is a code where an 8-bit parallel data input is mapped into a 10-bit output. This code achieves a DC-balance and bounded disparity and simultaneously allows a reasonable clock recovery.

In the coding scheme, the original 8 bits are split into two blocks, three most significant bits and five least significant bits, see Figure 13-20. After that the 3-bit block is encoded into 4 bits and the 5-bit block into 6 bits. The 4-bit and the 6-bit blocks are then combined into a 10-bit.

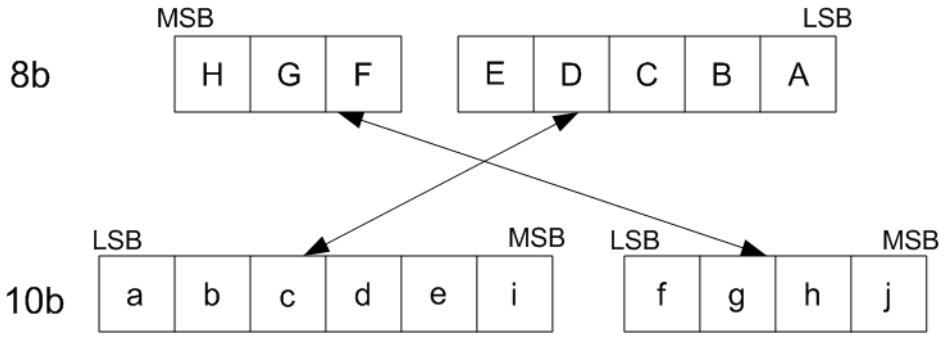



*Figure 13-20: Coding scheme of the 8b/10b*

The additional two bits I and j that join the stream are variable and ensure the running disparity.

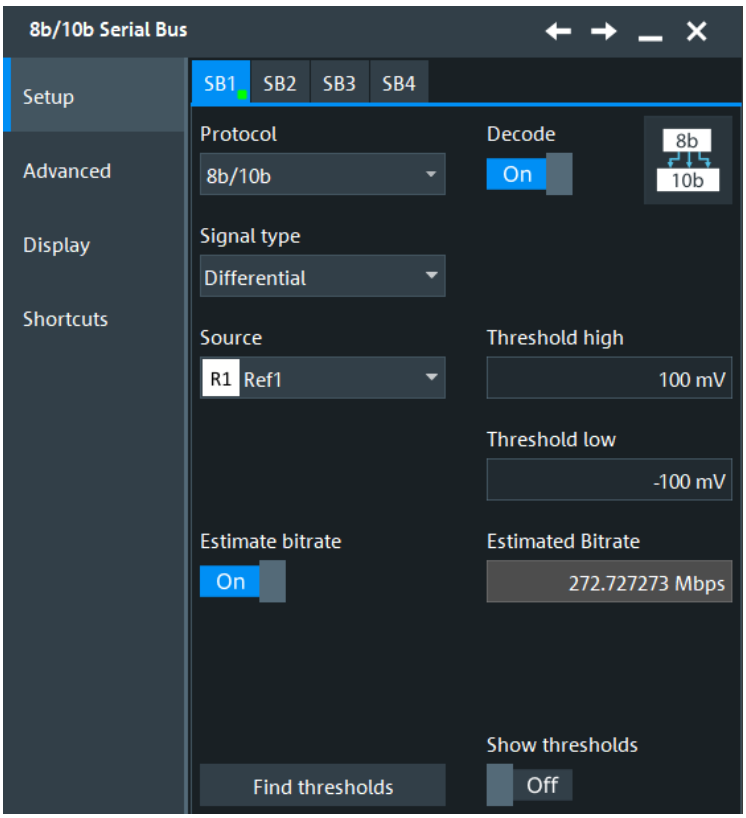
There are two types of characters used in the 8b/10b coding, special characters Kx.y and data characters Dx.y. x denotes the decimal value of EDCBA and is within the range from 0 to 31. y denotes the decimal value of HGF and is within the range 0 to 7.

## 13.16.2 8b/10b configuration

### 13.16.2.1 8b/10b configuration settings

Access: [Bus] > "8b/10b" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Signal type**
Selects the signal type that is used for the decoding. You can choose between a differential and single-ended signal.

Remote command:
`BUS<m>:EBTB:TYPE` on page 2093

**Source**
Selects the source of the provided differential signal , if "Signal type" > "Differential".

Remote command:
`BUS<m>:EBTB:DIFFerential:SOURce` on page 2089

**Source D+**
Selects the source of the provided single ended signal, if "Signal type" > "Single ended".

Remote command:
`BUS<m>:EBTB:DPLus:SOURce` on page 2091

**Source D-**
Selects the source of the provided single ended signal , if "Signal type" > "Single ended".

Remote command:
`BUS<m>:EBTB:DMINus:SOURce` on page 2090

**Threshold High**
Sets the threshold high of the signal, if "Signal type" > "Differential".

Remote command:
BUS<m>:EBTB:DIFFerential:THRHigh on page 2089

**Threshold Low**
Sets the threshold low of the signal, if "Signal type" > "Differential".

Remote command:
BUS<m>:EBTB:DIFFerential:THRLow on page 2090

**Threshold D+**
Sets the high threshold (D+) of the signal, if "Signal type" > "Single ended".

Remote command:
BUS<m>:EBTB:DPLus:THReshold on page 2091

**Threshold D-**
Sets the high threshold (D-) of the signal, if "Signal type" > "Single ended".

Remote command:
BUS<m>:EBTB:DMINus:THReshold on page 2091

**Find thresholds**
Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

Remote command:
BUS<m>:SETReflevels on page 1638

**Bitrate**
Sets the number of transmitted bits per second.

Remote command:
BUS<m>:EBTB:BITRate on page 2089

**Estimate Bitrate**
Starts a software algorithm for the automatic determination of the bitrate. The value is displayed in the "Estimated Bitrate".

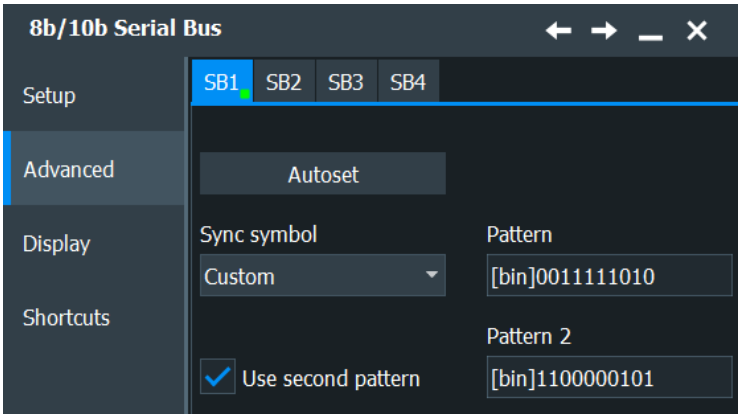
Remote command:
BUS<m>:EBTB:BITDetermi on page 2093
BUS<m>:EBTB:EBTRate? on page 2089

**Show thresholds**
If enabled, the threshold levels are displayed in the diagram.

Remote command:
BUS<m>:THReshold on page 1638

### 13.16.2.2 Advanced settings

Access: [Bus] > "Advanced" tab.




**Autoset**
Starts software algorithms for determining the signal threshold levels and bitrate.

Remote command:
`BUS<m>:FAUToset` on page 1638

**Sync symbol**
Selects the sync symbol, a control symbol used for low-level control functions. You can select one of the comma control symbols (K28.1, K28.5, K28.7) used for synchronization or you can enter a pattern.

Remote command:
`BUS<m>:EBTB:SYNC` on page 2092

**Pattern**
Selects a pattern that serves as a sync symbol.

Remote command:
`BUS<m>:EBTB:FCSY` on page 2092

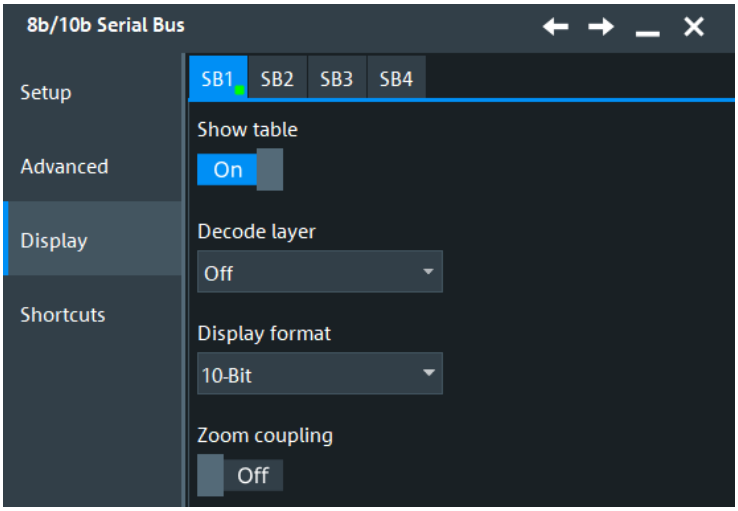
**Pattern 2**
Enables a second pattern that serves as a sync symbol and sets its value.

Remote command:
`BUS<m>:EBTB:SCSY` on page 2092
`BUS<m>:EBTB:USCS` on page 2093

### 13.16.2.3 Display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.




**Display Format**
Selects the display format for the results of the display table.

Remote command:
`BUS<m>:EBTB:DISF` on page 2090

### 13.16.2.4 Configuring 8b/10b

For configuration, you assign the line to the input channel, set the threshold, the bitrate, and the sync symbol.

For details on configuration settings, see Chapter 13.16.2.1, "8b/10b configuration settings", on page 782.

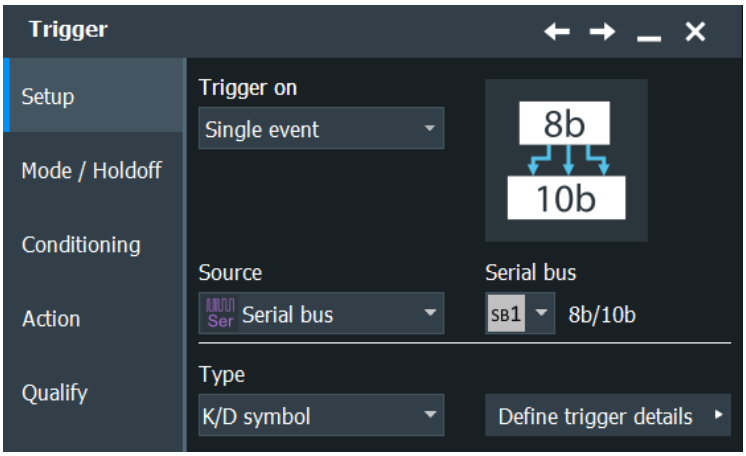
1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, set "Protocol"="8b/10b".
4. Tap the "Signal type" button, and select the used signal type for the measurement.
5. Select the source for the signal type.
6. Tap "Autoset".
7. Check the threshold and bitrate settings. Adjust the values if necessary.
8. Select the "Sync symbol" to be used for decoding.
9. Enable "Decode".

## 13.16.3 8b/10b trigger

If you need information on how to get started with triggering on 8b/10b signals, see Chapter 13.16.3.2, "Triggering on 8b/10b data", on page 790. Otherwise proceed with the 8b/10b trigger settings.

### 13.16.3.1 8b/10b trigger setup

Access: [Bus] > "Shortcuts" tab > "Setup trigger"



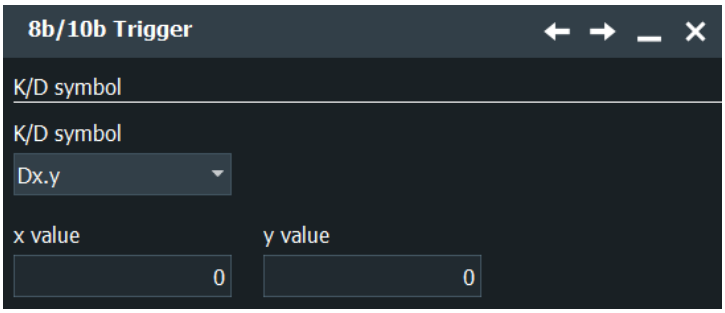

**Type 8b/10b**
Selects the type of condition to be triggered on. Available are:

- "K/D symbol"
- "8 bit pattern"
- "10 bit pattern"
- "Expression"
- "Error Condition"

Press "Define trigger details" to open a dialog for defining trigger type dependent settings.

Remote command:
`TRIGger<m>:EBTB:STYPe` on page 2096

**K/D symbol**
Selects the data character (Dx.y) or control character to be triggered on.



**K/D Codes Symbol ← K/D symbol**
Selects the control character.

Remote command:
TRIGger<m>:EBTB:SYMType on page 2095

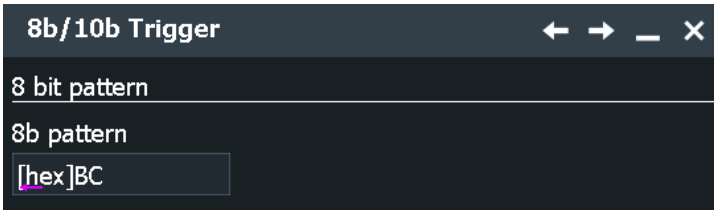
**x value ← K/D symbol**
Sets the x value of the data character Dx.y to be triggered on, if "K/D symbol" > "Dx.y".

Remote command:
TRIGger<m>:EBTB:DX on page 2094

**y value ← K/D symbol**
Sets the y value of the data character Dx.y to be triggered on, if "K/D symbol" > "Dx.y".

Remote command:
TRIGger<m>:EBTB:DY on page 2094

**8 bit pattern**
Sets the 8-bit pattern to be triggered on, if "Trigger Type" > "8 bit pattern".

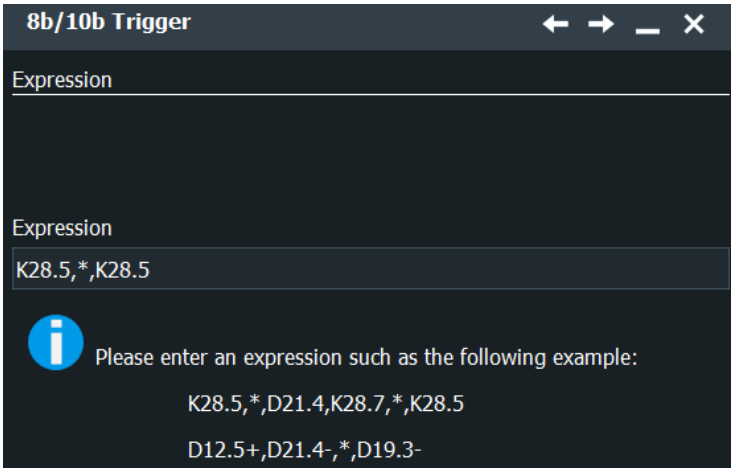



Remote command:
TRIGger<m>:EBTB:EBPA on page 2095

**10 bit pattern**
Sets the 10-bit pattern to be triggered on, if "Trigger Type" > "10 bit pattern".

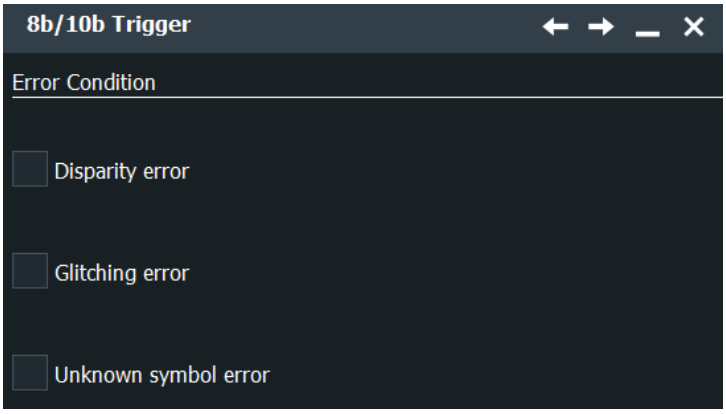
Remote command:
TRIGger<m>:EBTB:TBPA on page 2096

**Expression**
Defines the expression to be triggered on, if "Trigger Type" > "Expression".




Remote command:
TRIGger<m>:EBTB:SYME on page 2095

**Error Condition**
Triggers on the selected error conditions.




**Disparity error ← Error Condition**
Triggers on disparity errors.

Remote command:
TRIGger<m>:EBTB:DISParityerr on page 2094

**Glitching error ← Error Condition**
Triggers on glitch errors.

Remote command:
TRIGger<m>:EBTB:GLITcherror on page 2095

**Unknown symbol error ← Error Condition**
Triggers on a symbol that is not defined by the 8b/10b.

Remote command:
TRIGger<m>:EBTB:UNK on page 2096

#### 13.16.3.2 Triggering on 8b/10b data

Prerequisite: A bus is configured for the 8b/10b signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Tap "Serial bus" and select the serial bus that is set to 8b/10b.
5. Tap "Type" and select the trigger type to be used for 8b/10b protocol analysis.
6. Depending on the selected trigger type, more setup conditions have to be specified.

For information on how to proceed with the configuration settings, see Chapter 13.16.3.1, "8b/10b trigger setup", on page 787.

### 13.16.4 8b/10b decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".
2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

**Decode Results Table**

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

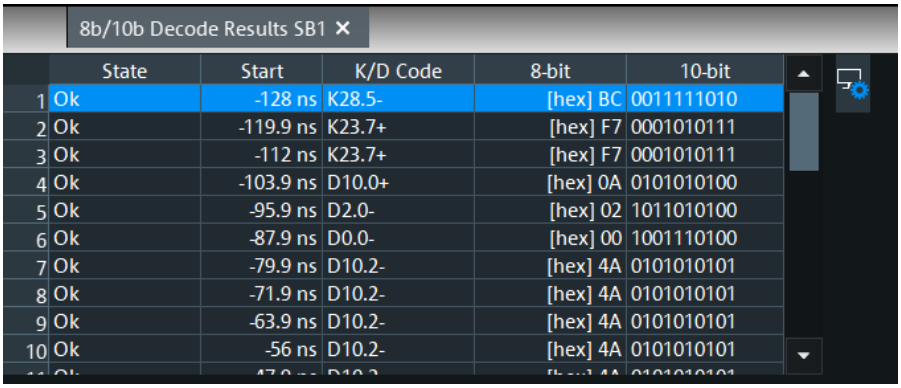



*Table 13-24: Content of the decode result table*

| Column | Description |
|---|---|
| Symbol | Shows the index number of the symbol |
| State | Overall state of the symbol |
| Start | Time of the symbol start in relation to the trigger point |
| K/D Code | The K/D code |
| 8-bit | The 8-bit value |
| 10-bit | 10-bit output value |

**Tools**

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

**Remote commands**

Remote commands are described in Chapter 24.17.17.3, "Decode results", on page 2096.

## 13.16.5 Search on decoded 8b/10b data

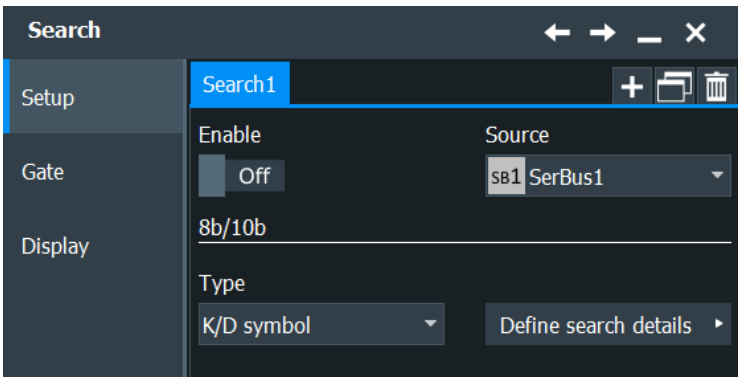
Using the search functionality, you can find various events in the decoded data, the same events which you also can trigger on. Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search "Source" to the serial bus that is configured for the protocol to be analyzed.

See also Chapter 11, "Search functions", on page 449.

### 13.16.5.1 8b/10b search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"




**Type**
Selects the type of condition to be searched for. Available are:

- "K/D symbol"
- "8 bit pattern"
- "10 bit pattern"
- "Expression"
- "Error Condition"

Press "Define search details" to open a dialog for defining search type dependent settings.

Remote command:
`TRIGger<m>:EBTB:STYPe` on page 2096

**K/D symbol**
Selects the data character (Dx.y) or control character to be searched for.

**K/D Codes Symbol ← K/D symbol**
Selects the data character (Dx.y) or control character to bea searched for. You can specify the value of the data character to be searched for with "Dx Value" and "Dy Value".

Remote command:
`SEARch:TRIGger:EBTB:SYMType` on page 2101

**y value ← K/D symbol**
Sets the y value of the data character Dx.y to be searched for, if "K/D symbol" > "Dx.y".

Remote command:
`SEARch:TRIGger:EBTB:DY` on page 2100

**x value ← K/D symbol**
Sets the x value of the data character Dx.y to be searched for, if "K/D symbol" > "Dx.y".

Remote command:
`SEARch:TRIGger:EBTB:DX` on page 2099

**Error Condition**
Searches for the selected error conditions.

**Disparity error**
Searches for disparity errors.

Remote command:
SEARch:TRIGger:EBTB:DISParityerr on page 2099

**Glitching error**
Searches for glitching errors.

Remote command:
SEARch:TRIGger:EBTB:GLITcherror on page 2100

**Unknown symbol error**
Searches for a symbol that is not defined by the the 8b/10b.

Remote command:
SEARch:TRIGger:EBTB:UNK on page 2102

**8b Pattern**
Sets the 8-bit pattern to be searched for, if "Type" > "8-bit".

Remote command:
SEARch:TRIGger:EBTB:EBPA on page 2100

**10b Pattern**
Sets the 10-bit pattern to be searched for, if "Type" > "10-bit".

Remote command:
SEARch:TRIGger:EBTB:TBPA on page 2101

**Expression**
Defines the expression to be searched for, if "Type" > "Expression".

Remote command:
SEARch:TRIGger:EBTB:SYME on page 2101

### 13.16.5.2 8b/10b search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote Commands:

- SEARch:RESult:EBTB:SCOunt? on page 2102
- SEARch:RESult:EBTB:SYMBol<m>:DATA? on page 2102
- SEARch:RESult:EBTB:SYMBol<m>:STARt? on page 2103
- SEARch:RESult:EBTB:SYMBol<m>:STATus? on page 2103

- SEARch:RESult:EBTB:SYMBol<m>:STOP? on page 2103

#### 13.16.5.3 Searching 8b/10b data

Prerequisite: A serial bus is configured for the 8b/10b signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".
3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Tap "Source" and select the serial bus that is set to 8b/10b (e.g. "SB1", unless already selected).

   The search dialog for 8b/10b protocol analysis is opened.
   There are no additional search criteria to be specified.
5. Tap "Source" and select the serial bus that is set to 8b/10b (e.g. "SB1", unless already selected).
6. Specify search criteria according to Chapter 13.16.5.1, "8b/10b search setup", on page 791.
7. To acquire a waveform, press [Single].

   The R&S RTP performs a 8b/10b decode according to the thresholds and protocol settings of the associated serial bus source (here in our example SB1).
8. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog:

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

## 13.17 MDIO (option R&S RTP-K55)

The R&S RTP-K55 option enables the R&S RTP to analyse Management Data Input/ Output (MDIO) protocols. The option is compatible with the Ethernet standard IEEE 802.3 (http://standards.ieee.org/findstds/standard/802.3-2012.html) and supports simplified triggering and decoding for both variants of MDIO: Clause 22 with basic addressing, and Clause 45 with advanced addressing that meets the requirements of 10 Gigabit Ethernet devices.

- The MDIO protocol....795
- MDIO configuration....796
- MDIO trigger....800
- MDIO decode results....803
- Search on decoded MDIO data....805


### 13.17.1 The MDIO protocol

MDIO is used for bidirectional transfer of control and status information between the physical layer entity (PHY) and the station management entities (STA).

A major application of MDIO is fault detection by interrogating registers of physical devices. Hence, MDIO serial bus visualization helps debugging new products by giving developers a quick insight into the native data on the bus without using a special decoder.

On physical level, MDIO is a clocked non-return-to-zero (NRZ) code similar to SPI. According to the Ethernet standard, the protocol defines two threshold levels, 2 V and 0.8 V, which establish a hysteresis.

On logical level, MDIO is a fairly simple protocol with a fixed word length of 64 bits. The structure of MDIO frames is shown in the following tables:

***Table 13-25: MDIO frame structure according to Clause 22***

| | Management Frame Fields | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Frame** | PRE | ST | OP | PHYAD | REGAD | TA | DATA | IDLE |
| WRITE | 1...1 | 01 | 01 | AAAAA | RRRRR | 10 | DDDDDDDDDDDDDDDD | Z |
| READ | 1...1 | 01 | 10 | AAAAA | RRRRR | Z0 | DDDDDDDDDDDDDDDD | Z |

***Table 13-26: MDIO frame structure according to Clause 45***

| | Management Frame Fields | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Frame** | PRE | ST | OP | PRTAD | DEVAD | TA | ADDRESS / DATA | IDLE |
| ADDRESS | 1...1 | 00 | 00 | PPPPP | EEEEE | 10 | AAAAAAAAAAAAAAAA | Z |
| WRITE | 1...1 | 00 | 01 | PPPPP | EEEEE | 10 | DDDDDDDDDDDDDDDD | Z |
| READ | 1...1 | 00 | 11 | PPPPP | EEEEE | Z0 | DDDDDDDDDDDDDDDD | Z |
| POST-READ increment address | 1...1 | 00 | 10 | PPPPP | EEEEE | Z0 | DDDDDDDDDDDDDDDD | Z |

The following abbreviations are used in the tables:

- PRE = preamble, consisting of 32 logic "one" bits ("1...1")
- PRE = preamble, consisting of 32 logic "one" bits ("1...1")
- ST = start of frame code (2 bits), "01" for Clause 22, "00" for Clause 45, "0X" for any, no other options permitted
- OP = operation code or "OpCode" (2 bits). This is a frame type code specifying the type of transaction.

- PHYAD = address of a physical layer entity (in Clause 22)
- PRTAD = address of a port (in Clause 45)
- REGAD = register address within a PHY (in Clause 22)
- DEVAD = device address within a port (in Clause 45)
- TA = turnaround time, a 2-bit time spacing between REGAD/DEVAD and DATA. The turnaround provides the slave some time to answer upon a read command. TA is hard-wired even in write commands, although it is not required there.
- ADDRESS / DATA = address or payload data, 16 bits
- IDLE = A single value (high-impedance state) indicating to the Physical Medium Attachment (PMA) that there is no data to convey

Instead of a specific hardware trigger, the option R&S RTP-K55 uses a predefined generic serial bus pattern trigger. It simply triggers on a bit pattern in the data stream. This is fast, but limited in the complexity of the conditions.

The MDIO trigger settings allow you to define the MDIO fields individually. The firmware concatenates the settings to a single search pattern that is then used by the serial bus pattern.

While this design is simple, it does not allow triggering on a data range or even inequality. This explains the much simpler structure compared to other protocols.

## 13.17.2 MDIO configuration

### 13.17.2.1 MDIO configuration settings

Access: [Bus] > "MDIO" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.

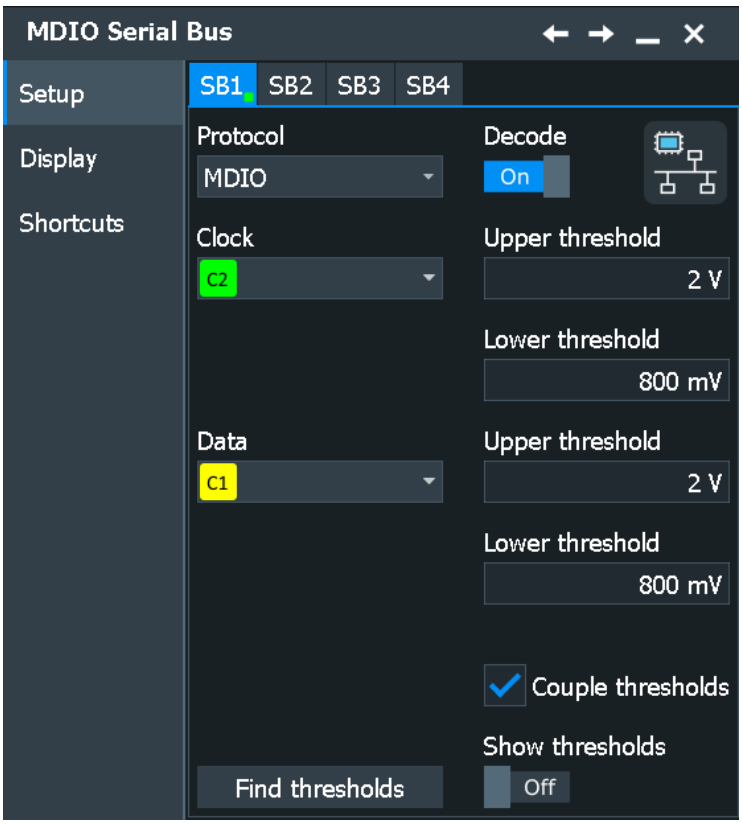



*Figure 13-21: Serial bus MDIO protocol configuration dialog*

See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Source**

MDIO requires two source channels, one for clock and one for data.

"Clock" Defines the source settings for the clock line (management data clock, MDC). Typically, select any of the analog channels "C1" – "C4" or digital channels "D0" – "D15", depending on the test application. "Math" or "Ref" waveforms are also permitted.

"Data" Defines the source settings for the data signal. Typically, select any of the analog channels "C1" – "C4" or digital channels "D0" – "D15", depending on your application, but not the same as for "Clock". "Math" or "Ref" waveforms are also permitted.

Remote command:

`BUS<m>:MDIO:CLOCk:SOURce` on page 2104

`BUS<m>:MDIO:DATA:SOURce` on page 2105

**Thresholds**

MDIO defines two thresholds for each source line:

- Vih(min) is being used for the rising edge evaluation. This "h" (high) threshold is the minimum value for the signal to be identified as "1". If the signal value comes from a low state (hence, rising edge), the state remains to be considered as low ("0"), until it has risen above Vih(min).

- Vil(max) is being used for the falling edge evaluation. This "I" (low) threshold is the maximum level for the signal to be identified as "0". If the signal value comes from a high state (hence, falling edge), the state remains to be considered as high ("1"), until it has fallen below Vil(max).

There are four ways to set the thresholds for the digitization of the signal lines:

- "Threshold"
  Enter the values directly in the fields.
- "Preset thresholds"
  Allows to select the default threshold settings according to the Ethernet standard: 2.0 V and 0.8 V.
- "Auto thresholds"
  Executes a measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.
- "Couple thresholds"
  Overwrites the data thresholds with the clock thresholds.

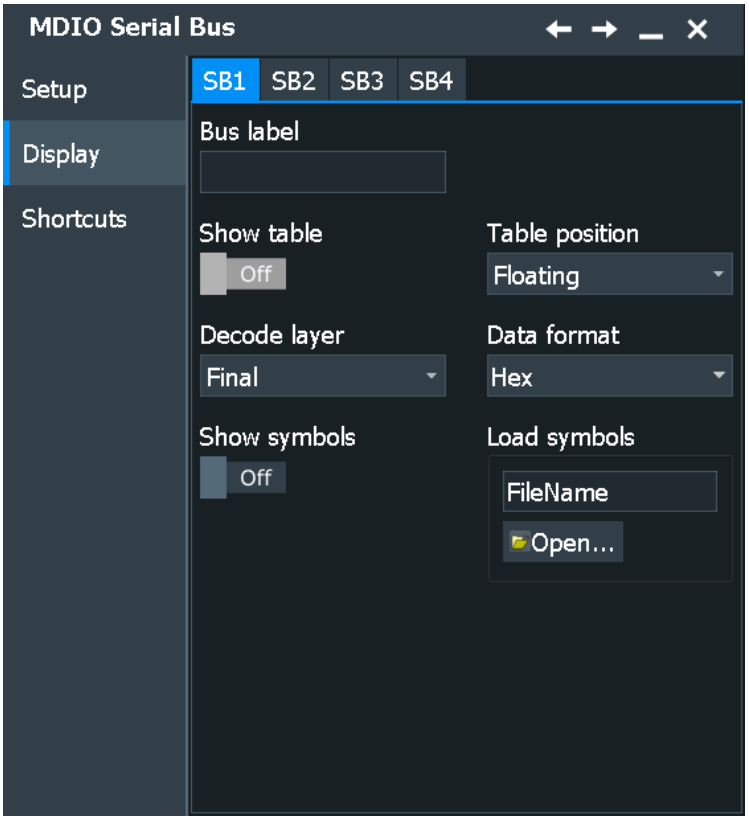
Remote command:

`BUS<m>:MDIO:CLOCk:THReshold:HIGH` on page 2105
`BUS<m>:MDIO:CLOCk:THReshold:LOW` on page 2105
`BUS<m>:MDIO:DATA:THReshold:HIGH` on page 2106
`BUS<m>:MDIO:DATA:THReshold:LOW` on page 2106
`BUS<m>:MDIO:PRESet` on page 2106
`BUS<m>:SETReflevels` on page 1638
`BUS<m>:MDIO:COUPling` on page 2107

#### 13.17.2.2 MDIO display settings

Access: [Bus] > "Setup" tab > "Protocol = MDIO" > "Display" tab

To enhance the decode possibilities of the MDIO protocol, you can use an additional setting in the "Display" tab: "Decode layer".




Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

**MDIO Specific Label Settings**

**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the end results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

Available are:

- "Final"
- "Edges"
- "Binary"

### 13.17.2.3 Configuring MDIO signals

For configuration, assign the lines to the input channels and define the active states and the logical thresholds.

For details on configuration settings, see Chapter 13.17.2.1, "MDIO configuration settings", on page 796.

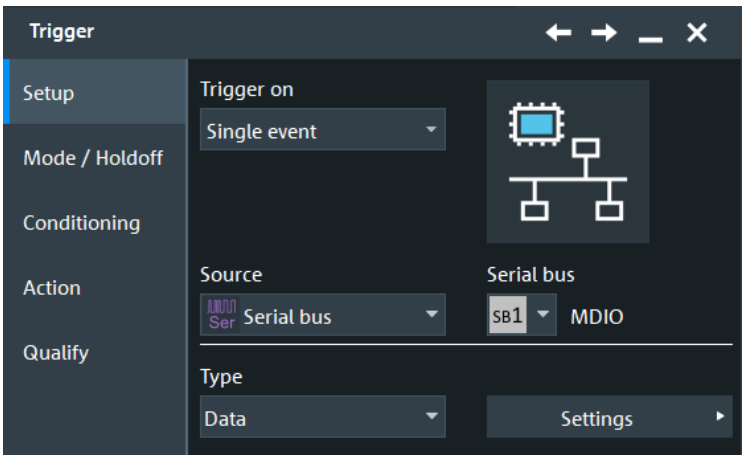
1. Press the [Bus] key on the front panel.

2. Select the tab of the bus you want to set up, for example "SB1".
3. Select the "Setup" tab.
4. In the "Setup" tab, set the "Protocol" > "MDIO".
5. Select the waveform for the "Clock" and "Data" lines.
6. Set the logical thresholds.
7. Enable "Decode".

## 13.17.3 MDIO trigger

### 13.17.3.1 MDIO trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"




In this section, all trigger settings are described. The user interface of the instrument displays only appropriate settings and guides you through the trigger setup.
For a list of supported trigger conditions, refer to the data sheet.

Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Serial bus**
Selects the serial bus to be triggered on. Make sure to select the correct bus before you enter the settings.

To trigger on a serial bus, the signals sources must be channel signals. If the data or clock source is a math or reference waveform, you cannot trigger on that bus.

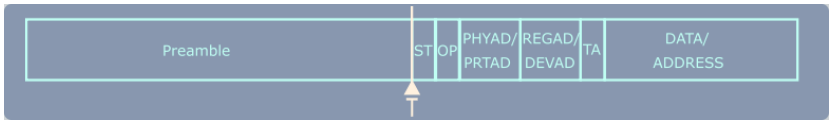
Remote command:
`TRIGger<m>:SOURce:SBSelect` on page 1642

**Protocol**
Defines the protocol type of the selected serial bus.

Remote command:
`BUS<m>:TYPE` on page 1637

**MDIO Trigger Type**
Selects the trigger type for MDIO analysis.

"Frame Start" Sets the trigger to the start of frame (SOF) field. The start of frame condition and the trigger instant is the end of the preamble.
Trigger pattern: preamble (32 bits "1")




There are no additional parameters to be specified.

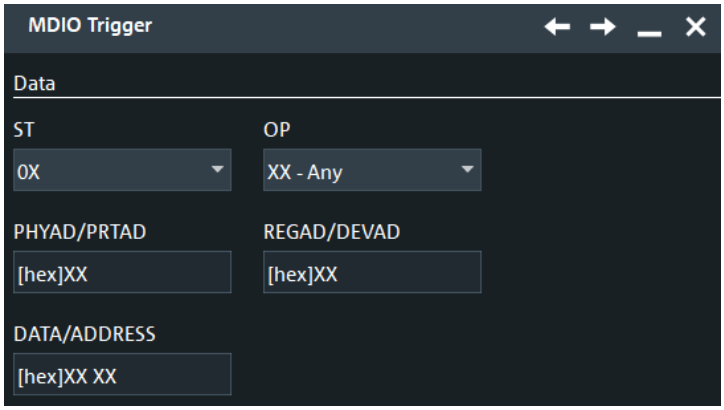
"Frame Stop" Sets the trigger to the end of frame (EOF) field. The trigger instant is after the last data bit.
Trigger pattern: preamble (32 bits "1") + 32 bits "X"

There are no additional parameters to be specified.

"Data" Sets the trigger to the data field. For more information on the data condition, see MDIO in the Ethernet standard. The trigger instant is at the end of the frame after the last data bit, as indicated in the GUI.

**Note:** All data triggers are always at the end of the frame, even if the specified pattern to trigger for is at a different position within the data word.

Trigger pattern: preamble (32 bits "1") + "ST" (2 bits, Start of Frame Code) + "OP" (2 bits, operation code or frame type code) + "PHYAD/PRTAD" (5 bits, Physical Layer Entity Address / Port Address) + "REGAD/DEVAD" (5 bits, Register Address / Device Address) + "TA" (2 "X" bits, turnaround time) + "DATA/ADDRESS" (16 bits)
For the parameters to be specified, see "ST" on page 802, "OP" on page 802, "PHYAD/PRTAD" on page 802, "REGAD/DEVAD" on page 803, and "DATA/ADDRESS" on page 803.




Remote command:
`TRIGger<m>:MDIO:TYPE` on page 2107

**ST ← MDIO Trigger Type**
Selects the start of frame code of the frame pattern; available only in trigger type "Data". Permissible frame patterns are: Clause 22, Clause 45, or Any

Remote command:
`TRIGger<m>:MDIO:ST` on page 2108

**OP ← MDIO Trigger Type**
Selects the type of frame code (or OP code, OpCode, operation code); available only in trigger type "Data". Available frame types are: Address, Write, Read, Post Read, or Any

Remote command:
`TRIGger<m>:MDIO:FRAMetype` on page 2108

**PHYAD/PRTAD ← MDIO Trigger Type**
Sets the physical address or port address (5 bits) of the frame pattern; available only in trigger type "Data".

Remote command:
`TRIGger<m>:MDIO:PHYS` on page 2109

**REGAD/DEVAD ← MDIO Trigger Type**
Sets the register address or device address (5 bits) of the frame pattern; available only in trigger type "Data".

Remote command:
`TRIGger<m>:MDIO:REGI` on page 2109

**DATA/ADDRESS ← MDIO Trigger Type**
Defines the payload data pattern or address pattern (16 bits); available only in trigger type "Data".

Remote command:
`TRIGger<m>:MDIO:DATA` on page 2109

#### 13.17.3.2 Triggering on MDIO

Prerequisite: A serial bus is configured for the MDIO signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Tap "Serial bus" and select the serial bus that is set to MDIO.
5. Tap "Trigger Type MDIO" and select the trigger type to be used for MDIO protocol analysis.
6. If the trigger type "Data" is selected, the frame pattern has to be specified.

For information on how to proceed with the configuration settings, see Chapter 13.17.3.1, "MDIO trigger settings", on page 800.

### 13.17.4 MDIO decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".
2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

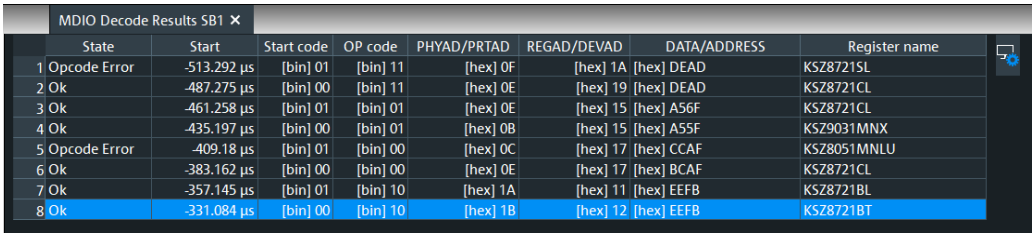
For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode Results Table

The "Decode results" table contains information about all decoded frames.




| | State | Start | Start code | OP code | PHYAD/PRTAD | REGAD/DEVAD | DATA/ADDRESS | Register name |
|---|---|---|---|---|---|---|---|---|
| 1 | Opcode Error | -513.292 µs | [bin] 01 | [bin] 11 | [hex] 0F | [hex] 1A | [hex] DEAD | KSZ8721SL |
| 2 | Ok | -487.275 µs | [bin] 00 | [bin] 11 | [hex] 0E | [hex] 19 | [hex] DEAD | KSZ8721CL |
| 3 | Ok | -461.258 µs | [bin] 01 | [bin] 01 | [hex] 0E | [hex] 15 | [hex] A56F | KSZ8721CL |
| 4 | Ok | -435.197 µs | [bin] 00 | [bin] 01 | [hex] 0B | [hex] 15 | [hex] A55F | KSZ9031MNX |
| 5 | Opcode Error | -409.18 µs | [bin] 01 | [bin] 00 | [hex] 0C | [hex] 17 | [hex] CCAF | KSZ8051MNLU |
| 6 | Ok | -383.162 µs | [bin] 00 | [bin] 00 | [hex] 0E | [hex] 17 | [hex] BCAF | KSZ8721CL |
| 7 | Ok | -357.145 µs | [bin] 01 | [bin] 10 | [hex] 1A | [hex] 11 | [hex] EEFB | KSZ8721BL |
| 8 | Ok | -331.084 µs | [bin] 00 | [bin] 10 | [hex] 1B | [hex] 12 | [hex] EEFB | KSZ8721BT |

***Table 13-27: Content of the Decode results table***

| Column | Description |
|---|---|
| Frame | Number of the acquired frame |
| State | State of frame, available messages are:<br>• OK<br>• OPcode error<br>• Length error<br>• Incomplete frame<br>• Unsynchronized bits |
| Start | Start time of the frame |
| Stop | Stop time of the frame |
| Start code | Start of frame code, 2 bits<br>• "01" (Clause 22)<br>• "00" (Clause 45) |
| OP code | Operation code (= Frame type), 2 bits<br>• "00" = Address frame (in Clause 45, only)<br>• "01" = Write frame (in Clause 22 or Clause 45)<br>• "10" = Read frame (in Clause 22) or Post Read frame (in Clause 45)<br>• "11" = Read frame (in Clause 45) |
| PHYAD/PRTAD | Address field, shown as 2 hex characters (corresponding to 5 binary bits)<br>• PHY address (in Clause 22)<br>• Port address (in Clause 45) |
| REGAD/DEVAD | Address field, shown as 2 hex characters (corresponding to 5 binary bits)<br>• Register address (in Clause 22)<br>• Device address (in Clause 45) |
| DATA/ADDRESS | Payload data field (in Clause 22 or Clause 45), or Address field (in Clause 45, only), shown as 4 hex characters or 16 binary bits |
| Register Name | Displays a translation of the PHYAD/PRTAD address label in textual form |

### Tools

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

**Remote commands**

Remote commands to retrieve decode results are described in Chapter 24.17.18.3, "Decode results", on page 2109.

## 13.17.5 Search on decoded MDIO data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search source to "SerBus" for the configured protocol.

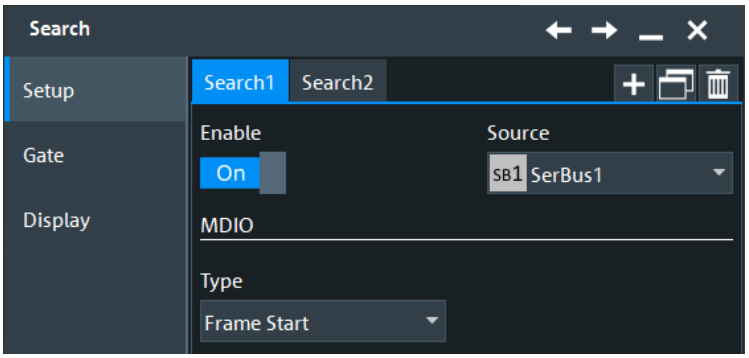
For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

If you need information on how to get started with searching MDIO data, see Chapter 13.17.5.3, "Searching MDIO data", on page 807. Otherwise proceed with the MDIO search setup.

### 13.17.5.1 MDIO search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"

**Search types**
Selects the event types to be searched.




Available event types are "Frame Start", "Frame Stop", and "Data".

Only if search criteria type "Data" is selected, individual search parameters can be specified in the tabs below the "Search criteria" dialog. For these parameters, see "ST" on page 806, "OP" on page 806, "PHYAD/PRTAD" on page 806, "REGAD/DEVAD" on page 806, and "DATA/ADDRESS" on page 806.

Remote command:
`SEARch:TRIGger:MDIO:TYPE` on page 2115

**ST**
Selects the start of frame code of the frame pattern; available only in search criteria type "Data". Permissible frame patterns are: Clause 22, Clause 45, or Any

Remote command:
`SEARch:TRIGger:MDIO:ST` on page 2115

**OP**
Selects the Type of Frame code (or OP code, OpCode, operation code); available only in search criteria type "Data". Available frame types are: Address, Write, Read, Post Read, or Any

Remote command:
`SEARch:TRIGger:MDIO:FRAMetype` on page 2114

**PHYAD/PRTAD**
Sets the physical address or port address (5 bits) of the frame pattern; available only in search criteria type "Data".

Remote command:
`SEARch:TRIGger:MDIO:PHYS` on page 2114

**REGAD/DEVAD**
Sets the register address or device address (5 bits) of the frame pattern; available only in search criteria type "Data".

Remote command:
`SEARch:TRIGger:MDIO:REGI` on page 2115

**DATA/ADDRESS**
Defines the payload data pattern or address pattern (16 bits); available only in search criteria type "Data".

Remote command:
`SEARch:TRIGger:MDIO:DATA` on page 2113

### 13.17.5.2 MDIO search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands:

- `SEARch:RESult:MDIO:WCOunt?` on page 2119
- `SEARch:RESult:MDIO:WORD<m>:STATe?` on page 2118
- `SEARch:RESult:MDIO:WORD<m>:STARt?` on page 2117

- SEARch:RESult:MDIO:WORD<m>:STOP? on page 2118
- SEARch:RESult:MDIO:WORD<m>:ST? on page 2117
- SEARch:RESult:MDIO:WORD<m>:TYPE? on page 2119
- SEARch:RESult:MDIO:WORD<m>:PHYS? on page 2116
- SEARch:RESult:MDIO:WORD<m>:REGI? on page 2117
- SEARch:RESult:MDIO:WORD<m>:DATA? on page 2116
- SEARch:RESult:MDIO:WORD<m>:SYMBol? on page 2119

#### 13.17.5.3 Searching MDIO data

Prerequisite: A serial bus is configured for the MDIO signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".
3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Tap "Source" and select the serial bus that is set to MDIO (e.g. "SB1", unless already selected).

   The search dialog for MDIO protocol analysis is opened.
   There are no additional search criteria to be specified.
5. To acquire a waveform, press [Single].

   The R&S RTP performs an MDIO decode according to the thresholds and protocol settings of the associated serial bus source (here in our example SB1).
6. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog.

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

## 13.18 USB (option R&S RTP-K60)

R&S RTP-K60 is a firmware option that enables the R&S RTP to analyze Universal Serial Bus (USB) protocols, by triggering and decoding them. The option is compatible with the standards USB 1.0, USB 1.1, USB 2.0 and USB HSIC (High-Speed Inter-Chip). R&S RTP-K60 supports the data rates "Low Speed" (1.5 Mbit/s), "Full Speed" (12 Mbit/s) and "High Speed" (480 Mbit/s, available in USB 2.0 and HSIC).

- The USB protocol....808
- USB 2.0 configuration....813
- USB 2.0 trigger....817
- USB 2.0 decode results....826
- Search on decoded USB 2.0 data....827

## 13.18.1 The USB protocol

The USB protocol was developed, starting in 1996, by the nonprofit organization USB Implementers Forum, Inc. (USB-IF), formed by Compaq, Hewlett-Packard, Intel, Lucent Technologies, Microsoft, NEC, and Philips. The purpose was to provide a common "plug-and-play" solution to replace a multitude of interfaces for the communication between computers and devices. It should allow even unskilled users to easily connect many devices to a PC. USB was originally used for devices that feature low signalling rates (up to 1.5 Mbit/s), especially human interface devices like mouse, joystick or keyboard. With the release of USB 2.0 in April 2000, devices such as printers, cameras and mass storage media were enabled to exchange data at faster rates (up to 480 Mbit/s). Also, the high-speed protocol HSIC (High Speed Inter Chip) is used for the communication between on-board devices.

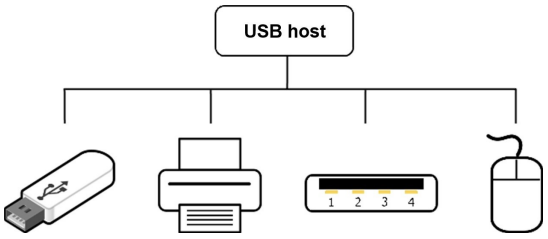



*Figure 13-22: USB Topology: mass storage device, printer, USB hub, and mouse (human interface device) connected to a USB host*

More information, including the USB specification, is available online within the web domain http://usb.org.

This chapter provides an overview of protocol characteristics, data transfer, packet structure, address and endpoint formats of USB and trigger possibilities.

### USB characteristics

Main characteristics of USB are:

- Four-wire design: USB requires a shielded cable containing four wires. Two of them, called D+ and D-, form a twisted pair (for low speed, they may not be twisted). These data lines transmit differential data signals and single-ended signal states, both referenced to a third wire: the GND or ground. The fourth wire, called VBUS (voltage bus), carries a nominal 5 V supply, which may be used to power a device.
- Host-to-device communication: in USBs "speak-when-spoken-to" protocol, communication is always initiated by the host. Consequently, there is no direct communication between USB devices, apart from few exceptions.

- Addressing scheme: a maximum of 127 connected devices can be distinguished, because a packet's address field length is limited to 7 bits. USB devices have up to 16 OUT endpoints (from host to device) and up to 16 IN endpoints (from device to host).
- USB transactions consist of two or three packets: token, data, and typically handshake
- Packet type: a packet identifier (PID) is sent as a first byte within the packet and specifies the different packet types.
- NRZI (Non Return to Zero Inverted): a zero (0) is encoded as a transition of the physical level, whereas a one (1) has no transition, thus it is represented by a steady level.

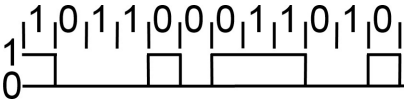



*Figure 13-23: Example of an NRZI sequence*

- Bit stuffing: a zero (0) is inserted after every 6 consecutive ones (111111). This ensures sufficient transitions to keep the phase-locked loop (PLL) synchronized and locked onto the data stream.
- Little Endian scheme: both multiple bits and multiple bytes are transmitted with the least significant bit/byte (LSB) sent first, while the most significant bit/byte (MSB) is sent last.
- HSIC (High-Speed Inter-Chip): an industry standard for USB chip-to-chip interconnection with a 2-signal (strobe, data) source synchronous serial interface, using 240 MHz DDR signaling to provide only high-speed (480 Mbps) data rate.

*Table 13-28: Simplified symbolic representation from the USB standard*

| Bus State | Protocol | Levels |
|---|---|---|
| Differential **"1"** | | D+ High, D- Low |
| Differential **"0"** | | D+ Low, D- High |
| Data **"J"** State | Low speed | D+ Low, D- High (differential **"0"**) |
| | Full speed | D+ High, D- Low (differential **"1"**) |
| Data **"K"** State | Low speed | D+ High, D- Low (differential **"1"**) |
| | Full speed | D+ Low, D- High (differential **"0"**) |

In the example in Table 13-28, High speed can be assumed to be like Full speed. For a complete overview of bus states and for an electrical definition of High and Low speed, refer to chapter 7 of the USB specification.

### Data transfer

In contrast to a conventional bus, USB is more like a network protocol, using target addresses and endpoints. However, USB features a bus master, called the host. It transmits packets of data to all devices or hubs connected to the host (or connected to a device or hub, which is in turn connected to the host). All connected devices receive

each data packet, but disregard it unless it carries the correct address. In reply, only the addressed device (one at a time) can send data upstream, to the host.

The USB architecture supports four different kinds of data transfer: control, bulk, interrupt or isochronous transfer. For example, a "bulk OUT" transfer (from host to device) would look like this:

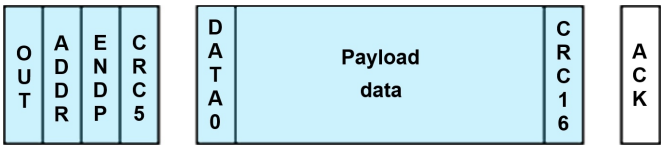



*Figure 13-24: Example of a bulk OUT transfer. Blue: host speaks, white: device speaks*

### Packet structure

All packets must start with a **SYNC** field, also called **SOP** (start of packet), which indicates data transmission. It consists of "KJ" pairs, followed by one "KK". At low speed and full speed (USB 1.x), it is 8 bits long, encoded as "KJKJKJKK". At high speed (USB 2.0 and HSIC), it is up to 32 bits long, encoded as "KJKJKJKJKJKJ...KK".

The SYNC field is used to synchronize the clock of the receiver with that of the transmitter. The final 2 bits ("KK") indicate where the PID fields starts.

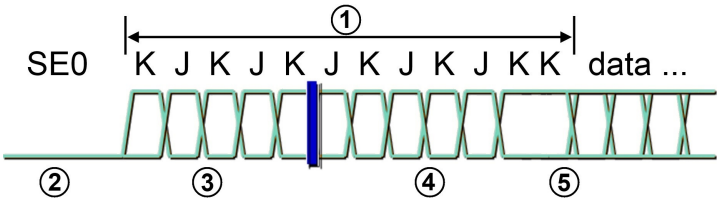



*Figure 13-25: Start of a high-speed packet: after an SE0, the packet starts with a SYNC field (or SOP), followed by the data packet*

SE0 = Single-ended zero, both D+ and D- wire are on low level
1 = High-speed SYNC field (or SOP), length up to 32 bits
2 = Receiver-squelched differential envelope, below 100 mV
3 = Differential envelope exceeds 150 mV, reception enabled with 4 symbol times
4 = Clock recovery must lock in time to detect end of SYNC
5 = End of SYNC detected, receiver begins data recovery

After the SYNC (or SOP), USB data packets consist of (in this order):

- **PID** (packet identifier): specifies the type of packet. Examples are:
  - Token packets (IN, OUT, SETUP) are always the first packet in a USB transaction, containing the address of the device and its endpoint.
  - Token SOF packets (start of frame) are there to schedule a data transfer.
  - Handshake packets provide the status of the transaction.
  - Data packets are dedicated for the data transfer payload.
  - Various special packets provide special characteristics: for instance, SPLIT packets are used to communicate with low or full speed devices on a high-speed link.

The PID's length is 8 bit, beginning (LSB) with the fields PID0, PID1, PID2 and PID3. Those 4 bits are repeated in PID4, PID5, PID6 and PID7, but in complementary form (for integrity checking). The PID is the only part of a packet not to be covered by the CRC (see below). Packets with invalid or non-supported PID fields are rejected. For valid PIDs, see Table 13-29.

- **ADDR** (address): contains the USB address of the target device. The length of this field is 7 bits, resulting in addresses 0 through 127 (with 0 reserved as the default address after reset).
- **ENDP** (endpoint): number of the endpoint. The length of this field is 4 bits, resulting in a maximum of 16 different endpoints.
- **Frame number**: this field with a length of 11 bits specifies the current frame number 0 to 2047. It is only sent in SOF packets.
- **Data field** (or "payload data"): the length of this field is not fixed, it can contain between 0 bytes and 1024 bytes.
- **CRC** (cyclic redundancy check): Finally, the CRC field ensures that the received data is correct. If an error is detected in the CRC calculation, the packet is rejected. Packets (in this case: IN, OUT, SOF, SETUP, PING, SPLIT) are followed by a CRC5 (which is 5 bits long) over ADDR and ENDP (or the frame number field, in case of SOF); data packets (DATA0, DATA1, MDATA, DATA2) are followed by a CRC16 (which is 16 bits long) over the data field. No CRC is used for handshake packets.
- **EOP** (end of package): the devices determine the length of the data field with the EOP signal, which indicates the end of a packet. In low and full speed, it consists of two SE0 bits, followed by a J state. For high speed, the current state is inverted and kept for 7-bit times, leading to a Bitstuff error that is used as an EOP indicator. This corresponds to the bit sequence 01111111 ahead of NRZI; for the SOF token, the bit sequence 01111111111111111111111111111111111111111111 is transmitted (five NRZI bytes), which is used to recognize disconnects (since high-speed devices normally do not apply voltage to D+/D-). If no device is present, a higher voltage is measured. After that, the idle state (SE0) is occupied.

***Table 13-29: Valid PIDs for different packet types***

| Packet type | Name | PID value | Meaning |
|---|---|---|---|
| Token | OUT | 0xE1 | Starts data transfer towards a device |
| | IN | 0x69 | Starts data transfer towards the host |
| | SOF | 0xA5 | Indicates start of frame |
| | SETUP | 0x2D | Starts a setup transfer and sends information on this to the device |
| Data | DATA0 | 0xC3 | Data packet with data-toggle bit 0 |
| | DATA1 | 0x4B | Data packet with data-toggle bit 1 |
| | DATA2 | 0x87 | Data packet for high speed IN isochronous transfers that require high bandwidth |
| | MDATA | 0x0F | Data packet for high-speed OUT isochronous transfers that require high bandwidth |
| Handshake | ACK | 0xD2 | Acknowledgement of a packet received without error |

| Packet type | Name | PID value | Meaning |
|---|---|---|---|
| | NAK | 0x5A | Data not accepted, typically equivalent with some type of EAGAIN, meaning that the data should be resent later |
| | STALL | 0x1E | A severe error has occurred, the target endpoint cannot be addressed until it is explicitly cleared again |
| | NYET | 0x96 | Only used in high-speed transfers, meaning ACK, but in the next interval no data can be received, therefore the host should first apply a PING |
| Special | PRE | 0x3C | Starts a low speed transfer via a full speed bus |
| | ERR | 0x3C | Indicates an error in an SPLIT transaction (using the same PID as PRE, however, these cannot be mistaken for each other) |
| | SPLIT | 0x78 | Starts an SPLIT transaction (thus: a low speed or full speed transfer via a high-speed bus) |
| | PING | 0xB4 | Used for monitoring high-speed data flow |

- Token packets (IN, OUT, SETUP) and PING packets have the following format:

| SYNC | PID | ADDR | ENDP | CRC5 | EOP |
|---|---|---|---|---|---|

- Start of frame packets (SOF) have the following format:

| SYNC | PID | Frame number | CRC5 | EOP |
|---|---|---|---|---|

- Data packets have the following format:

| SYNC | PID | Data | CRC16 | EOP |
|---|---|---|---|---|

- Handshake packets have the following format:

| SYNC | PID | EOP |
|---|---|---|

- SPLIT packets have the following format:

| SYNC | PID | ADDR | SC | PORT | S | E | ET | CRC5 | EOP |
|---|---|---|---|---|---|---|---|---|---|

**Trigger possibilities**

Signals on the input channels CH1 - CH4 of the R&S RTP can be triggered by the option R&S RTP-K60. The following trigger types are available:

- ANY Packet - Packet Sync: Triggering on the first rising slope after transmission of the packet Sync. Various lengths according to standard (in USB 2.0: 32 bit)
- Any token, OUT, IN, SOF, SETUP, AND-ing with user defined PID check, address, endpoint, CRC5: For OUT, IN, SETUP, the endpoint and CRC5 follow from the bit order, therefore such patterns can be recognized.

- Data Selection: DATA0, DATA1, DATA2, MDATA, (for USB 1.x only: AND-ing with user defined PID check, payload and CRC values)
- Handshake Packet Setup: Triggering on handshake packet, trigger with specific settings: ACK, NAK, NYET, STALL or ERR handshake packet
- Protocol Error: Triggering on PID/check error, CRC5 error, CRC16 (for USB 1.x only), frame length error (for USB 1.x only)
- Bus Event: Triggering on reset, resume, or suspend

## 13.18.2 USB 2.0 configuration

If you need information on how to get started with configuring the USB 2.0 setup, see Chapter 13.18.2.2, "Configuring USB signals", on page 817. Otherwise proceed with the configuration settings.

### 13.18.2.1 USB 2.0 configuration settings

Access: [Bus] > "USB 2.0" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.

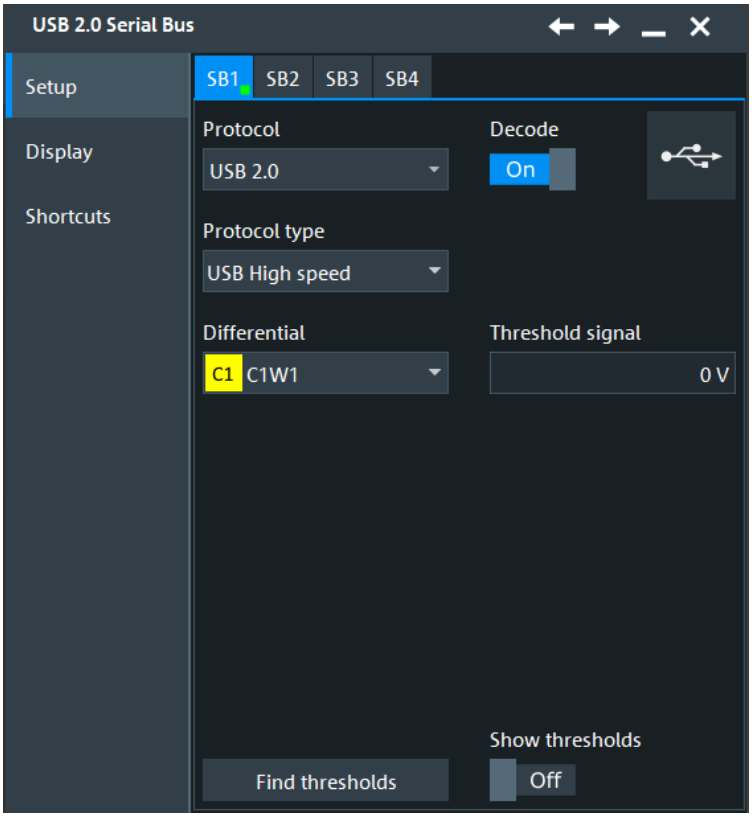



*Figure 13-26: Serial bus protocol configuration dialog*

See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Protocol type**
Defines the USB protocol technology and transmission speed.

"USB Low speed"
Selects USB 2.0 low speed protocol (1.5 Mbit/s).

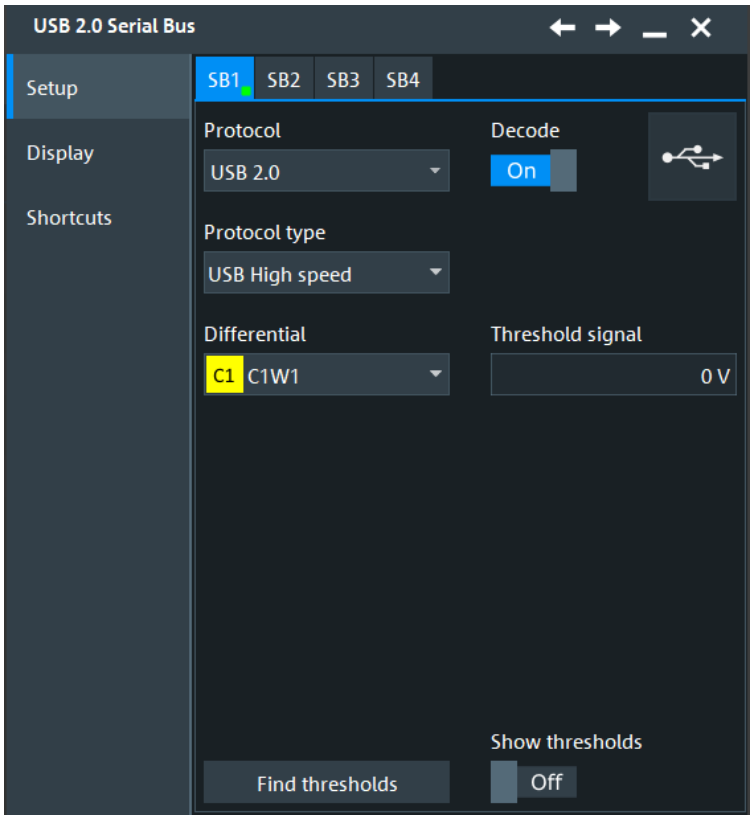



*Figure 13-27: USB low speed protocol configuration*

"USB Full speed"

Selects USB 2.0 full speed protocol (12 Mbit/s).

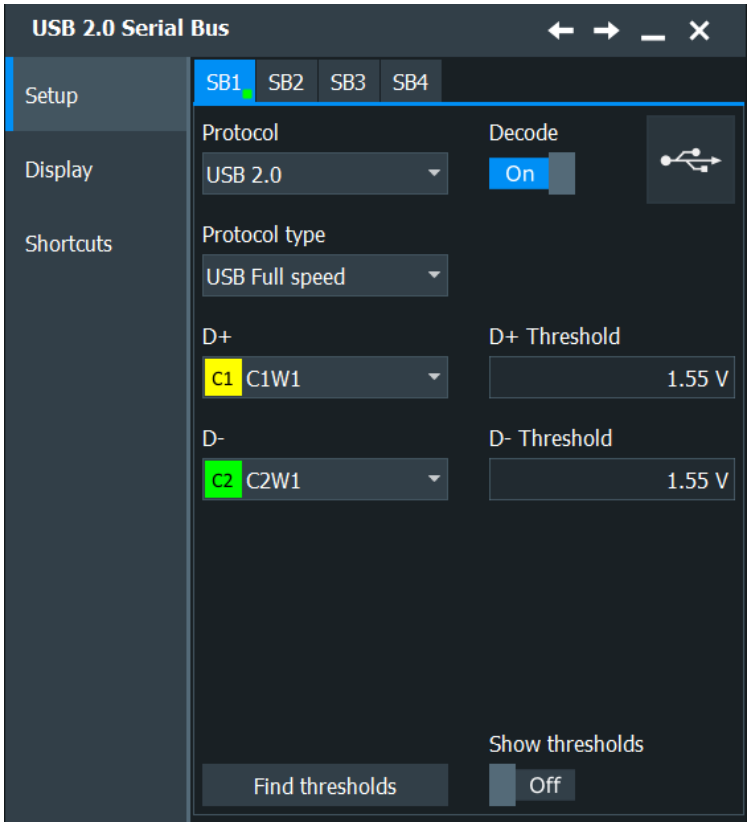



*Figure 13-28: USB full speed protocol configuration*

"USB High speed"

Selects USB high-speed protocol (480 Mbit/s). As the signal is differential, there is only one source and one threshold to be defined.

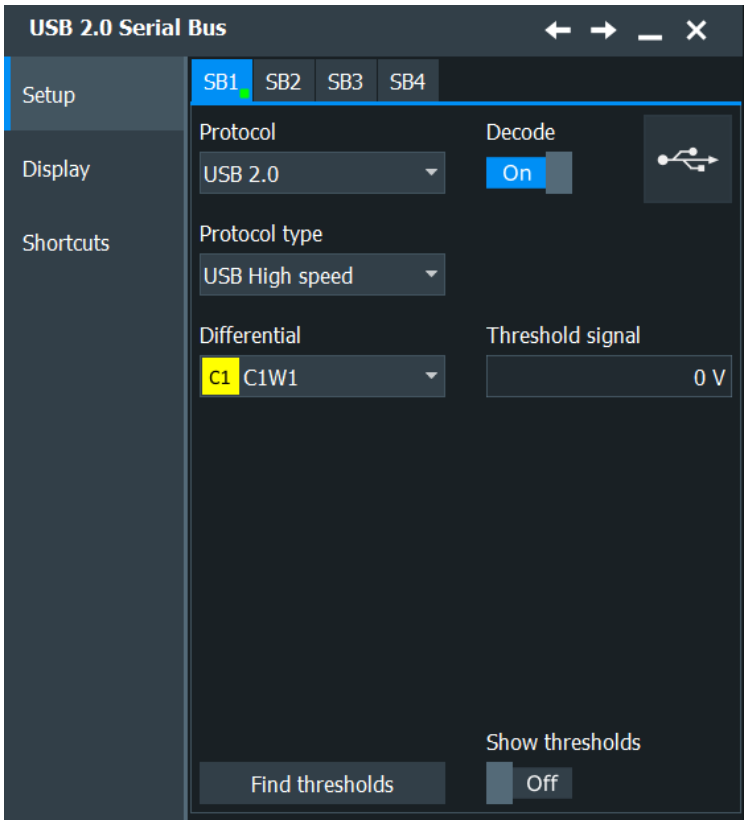



*Figure 13-29: USB high-speed protocol configuration*

Remote command:
`BUS<m>:USB:TECHnology` on page 2120

**D+, D-**

Define the source settings for the D+ and the D- data signals for USB low speed and USB full speed protocols.

Usually, the source is one of the analog channels. Reference and math waveforms are available as source if the trigger source is one of the analog channels but not the serial bus.

Remote command:
`BUS<m>:USB:DPLus:SOURce` on page 2121
`BUS<m>:USB:DMINus:SOURce` on page 2121

**Differential**

Defines the source settings for the differential data signal (in USB high-speed protocol, only). You can select analog channels "C1" to "C4", and also active math and reference waveforms as source. If you want to trigger on the serial bus, analog channel sources are required.

Remote command:
BUS<m>:USB:DIFFerential:SOURce on page 2121

**Thresholds**
Sets the threshold value for the digitization of each signal line. If the signal value on the line is higher than the threshold, the signal state is high. Otherwise, if the signal value is below the threshold, the signal state is considered low.

"Threshold" Enter the value directly in the field.

"Find thresholds" Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

Remote command:
BUS<m>:USB:DPLus:THReshold on page 2121
BUS<m>:USB:DMINus:THReshold on page 2122
BUS<m>:USB:DIFFerential:THReshold on page 2122
BUS<m>:SETReflevels on page 1638

### 13.18.2.2 Configuring USB signals

For configuration, assign the lines to the input channels and define the active states and the logical thresholds.

For details on configuration settings, see Chapter 13.18.2.1, "USB 2.0 configuration settings", on page 813.

**Serial Bus Setup**

1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, select "Protocol" = "USB 2.0".
4. Tap "Protocol type" and select the protocol type that you want to set up.
5. Depending on the protocol type, select the waveform for each of the available "D+", "D-", "Differential", "Data", and "Strobe" lines.
6. Set the logical thresholds: Either to the middle reference levels with "Find thresholds", or enter a user-defined value directly in the "Threshold" fields.
7. Enable "Decode".

## 13.18.3 USB 2.0 trigger

If you need information on how to get started with triggering on USB 2.0 signals, see Chapter 13.18.3.2, "Triggering on USB 2.0", on page 825. Otherwise proceed with the USB 2.0 trigger settings.

### 13.18.3.1 USB 2.0 trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"

In this section, all trigger settings are described. Their availability on the instrument depends on the selected USB 2.0 protocol type and trigger type. The user interface of the instrument displays only appropriate settings and guides you through the trigger setup.

For a list of supported trigger conditions, refer to data sheet.

Trigger on protocol types USB high speed and USB HSIC is currently not supported.

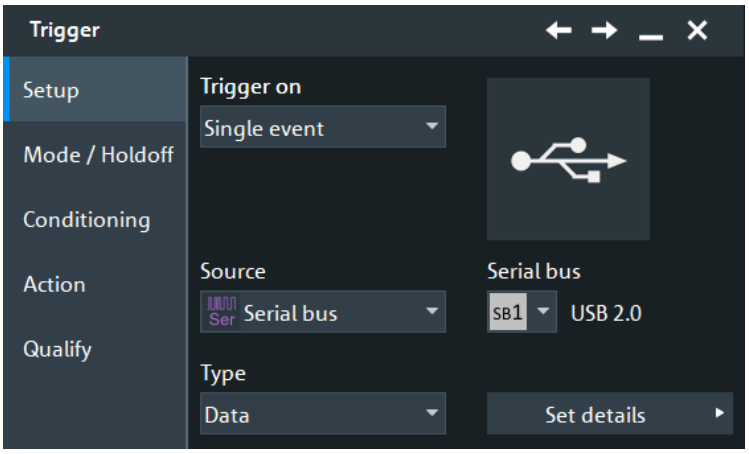



*Figure 13-30: USB trigger event settings dialog*

Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Trigger type USB**
Selects the trigger type for USB 2.0 analysis. The available trigger types depend on the USB 2.0 protocol type that is selected in the configuration setup, see "Protocol type" on page 814.

Remote command:
`TRIGger<m>:USB:TYPE` on page 2123

**Start of packet ← Trigger type USB**
Sets the trigger to the SOP (start of packet) field. The start of packet condition is the end of the SYNC field. The trigger instant is the end of the SOP field.

**End of packet ← Trigger type USB**
Sets the trigger to the EOP (end of packet) field. The trigger instant is the beginning of the EOP field.

**Reset ← Trigger type USB**
Sets the trigger to the Reset field. For more information on the reset condition, see the USB standard. The trigger instant is the end of the 10 ms period after the SE0 field.

**Suspend ← Trigger type USB**
Sets the trigger to the Suspend field. For more information on the suspend condition, see the USB standard. The trigger instant will be declared after the defined 3 ms time-out.

**Resume ← Trigger type USB**
Sets the trigger to the Resume field. For more information on the resume condition, see the USB standard. The trigger instant will be declared after the defined 20 ms timeout.

**Token ← Trigger type USB**
Sets the trigger to one out of four different token trigger types: OUT, IN, SOF, or SETUP.

See "Token" on page 820

**Data ← Trigger type USB**
Sets the trigger to one out of four different data trigger types: DATA0, DATA1, DATA2, or MDATA.

See "Data" on page 820

**Handshake ← Trigger type USB**
Sets the trigger to one out of four different handshake trigger types: ACK, NAK, STALL, or NYET.

See "Handshake" on page 821

**Special PID ← Trigger type USB**
Sets the trigger to one out of four different Special PID trigger types: PREamble, ERR, SPLIT, or PING.

See "Special PID" on page 821

**Error condition ← Trigger type USB**
Sets the trigger to one out of eight different error condition trigger types: Any error, PID error, CRC5 error, CRC16 error, Bitstuff error, Unexpected PID error, SE1 error, or Glitch error.

See "Error condition" on page 822

**Token**
Sets the trigger to one out of four different token types:
- "OUT"
- "IN"
- "SOF"
- "SETUP"

If no additional conditions are set, the trigger instant is after the PID.

If the PID error check is selected:
- If an error is found in the PID's complementary form, the trigger is set immediately after the 8th bit of the PID
- If no PID error is found, no trigger is set, even if other selected conditions are met.

Otherwise, the trigger instant is defined by the first occurrence of any of the specified additional conditions.

Remote command:
`TRIGger<m>:USB:TOKen` on page 2133

**OUT, IN or SETUP ← Token**
For the trigger token types "OUT", "IN" or "SETUP", the following conditions can be set:

You can refine the trigger condition:
- See "PID error " on page 823
- See "End Point" on page 823
- See "Address" on page 823

**SOF ← Token**
For the trigger token type "SOF", the following conditions can be set:

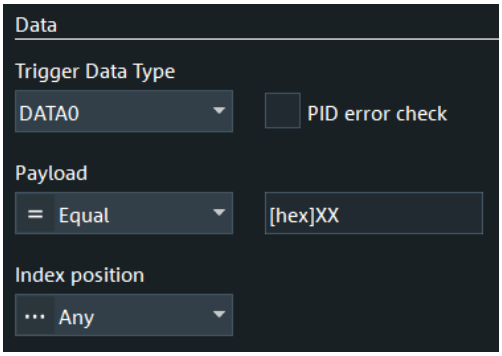
You can refine the trigger condition:
- See "PID error " on page 823
- See "Frame number" on page 824

**Data**
Sets the trigger to one out of four different data types:
- "DATA0"
- "DATA1"
- "DATA2"
- "MDATA"

If no additional conditions are set, the trigger instant is after the PID.

If the PID error check is selected:
- If an error is found in the PID's complementary form, the trigger is set immediately after the 8th bit of the PID
- If no PID error is found, no trigger is set, even if other selected conditions are met.

Otherwise, the trigger instant is defined by the first occurrence of any of the specified additional conditions.



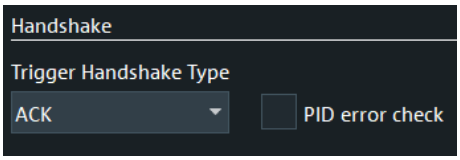

You can refine the trigger condition:
- See "PID error " on page 823
- See "Payload" on page 823

Remote command:
`TRIGger<m>:USB:DATA` on page 2125

**Handshake**
Sets the trigger to one out of four different handshake types:
- "ACK"
- "NAK"
- "STALL"
- "NYET"

If no handshake condition is set, the trigger instant is after the PID.

If the PID error check is selected:
- If an error is found in the PID's complementary form, the trigger is set immediately after the 8th bit of the PID.
- If no PID error is found, no trigger is set.




See "PID error " on page 823

Remote command:
`TRIGger<m>:USB:HAND` on page 2129

**Special PID**
Sets the trigger to one out of four different "Special PID" types:
- "PREamble"
- "ERR"
- "SPLIT"
- "PING"

If no additional conditions are set, the trigger instant is after the PID.

If the PID error check is selected:
- if an error is found in the PID's complementary form, the trigger is set immediately after the 8th bit of the PID

- if no PID error is found, no trigger is set, even if other selected conditions are met.

Otherwise, the trigger instant is defined by the first occurrence of any of the specified additional conditions.

Remote command:
`TRIGger<m>:USB:SPEC` on page 2131

**PREamble or ERR ← Special PID**
For the Trigger Special Types "PREamble" and "ERR", the following condition can be set:

See "PID error " on page 823

**SPLIT ← Special PID**
For the Trigger Special Type "SPLIT", the following conditions can be set:

You can refine the trigger condition:
- See "Address" on page 823
- See "SC" on page 824
- See "Port" on page 824
- See "SEU" on page 825
- See "ET" on page 825
- See "PID error " on page 823

**PING ← Special PID**
For the Trigger Special Type "PING", the following conditions can be set:

You can refine the trigger condition:
- See "Address" on page 823
- See "End Point" on page 823
- See "PID error " on page 823

**Error condition**
Sets the trigger in case of one of the following eight different error condition types.

The trigger instant is the first occurrence of the specified error.

| | |
|---|---|
| "Any error" | Triggers on any of the errors listed below. |
| "PID error" | Triggers on any packet identifier error. |
| "CRC5 error" | Triggers on any CRC5 error event |
| "CRC16 error" | Triggers on any CRC16 error event |
| "Bitstuff error" | Triggers in the event of an erroneous or missing bit stuffing sequence (see USB standard). |
| "Unexpected PID error" | Triggers on any illegal PID. This is a PID that is not allowed in USB low speed and USB full speed protocols, especially PID's announcing packets such as SPLIT, DATA2, MDATA, or other noncompliant packets. |
| "SE1 error" | Triggers on the illegal bus state Single Ended 1 (SE1 = both lines high). |
| "Glitch error" | Triggers on an error in the bit period (see USB standard for the definition of glitch). |

Remote command:
TRIGger<m>:USB:ERRC on page 2127

**PID error**
Defines, whether a packet ID error check is executed or not.

Remote command:
TRIGger<m>:USB:WPID on page 2134

**End Point**
Defines, whether an endpoint check that meets specific conditions is executed or not.

"Condition" Defining a specific endpoint or an endpoint range requires to set the operator to one of the following: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies an endpoint, or sets the start value of an endpoint range.

"Max" Sets the end value of an endpoint range if "Condition" (TRIGger<m>:USB:ECONdition) is set to INRange or OORange.

Remote command:
TRIGger<m>:USB:WEND on page 2133
TRIGger<m>:USB:ECONdition on page 2127
TRIGger<m>:USB:EMIN on page 2127
TRIGger<m>:USB:EMAX on page 2127

**Address**
Defines, whether an address check that meets specific conditions is executed or not.

"Condition" Defining a specific address or an address range requires to set the operator to one of the following: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies an address, or sets the start value of an address range.

"Max" Sets the end value of an address range if "Condition" (TRIGger<m>:USB:ACONdition) is set to INRange or OORange.

Remote command:
TRIGger<m>:USB:WADD on page 2133
TRIGger<m>:USB:ACONdition on page 2125
TRIGger<m>:USB:AMAX on page 2125
TRIGger<m>:USB:AMIN on page 2125

**Payload**
Defines, whether a payload check that meets specific conditions is executed or not.

"Condition" Sets the operator "any" or "equal" that allows to trigger for payload data at any position or at a specified position.

"Position" Available only if "Condition" (TRIGger<m>:USB:DPOPerator) is set to *equal*. Specifies the position in which a special data pattern is to be triggered within the payload data packet.

"Data Condition" Sets the operator ("equal" or "unequal", TRIGger<m>:USB:DCONdition) to set a specific payload data pattern.

"Payload pattern" Specifies the payload data pattern that is to be triggered.

Remote command:
TRIGger<m>:USB:WPAY on page 2134
TRIGger<m>:USB:DPOPerator on page 2126
TRIGger<m>:USB:DPOSition on page 2126
TRIGger<m>:USB:DCONdition on page 2126
TRIGger<m>:USB:PATT on page 2129

**Frame number**
Defines, whether a frame number check that meets specific conditions is executed or not.

"Condition" Defining a specific frame number or a frame number range requires to set the operator to one of the following: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies a frame number, or sets the start value of a frame number range.

"Max" Sets the end value of a frame number range if "Condition" (TRIGger<m>:USB:FCONdition) is set to INRange or OORange.

Remote command:
TRIGger<m>:USB:WFRN on page 2134
TRIGger<m>:USB:FCONdition on page 2128
TRIGger<m>:USB:FMIN on page 2128
TRIGger<m>:USB:FMAX on page 2129

**SC**
Defines, whether a Start / Complete SPLIT transaction check is executed or not.

Remote command:
TRIGger<m>:USB:WSTC on page 2135
TRIGger<m>:USB:STCO on page 2132

**Port**
Defines, whether a port check that meets specific conditions is executed or not.

"Condition" Defining a specific port number or a port number range requires to set the operator to one of the following: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies a port number, or sets the start value of a port number range.

"Max" Sets the end value of a port number range if "Condition" (TRIGger<m>:USB:PCONdition) is set to INRange or OORange.

Remote command:
TRIGger<m>:USB:WPOR on page 2134
TRIGger<m>:USB:PCONdition on page 2130

TRIGger<m>:USB:PMIN on page 2130
TRIGger<m>:USB:PMAX on page 2130

**SEU**
Defines, whether an SEU check that meets specific conditions is executed or not. S and E represent the Start and End of a start-split transaction, U represents the reserved/Unused bit of a complete-split transaction. Permissible binary SEU values are 00, 01, 10, and 11.

"Condition" Defining a specific SEU value or an SEU value range requires to set the operator to one of the following: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies an SEU value, or sets the start value of an SEU value range.

"Max" Sets the end value of an SEU value range if "Condition" (TRIGger<m>:USB:SCONdition) is set to INRange or OORange.

Remote command:
TRIGger<m>:USB:WSEU on page 2135
TRIGger<m>:USB:SCONdition on page 2130
TRIGger<m>:USB:SMIN on page 2131
TRIGger<m>:USB:SMAX on page 2131

**ET**
Defines, whether an Endpoint Type (ET) check that meets specific conditions is executed or not.

"Condition" Defining a specific endpoint type or an ET range requires to set the operator to one of the following: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies an ET, or sets the start value of an ET range.

"Max" Sets the end value of an ET range if "Condition" (TRIGger<m>:USB:TCONdition) is set to INRange or OORange.

Remote command:
TRIGger<m>:USB:WETCheck on page 2134
TRIGger<m>:USB:TCONdition on page 2132
TRIGger<m>:USB:TMIN on page 2132
TRIGger<m>:USB:TMAX on page 2133

### 13.18.3.2 Triggering on USB 2.0

Prerequisite: A bus is configured for the USB 2.0 signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".

3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Select the serial bus that is set to USB 2.0.
5. Tap "Trigger Type USB" and select the trigger type to be used for USB protocol analysis.
   Available trigger types depend on the USB 2.0 protocol type that is activated in "Protocol type" on page 814.
6. Depending on the selected USB 2.0 protocol type and trigger type, more setup conditions have to be specified.

For information on how to proceed with the configuration settings, see Chapter 13.18.3.1, "USB 2.0 trigger settings", on page 818.

## 13.18.4 USB 2.0 decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Serial Bus" dialog > "Setup" tab, enable "Decode".
2. In the "Serial Bus" dialog > "Display" tab, select additional result display settings: "Show table" and "Show binary".
3. If required, enable "Zoom coupling"

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

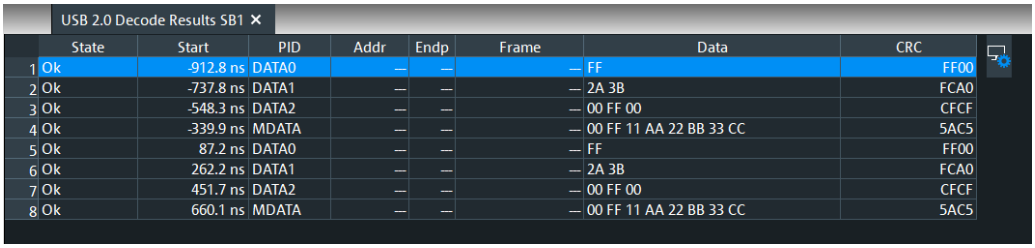
The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode Results Table

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.




| | State | Start | PID | Addr | Endp | Frame | Data | CRC |
|---|---|---|---|---|---|---|---|---|
| 1 | Ok | -912.8 ns | DATA0 | — | — | — | FF | FF00 |
| 2 | Ok | -737.8 ns | DATA1 | — | — | — | 2A 3B | FCA0 |
| 3 | Ok | -548.3 ns | DATA2 | — | — | — | 00 FF 00 | CFCF |
| 4 | Ok | -339.9 ns | MDATA | — | — | — | 00 FF 11 AA 22 BB 33 CC | 5AC5 |
| 5 | Ok | 87.2 ns | DATA0 | — | — | — | FF | FF00 |
| 6 | Ok | 262.2 ns | DATA1 | — | — | — | 2A 3B | FCA0 |
| 7 | Ok | 451.7 ns | DATA2 | — | — | — | 00 FF 00 | CFCF |
| 8 | Ok | 660.1 ns | MDATA | — | — | — | 00 FF 11 AA 22 BB 33 CC | 5AC5 |

*Table 13-30: Content of the Decode results table*

| Column | Description |
|---|---|
| State | Overall state of the packet: either OK or the relevant error condition (CRC, glitching, ...) |
| Start | Start time of the packet. |
| PID | PID type (OUT, IN, DATA0, ...) |
| Addr | Address of the recipient |
| Endp | Endpoint of the recipient |
| Frame | Frame number (in SOF packet) |
| Data | Values of the payload data bytes. The data format is selected below the table. |
| CRC | Either CRC5 or CRC16 (data packet PID) |

**Tools**

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

**Remote commands**

Remote commands to retrieve decode results are described in Chapter 24.17.19.3, "Decode results", on page 2135.

## 13.18.5 Search on decoded USB 2.0 data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

Before you can start the search, you have to configure the bus correctly and acquire decoded data.

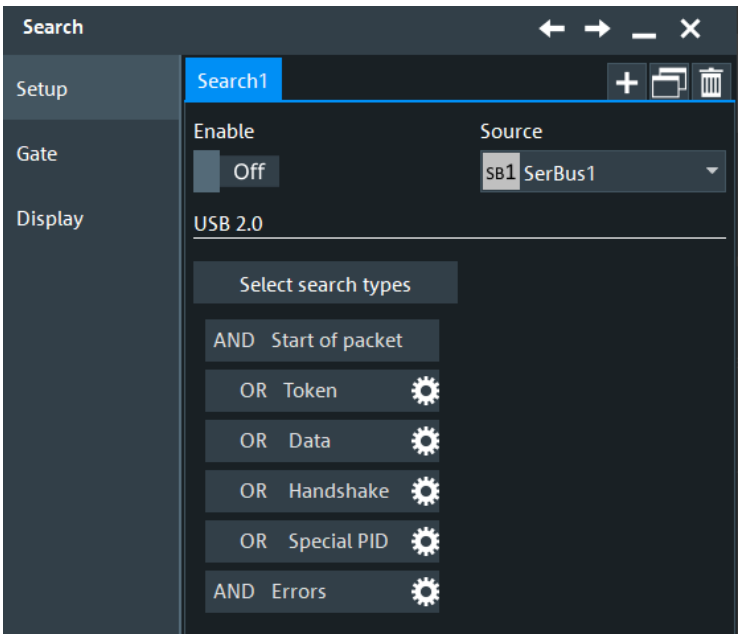
To search on decoded data, set the search source to "SerBus" for the configured protocol.

For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

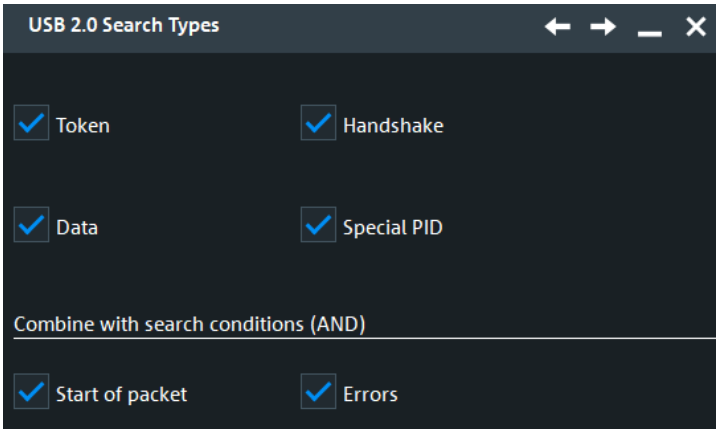
If you need information on how to get started with searching USB 2.0 data, see Chapter 13.18.5.3, "Searching USB data", on page 833. Otherwise proceed with the USB 2.0 search setup.

### 13.18.5.1 USB 2.0 search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"




**Search types**
Use the "Search Types" dialog to define the event types to be searched for.




Individual search parameters (which do not depend on the USB protocol type and trigger settings), can be specified in the tabs below the "Search criteria" dialog.

Remote command:
`SEARch:TRIGger:USB:SSOP` on page 2151
`SEARch:TRIGger:USB:STOKen` on page 2152
`SEARch:TRIGger:USB:SDATa` on page 2150
`SEARch:TRIGger:USB:SHANdshake` on page 2151
`SEARch:TRIGger:USB:SSPE` on page 2151
`SEARch:TRIGger:USB:SERRor` on page 2150

**SOP**
Searches for any start of packet. There are no additional parameters to be defined.

**Token**
Searches for four different token types: "OUT", "IN", "SOF", or "SETUP", and "Any" token.

For "Any" token, there are no additional parameters to be defined.

For "OUT", "IN", or "SETUP" tokens, additional search parameters are "Address check" and "End Point check".

You can refine the search condition:
- See "Address check" on page 830
- See "End Point check" on page 830

For "SOF" tokens, the additional search parameter is "Frame number check".

You can refine the search condition:
- See "Frame number check" on page 831

Remote command:
`SEARch:TRIGger:USB:TOKen` on page 2154

**Data**
Searches for data packets of the following types: DATA0, DATA1, DATA2, or MDATA, and "Any" data packet.

To search for payload in any data packet type, a data pattern and optionally a packet position have to be specified.

You can refine the search condition:
- See "Payload check" on page 831

Remote command:
`SEARch:TRIGger:USB:DATA` on page 2144

**Handshake**
Searches for four different handshake packet types: "ACK", "NAK", "STALL", or "NYET", and "Any" handshake packet. There are no additional parameters to be defined.

Remote command:
`SEARch:TRIGger:USB:HAND` on page 2147

**Special PID**
Searches for four different special packet identifier types: "PREamble", "ERR", "SPLIT", or "PING", and "Any" special PID.

For "Any", "PREamble", or "ERR", there are no additional parameters to be defined.

For "SPLIT", additional search parameters are "Address check", "SC check", "Port check", "SEU check", and "ET check".

You can refine the search condition:
- See "Address check" on page 830
- See "SC check" on page 832
- See "Port check" on page 832
- See "SEU check" on page 832
- See "ET check" on page 832

For "PING", additional search parameters are "Address check" and "End Point check".

You can refine the search condition:
- See "Address check" on page 830
- See "End Point check" on page 830

Remote command:
SEARch:TRIGger:USB:SPEC on page 2151

**Error condition**
Searches for the following error conditions:

"PID Error" Searches for packet identifier errors.

Remote command:
SEARch:TRIGger:USB:PIDerror on page 2149

"CRC5 Error" Searches for any CRC5 error event.

Remote command:
SEARch:TRIGger:USB:CRC5error on page 2143

"CRC16 Error" Searches for any CRC16 error event.

Remote command:
SEARch:TRIGger:USB:CRC16error on page 2143

"Bitstuff Error" Searches for bitstuff errors, thus an erroneous or missing bit stuffing sequence (see USB standard).

Remote command:
SEARch:TRIGger:USB:BITSterror on page 2143

"Glitch Error" Searches for glitch errors (errors in the bit period, see USB standard for the definition of glitch).

Remote command:
SEARch:TRIGger:USB:GLITcherror on page 2147

**Address check**
Enables the search for an address that meets specific conditions.

"Condition" Defining a specific address or an address range requires to set the operator to one of the following: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies an address, or sets the start value of an address range.

"Max" Sets the end value of an address range if "Condition" (SEARch:TRIGger:USB:ACONdition) is set to INRange or OORange.

Remote command:
SEARch:TRIGger:USB:WADD on page 2154
SEARch:TRIGger:USB:ACONdition on page 2142
SEARch:TRIGger:USB:AMIN on page 2142
SEARch:TRIGger:USB:AMAX on page 2142

**End Point check**
Enables the search for an endpoint that meets specific conditions.

"Condition" Defining a specific endpoint or an endpoint range requires to set the operator to one of the following: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies an endpoint, or sets the start value of an endpoint range.

"Max" Sets the end value of an endpoint range if "Condition" (`SEARch:TRIGger:USB:ECONdition`) is set to `INRange` or `OORange`.

Remote command:
`SEARch:TRIGger:USB:WEND` on page 2154
`SEARch:TRIGger:USB:ECONdition` on page 2145
`SEARch:TRIGger:USB:EMIN` on page 2146
`SEARch:TRIGger:USB:EMAX` on page 2146

**Frame number check**
Enables the search for a frame number that meets specific conditions.

"Condition" Defining a specific frame number or a frame number range requires to set the operator to one of the following: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies a frame number, or sets the start value of a frame number range.

"Max" Sets the end value of a frame number range if "Condition" (`SEARch:TRIGger:USB:FCONdition`) is set to `INRange` or `OORange`.

Remote command:
`SEARch:TRIGger:USB:WFRN` on page 2155
`SEARch:TRIGger:USB:FCONdition` on page 2146
`SEARch:TRIGger:USB:FMIN` on page 2147
`SEARch:TRIGger:USB:FMAX` on page 2147

**Payload check**
Enables the search for a payload data pattern that meets specific conditions.

"Condition" Sets the operator "any" or "equal" that allows to search for payload data at any position or at a specified position.

"Position" Available only if "Condition" (`SEARch:TRIGger:USB:DPOPerator`) is set to *equal*. Specifies the position in which a special data pattern is to be searched within the payload data packet.

"Data Condition" Sets the operator ("equal" or "unequal", `SEARch:TRIGger:USB:DCONdition`) to set a specific payload data pattern.

"Payload pattern" Specifies the payload data pattern that is to be searched.

Remote command:
`SEARch:TRIGger:USB:WPAY` on page 2155
`SEARch:TRIGger:USB:DPOPerator` on page 2144
`SEARch:TRIGger:USB:DPOSition` on page 2145
`SEARch:TRIGger:USB:DCONdition` on page 2144
`SEARch:TRIGger:USB:PATT` on page 2148

**SC check**
Searches for the selected Start (0) or Complete (1) SPLIT transaction endpoint, or X (do not care).

Remote command:
SEARch:TRIGger:USB:WSTC on page 2156
SEARch:TRIGger:USB:STCO on page 2152

**Port check**
Enables the search for a port that meets specific conditions.

"Condition" Defining a specific port number or a port number range requires to set the operator to one of the following: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies a port number, or sets the start value of a port number range.

"Max" Sets the end value of a port number range if "Condition" (SEARch:TRIGger:USB:PCONdition) is set to INRange or OORange.

Remote command:
SEARch:TRIGger:USB:WPOR on page 2156
SEARch:TRIGger:USB:PCONdition on page 2148
SEARch:TRIGger:USB:PMIN on page 2148
SEARch:TRIGger:USB:PMAX on page 2149

**SEU check**
Enables the search for an SEU that meets specific conditions. (For SEU, see "SEU" on page 825.)

"Condition" Defining a specific SEU value or an SEU value range requires to set the operator to one of the following: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies an SEU value, or sets the start value of an SEU value range.

"Max" Sets the end value of an SEU value range if "Condition" (SEARch:TRIGger:USB:SCONdition) is set to INRange or OORange.

Remote command:
SEARch:TRIGger:USB:WSEU on page 2156
SEARch:TRIGger:USB:SCONdition on page 2149
SEARch:TRIGger:USB:SMIN on page 2150
SEARch:TRIGger:USB:SMAX on page 2150

**ET check**
Enables the search for an Endpoint Type (ET) that meets specific conditions.

"Condition" Defining a specific endpoint type or an ET range requires to set the operator to one of the following: equal, not equal, less than, less than or equal, greater than, greater than or equal, in range, out of range.

"Min" Specifies an ET, or sets the start value of an ET range.

"Max" Sets the end value of an ET range if "Condition" (`SEARch:TRIGger:USB:TCONdition`) is set to `INRange` or `OORange`.

Remote command:
`SEARch:TRIGger:USB:WETCheck` on page 2155
`SEARch:TRIGger:USB:TCONdition` on page 2153
`SEARch:TRIGger:USB:TMIN` on page 2153
`SEARch:TRIGger:USB:TMAX` on page 2153

### 13.18.5.2 USB 2.0 search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands:

- `SEARch:RESult:USB:PCOunt?` on page 2161
- `SEARch:RESult:USB:PACKet<m>:STATus?` on page 2161
- `SEARch:RESult:USB:PACKet<m>:STARt?` on page 2160
- `SEARch:RESult:USB:PACKet<m>:STOP?` on page 2161
- `SEARch:RESult:USB:PACKet<m>:ADDRess?` on page 2157
- `SEARch:RESult:USB:PACKet<m>:DATA?` on page 2158
- `SEARch:RESult:USB:PACKet<m>:CRC?` on page 2157
- `SEARch:RESult:USB:PACKet<m>:ENDPoint?` on page 2158
- `SEARch:RESult:USB:PACKet<m>:ET?` on page 2158
- `SEARch:RESult:USB:PACKet<m>:FRAMe?` on page 2159
- `SEARch:RESult:USB:PACKet<m>:PID?` on page 2159
- `SEARch:RESult:USB:PACKet<m>:PORT?` on page 2159
- `SEARch:RESult:USB:PACKet<m>:SC?` on page 2160
- `SEARch:RESult:USB:PACKet<m>:SEU?` on page 2160

### 13.18.5.3 Searching USB data

Prerequisite: A serial bus is configured for the USB signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".

3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Tap "Source" and select the serial bus that is set to USB 2.0 (e.g. "SerBus1", unless already selected).
5. Specify search criteria according to Chapter 13.18.5.1, "USB 2.0 search setup", on page 827.
6. To acquire a waveform, press [Single].

   The R&S RTP performs a USB 2.0 decode according to the thresholds and protocol settings of the associated serial bus source (here in our example SB1).
7. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog.

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

## 13.19 USB 3.1 (option R&S RTP-K61/K62)

Option R&S RTP-K61 analyzes Universal Serial Bus (USB) signals encoded by the USB 3.1 Gen1 coding standard. To use the full functionality of the option, you need an instrument that covers the frequency range up to 5 GHz, because the USB 3.1 Gen 1 protocol is defined for data rates up to 5 GHz.

Option R&S RTP-K62 analyzes USB signals encoded by the USB 3.1 Gen2 coding standard.

For analysis, USB 3.1 encoded signals can be triggered, decoded and searched.

This chapter describes:

- The USB 3.1 protocol....834
- USB 3.1 configuration....843
- USB 3.1 trigger....848
- USB 3.1 decode results....851
- Search on decoded USB 3.1 data....853

### 13.19.1 The USB 3.1 protocol

The USB 3.1 protocol standard was published on July 26th, 2013, strongly enhancing the performance of the USB 2.0 protocol. Compared to USB 2.0, USB 3.1 defines a new physical layout, better power distribution and higher data rates.

The USB 3.1 standard features two speed levels (generations):

- **USB 3.1 Gen 1** for 5 Gbps (also labeled "SuperSpeed" or "SS"), encompassing and superseding USB 3.0 (published on November 12th, 2008)

- **USB 3.1 Gen 2** for 10 Gbps (also labeled "SuperSpeedPlus" or "SSP")

While USB 3.1 Gen 2 is not yet very common, you can use a fast oscilloscope (minimum 5 GHz) with option **R&S RTP-K61** to analyze signals encoded according to the **USB 3.1 Gen 1** protocol.

**NOTICE**

**Oscilloscope must range up to 5 GHz, at least**

As the USB 3.1 Gen 1 protocol is defined for data rates up to 5 GHz, option R&S RTP-K61 requires a fast oscilloscope that covers this frequency range.

USB 3.1 supersedes the USB 3.0 standard but uses, for example, the USB 3.0 Standard-A connector design with 9 pins. Hence, USB 3.1 cables and connectors contain 5 additional wires and pins compared to USB 2.0.

*Figure 13-31: USB 3.1 cable cross-section and Standard-A plug - backward compatible with USB 2.0*

A = Shield (braid) / connector shell
B = USB 2.0 unshielded twisted pair
C = USB 3.0 / USB 3.1 shielded twisted pair
D = USB 3.0 / USB 3.1 shielded twisted pair
1 = Voltage bus $V_{CC}$ power supply pin, +5 V
2 = Differential data signal D-
3 = Differential data signal D+
4 = Ground pin for power return
5 = SuperSpeed receiver differential pair Rx-
6 = SuperSpeed receiver differential pair Rx+
7 = Ground drain for signal return
8 = SuperSpeed transmitter differential pair Tx-
9 = SuperSpeed transmitter differential pair Tx+
* = Connector images courtesy of Wikipedia authors smial and Unconventional2

*Table 13-31: Backward compatibility of USB 3.1 connectors*

| Connector type | USB 3.1 | Compatibility | USB 2.0 |
|---|---|---|---|
| **Standard-A** | plug | ... is backward compatible with ... | receptacle |
| | receptacle | ... is backward compatible with ... | plug |
| **Standard-B** | plug | ... is **not** compatible with ... | receptacle |
| | receptacle | ... is backward compatible with ... | plug |

| Connector type | USB 3.1 | Compatibility | USB 2.0 |
|---|---|---|---|
| Micro-USB | plug | ... is **not** compatible with ... | receptacle |
| | receptacle | ... is backward compatible with ... | plug |

To achieve the data throughput of USB 3.1, all involved components (host, cable, device and optional hub) must comply with USB 3.1 specifications. If any component only complies, for example, with USB 2.0 Hi-Speed specifications, the setup works, but limited to Hi-Speed USB data rates.

USB 3.1 also specifies a new connector format, called Type-C, with a reversible plug. This small and durable connector is, however, not mechanically compatible with USB 2.0 connectors.

### USB 3.1 Type-C connector

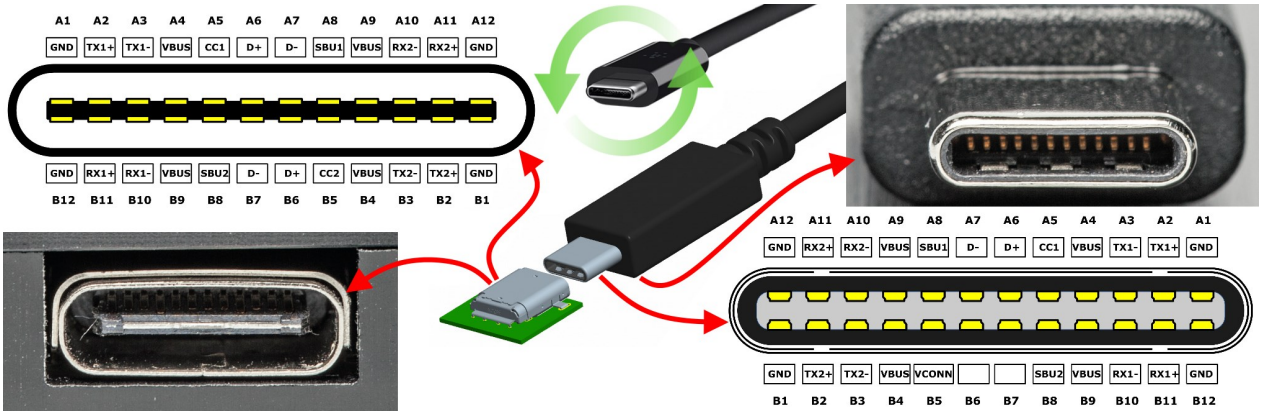



*Figure 13-32: USB 3.1 Type-C receptacle and reversible plug*

A1 to A12 = Twelve USB Type-C pins
B1 to B12 = The same USB Type-C pins as A1 to A 12, but in reverse order
* = Images courtesy of Heise Medien, Acon, phoneArena and Wikipedia author Chindi.ap

### Specific protocol features

Key features of USB 3.1 Gen 1 that are addressed by option R&S RTP-K61 include:

- Since 5 Gbps is too fast for serial pattern triggering, the option uses a search-based software trigger
- The protocol uses a non-return-to-zero (NRZ) unclocked signal with **8b/10b encoding** (while USB 3.1 Gen 2 uses 128b/132b encoding)
- The signal is scrambled with a 32-bit minimum correlation sequence
- COMMA (K28.5), which occurs only during the reset phase, is the only unique bit sequence
  - COMMA is used for reset of the scrambling sequence, which has 32-bit minimum correlation
  - COMMA and packet framing are used for bit alignment
- The protocol uses a set of different frame types that are all covered by individual trigger and search dialogues of option R&S RTP-K61

### Frame types

All frame types are listed below. The frame types are sorted in an almost alphabetical order, as in the oscilloscope's graphical user interface (GUI).

Each item in this list contains:

- the SCPI name of the frame
  (hence the name of the frame according to the related `remote control command (SCPI)`)
- the GUI name of the frame, which is sometimes identical with the SCPI name
  (the GUI names are highlighted in **bold** font and linked to one of Table 13-32 / Table 13-33 / Table 13-34)
- a brief description of the frame type

### List of frame types

- ACK - **ACK** = Handshake packet indicating a positive acknowledgment.
- BCNT - **BCNT** = BERT count, with BERT = Bit Error Rate Test. BCNT is a K28.3 sequence followed by the error count symbol EC. See also BERC.
- BDAT - **BDAT** = BERT data, ordered set of a BERT command. This data sequence (output of the scrambler) is used for BER testing and is equivalent to the logical idle sequence, consisting of scrambled 0.
- BERC - **BERC** = BERT error count, ordered set of a BERT command that does not increment the error count register. BERC is not looped back to the receiver but replaced with BCNT.
- BRST - **BRST** = BERT reset, ordered set of a BERT command. Any time a BRST is received, the error count register EC is set to 0 and the linear feedback shift register (LFSR) for scrambling is set to 0FFFFh. Any number of consecutive BRST ordered sets may be received.
- BIAM - **Bus Interval Adjustment** = The bus interval is the period that establishes the integral boundary of service intervals. It is equivalent to the Microframe interval (THSFRAM) defined in the USB 2.0 specification.
- CP0 - compliance data pattern. Runs in scrambled mode.
- CP1/CP2 - compliance data pattern. Run in unscrambled mode.
- CP3/CP5or6 - compliance data pattern. Run in both scrambled and unscrambled mode.
- DPH - **DPH** = Data Packet Header, containing a type field, the data packet's address, route string, length, and other information about the packet, including a 2-byte checksum (CRC-16).
- DPH n-def - = non-deferred Data Packet Header, containing a type field, the data packet's address, route string, length field replica, and other information about the packet, including a 2-byte checksum (CRC-16). Valid only for USB 3.1 Gen 2.
- DPP - **Data Packet Payload** = The packet's payload data and a 32-bit CRC.
- DPPA - **Data Packet Payload Abort** = Frame ordered set used to abort a data packet payload.
- ERDY - **ERDY** = The Endpoint Ready notification is a handshake packet indicating a positive acknowledgment.

- FWAKe - **Function Wake** = The function wake device notification is sent from a device (that is enabled for remote wakeup) to the host, if it wants to exit from "device suspend" or "function suspend".
- HRR - **Host Role Request** = This notification type is reserved for being issued by a SuperSpeed On-the-Go (SS-OTG) device, if its Role Swap Protocol (RSP) capability has been enabled.
- IDLE - **IDLE** = Logical Idle is a period of one or more symbol periods when no information (packets or link commands) is transferred on the link. The Idle Symbol (IS) is transmitted by a port at any time in the active (U0) state meeting the logical idle definition.
- ITP - **ITP** = Isochronous Timestamp Packet, sent periodically by a host to inform devices on the USB of the current bus time. ITPs are multicast on all active links.
- LTM - **Latency Tolerance** = Latency Tolerance Messaging is an optional normative USB power management feature that utilizes reported BELT (Best Effort Latency Tolerance) values to enable more power efficient platform operation.
- LC - **Link Command** = An eight-symbol sequence used for link-level flow control, retries, power management and device removal.
- LDM - **Link Delay Measurement** = A protocol for determining propagation delays through the USB topology with a high degree of accuracy (sometimes called "Precision Time Measurement").
- NRDY - **NRDY** = The Not Ready response is a handshake packet indicating a negative acknowledgment.
- PING - **PING** = This transaction packet can only be sent by the host, to transition all links, which are in the path to a device, back to the active (U0) state prior to initiating an isochronous transfer.
- PNGR - **PING_RESPONSE** = This transaction packet is sent by a device in response for each PING received from the host.
- PCAP - **Port Capability** = This link management packet describes each port's link capabilities and is sent by both link partners after the successful completion of training and link initialization.
- PCNF - **Port Configuration** = All Enhanced SuperSpeed ports that support downstream port capability can send this link management packet (LMP), which is similar to Port Capability. If a port, which was to be configured in the upstream facing mode, does not receive this LMP within a defined time after link initialization, then the upstream port shall transition to "eSS.Disabled" and a peripheral device shall try to connect at the other speeds this device supports.
- PCNR - **Port Configuration Response** = This link management packet is sent by the upstream port in response to a Port Configuration and indicates its acceptance or rejection.
- SDS - **Start of Data Stream** marks the start of an SDS ordered set. Valid only for USB 3.1 Gen 2.
- SKIP - **SKIP** = SKP compensates for different bit rates between two communicating ports. SKPs may be dynamically inserted or removed from the data stream. SKPEND marks the boundary between SKP symbols and the remainder of the SKP ordered set. (For SuperSpeedPlus operation, unscrambled.)

- STALL - **STALL** = This transaction packet can only be sent by an endpoint on the device. It is used to inform the host that the endpoint is halted or that a control transfer is invalid.
- STATUS - **STATUS** = This transaction packet can only be sent by the host to a control endpoint. It is used to inform a control endpoint that the host has initiated the Status stage of a control transfer.
- SLF - **Set Link Function** = This link management packet is used to configure functionality that can be changed without leaving the active (U0) state.
- SSPD - **Sublink Speed** = Speed of a sublink, which is the collection of receive or transmit lanes between a downstream facing port (DFP) and an upstream facing port (UFP).
- TS1/TS1A/TS1B - **TS1** = Training sequences are ordered sets for initializing bit and symbol alignment and receiver equalization. Examples are TS1, TS2 and TSEQ.
- TS2 - **TS2** = See TS1
- TSEQ - **TSEQ** = See TS1
- U2IT - **U2 Inactivity Timeout** = This link management packet is used to define the timeout from U1 to U2.
- VDT - **Vendor Device Test** = Use of this link management packet is intended for vendor-specific device testing and shall not be used during normal operation of the link.

**Other types**

- ERRor - **Error** = This is not a frame type, but it is included as an artificial entry in the search and navigate dialog to allow selecting and activating the error search. If the host error bit is set to "1", this state indicates that the host was unable to accept the data. The reason can be a CRC error, length error, value out of range, packet incomplete or unknown error.
- UDEF - **Unknown** = Undefined data cannot be identified as any specific USB 3.1 frame, because the measured signal does not match with the specifications of any defined frame. Therefore, in this case the field is represented by a "?". You cannot search for "Unknown" or trigger on it, but these types are returned, if an unexpected (sub)type has been found. If some rough type identification is possible, the unknown data may fit one of the following descriptions of unknown types.
- UDVN - **Unknown Device Notification** = A Device Notification transaction packet can only be sent by a device, to inform the host of an asynchronous change in a device or interface state (e.g., to identify the function within a device that caused the device to perform a remote wake operation).
- UHP - **Unknown Header Packet** = An undefined header packet that starts with HPSTART, but then the type is invalid.
- ULMP - **Unknown LMP** = A Link Management Packet is a type of header packet primarily used to manage links by communicating information between links partners. LMPs only travel between pairs of directly connected ports.
- UTP - **Unknown TP** = A Transaction Packet is a type of header packet used to communicate information between a device and the host. TPs traverse all the links, directly connecting the host to a device. They have no payload data, instead they are used, e.g., to control the flow of data packets or configure devices and hubs.

The frames listed above consist of individual sets of fields. Some frames only contain one field, others are much longer: up to 29 fields (in case of the "ACK" frame).

We try to represent the structure of these frames. However, one single table that contains all frames and all fields would be very large: 30 by 40 cells. We have therefore split up the full table into the following four smaller tables:

- Table 13-32 contains the **short** frames that end with the "Link Control Word". (This table uses abbreviation, see below.)
- Table 13-33 contains the **long** frames that end with the "Link Control Word". (This table uses abbreviation, see below.)
- Table 13-34 contains the frames that do not end with the "Link Control Word". These frames have no additional fields other than those in the table.
- Table 13-35 contains the types that are no specified USB 3.1 frames.

The frames in both Table 13-32 and Table 13-33 are abbreviated. For the full set of fields, you must add to each frame the 2 fields "HPSTART" and "Type" in the beginning of the frame. Also add a "CRC" field and the 6 fields of the "Link Control Word" at the end of each frame. In Table 13-35, only the "Link Control Word" is abbreviated.

The 2-byte **"Link Control Word"** is used for both link level and end-to-end flow control. The fields in the Link Control Word are detailled in Table 13-33.

***Table 13-32: Short frames that start with fields HPSTART and Type and end with CRC and the Link Control Word***

| Frame | Field3 | Field4 | Field5 | Field6 | Field7 | Field8 | Fld.9 | Field10 | Fld.11 |
|---|---|---|---|---|---|---|---|---|---|
| Bus Interval Adjustment | Rsvd | Device Address | SubType | NotType | Rsvd | BusIntAdj | Rsvd | | |
| Function Wake | Rsvd | Device Address | SubType | NotType | Interface | Rsvd | Rsvd | | |
| Host Role Request | Rsvd | Device Address | SubType | NotType | RSP | Rsvd | Rsvd | | |
| ITP | Counter | Delta | Adj Ctrl | Corr | Rsvd | Rsvd | | | |
| Latency Tolerance | Rsvd | Device Address | SubType | NotType | BELT | Rsvd | Rsvd | | |
| Link Delay Measurement | SubType | LDM Type | LDMS | Rsvd | Response Delay | Rsvd | Rsvd | | |
| NRDY | Rsvd | Device Address | SubType | Rsvd | D | EPTNum | Rsvd | StreamId | Rsvd |
| PING | Route String | Device Address | SubType | Rsvd | D | EPTNum | Rsvd | Rsvd | |
| PING_RESPONSE | Rsvd | Device Address | SubType | Rsvd | D | EPTNum | Rsvd | Rsvd | |
| Port Configuration | SubType | Link Speed | Rsvd | Rsvd | Rsvd | | | | |
| Port Configuration Response | SubType | Response Code | Rsvd | Rsvd | Rsvd | | | | |
| STALL | Rsvd | Device Address | SubType | Rsvd | D | EPTNum | Rsvd | Rsvd | |
| Set Link Function | SubType | SLF | Rsvd | Rsvd | Rsvd | | | | |

| Frame | Field3 | Field4 | Field5 | Field6 | Field7 | Field8 | Fld.9 | Field10 | Fld.11 |
|---|---|---|---|---|---|---|---|---|---|
| U2 Inactivity Time-out | SubType | U2 IT | Rsvd | Rsvd | Rsvd | | | | |
| Vendor Device Test | SubType | VDT | Rsvd | Vendor Def | ... for fields no. 1, 2 and n+x see Table 13-33 | | | | |

*Table 13-33: Long frames that start with fields HPSTART and Type and end with CRC and the Link Control Word*

| **Frame** | **Fld.3** | **Field4** | **Fld.5** | **Field6** | **Fld.7** | **Fld.8** | **Field9** | **F.10** | **Fld.11** | **Fld.12** | **Fld.13** | **F.14** | **F.15** | **Fld.16** | **F.17** | **F.18** | **F.19** | **F.20** | **F.21** | **F.22** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ACK | Route String | Device Address | Sub-Type | Rsvd | rty | D | EPT-Num | TT | HE | NumP | SeqNum | Rsvd | TPF | StreamId | Rsvd | SSI | WPA | DBI | PP | NBI |
| DPH | Route String | Device Address | Seq-Num | Rsvd | EOB | D | EPT-Num | TT | S | Length | StreamId | Rsvd | SSI | WPA | DBI | PP | NBI | | | |
| ERDY | Rsvd | Device Address | Sub-Type | Rsvd | D | EPT-Num | Rsvd | Num P | Rsvd | StreamId | Rsvd | | | | | | | | | |
| Port Capa-bility | Sub-Type | Link Speed | Rsvd | Num HP Buf | Rsvd | Dir | OTG | Rsvd | Tie-breaker | Rsvd | Rsvd | | | | | | | | | |
| STA-TUS | Route String | Device Address | Sub-Type | Rsvd | D | EPT-Num | Rsvd | Rsvd | PP | Rsvd | | | | | | | | | | |
| Sublink Speed | Rsvd | Device Address | Sub-Type | Not-Type | Rsvd | TPF | Rsvd | LSE | ST | Rsvd | Lanes | LP | LSM | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
| Ahead of **Field 3**, add these two fields (**Field1** + **Field2**): | | | | | | | **Field1** | | **Field2** | | **Field3** | ... | | | | | | | | |
| | | | | | | | HPSTART | | Type | | see above | ... | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
| After the **last field (n)**, add CRC + the **"Link Control Word"**: | | | | | | | **Field n+1** | | **Field n+2** | | **Field n+3** | **Field n+4** | | | **Field n+5** | | **Field n+6** | | **Field n+7** | |
| | | | | | | | CRC | | **Hdr#** | | **Rsvd** | **Hub#** | | | **DLY** | | **DFR** | | **CRC** | |

*Table 13-34: Frames without additional fields other than in this table*

| Frame | Field1 | Field2 | Field3 | Field4 | Field5 | Field6 | F.7 | Fld.8 | Field9 |
|---|---|---|---|---|---|---|---|---|---|
| BCNT | BERC | EC | | | | | | | |
| BDAT | BDAT | | | | | | | | |
| BERC | BERC | | | | | | | | |
| BRST | COM | BRST | | | | | | | |
| Data Packet Payload | HPSTART | Data | CRC | DPPEND | | | | | |
| Data Packet Payload Abort | HPSTART | Data | CRC | DPPABORT | | | | | |
| IDLE | Idle | | | | | | | | |
| Link Command | LCSTART | SubType | CRC | SubType | CRC | | | | |
| SKIP | SKP | | | | | | | | |
| TS1 | COM | Rsvd | Train | Rsvd | Loop | Scramble | LL | Rsvd | TS1ID |
| TS2 | COM | Rsvd | Train | Rsvd | Loop | Scramble | LL | Rsvd | TS2ID |
| TSEQ | COM | Symbol | | | | | | | |

And finally the remaining types that are no specified USB 3.1 frames:

*Table 13-35: Other types*

| Type | Field1 | Field2 | Field3 | Field4 | Field5 | Field6 | Field7 | Field8 | Field9 |
|---|---|---|---|---|---|---|---|---|---|
| Error | CRC Error | Length Error | Unknown | Value out of range | Packet Incomplete | | | | |
| Unknown | ? | | | | | | | | |
| Unknown Device Notification | HPSTART | Type | Rsvd | Device Address | SubType | NotType | ? | CRC | **LCW** |
| Unknown Header Packet | HPSTART | Type | ? | ? | CRC | **LCW** | | | |
| Unknown LMP | HPSTART | Type | SubType | ? | ? | CRC | **LCW** | | |
| Unknown TP | HPSTART | Type | Rsvd | Device Address | SubType | ? | ? | CRC | **LCW** |
| **LCW** = Link Control Word, see Table 13-33, occupies the last 6 fields. | | | | | | | | | |

More information on the USB 3.1 protocol, including all specifications down to the field contents, is available in the "Universal Serial Bus 3.1 Specification" documentation. Refer to the online resources at www.usb.org.

## 13.19.2 USB 3.1 configuration

If you need information on how to get started with configuring the USB setup, see Chapter 13.19.2.3, "Configuring USB 3.1 signals", on page 847. Otherwise proceed with the configuration settings.

### 13.19.2.1 USB 3.1 configuration settings

Access: [Bus] > "USB 3.1" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.

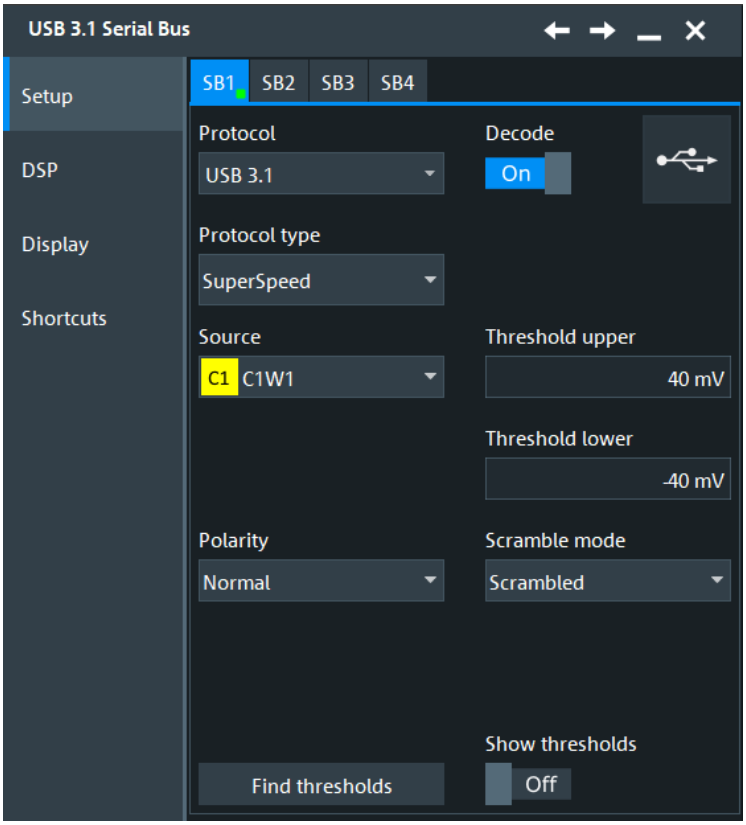



*Figure 13-33: Serial bus protocol configuration dialog*

See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Protocol type**
Selects the USB 3.1 protocol type.

"SuperSpeed"
USB 3.1 Gen 1 with 5 Gbit/s data rate.

"SuperSpeedPlus"
USB 3.1 Gen 2 with 10 Gbit/s data rate.

Remote command:
`BUS<m>:USBThree:TYPE` on page 2164

**Source**
Defines the source for the data signal. You can select analog channels "C1" to "C4", and also active math and reference waveforms as source. If you want to trigger on the serial bus, analog channel sources are required.

Remote command:
`BUS<m>:USBThree:SOURce` on page 2162

**Polarity**
Defines the signal polarity settings. Permitted selections are "Normal" and "Inverted".

Remote command:
`BUS<m>:USBThree:POLarity` on page 2162

**Thresholds**
Sets the threshold values for digitization of the analog signals.

The USB 3.1 standard uses 2-state signals with an upper and a lower voltage level. Permitted thresholds values are in the range of -2 V to +2 V, the default is -40 mV to +40 mV.

If the signal value on the line is higher than the upper threshold, the signal state is considered high. Otherwise, if the signal value is below the lower threshold, the signal state is considered low. The two threshold levels allow configuring a hysteresis setting.

There are two ways to set the thresholds:

| | |
|---|---|
| "Threshold" | Enter the upper and lower values directly into the fields. |
| "Find thresholds" | Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude. |

Remote command:
`BUS<m>:USBThree:THRHigh` on page 2163
`BUS<m>:USBThree:THRLow` on page 2163
`BUS<m>:SETReflevels` on page 1638

**Scramble Mode**
Selects if the data pattern is scrambled or unscrambled. Scrambling as specified in the standard minimizes EMI emissions, the unscrambled mode can simplify testing and debugging.

Remote command:
`BUS<m>:USBThree:SCRMode` on page 2163

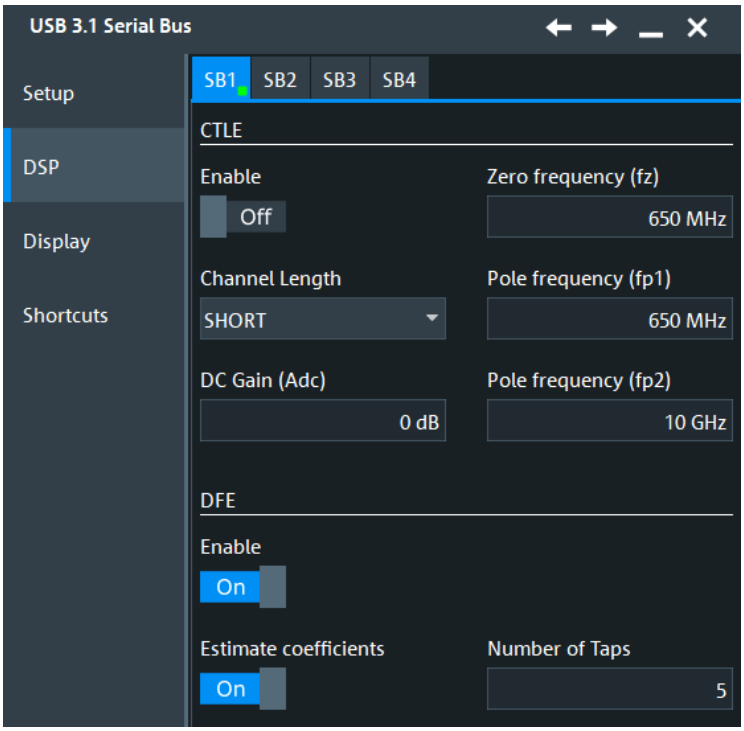
**Show thresholds**
If enabled, the threshold levels are displayed in the diagram.

Remote command:
`BUS<m>:THReshold` on page 1638

#### 13.19.2.2 USB 3.1 DSP settings

Access: [Bus] > "Setup" tab > "Protocol = USB 3.1" > "DSP" tab

At this tab, you can enable and define the equalization techniques used for the equalization at the reciever side of the signal.




**CTLE**
Continuous time linear equalization (CTLE) is an equalization that can be used for improving the signal performance. It boosts the high frequency noise and the signal. Thus it helps reduce the inter-symbol interference (ISI) on the reciever side.

**Enable ← CTLE**
Enables the CTLE equalization.

Remote command:
`BUS<m>:CTLE:ENBLe` on page 2164

**Zero frequency (fz) ← CTLE**
Sets the zero frequency for the transfer function of the CTLE algorithm.

Remote command:
`BUS<m>:CTLE:Z1FReq` on page 2165

**Pole frequency (wp1)/(wp2) ← CTLE**
Sets the pole 1/2 frequency for the transfer function of the CTLE algorithm.

Remote command:
`BUS<m>:CTLE:P1FReq` on page 2165
`BUS<m>:CTLE:P2FReq` on page 2165

**DC Gain ($A_{DC}$) ← CTLE**
Sets the adjustable DC gain. This value can be chosen according to the channel loss characteristics. For low-loss channels, you can set a value close to 0 dB.

Remote command:
BUS<m>:CTLE:DCGain on page 2164

**Channel Length ← CTLE**
Selects the USB Super Speed channel length.

Remote command:
BUS<m>:USBThree:PROTocol on page 2167

**DFE**
Defines the settings of the decision feedback equalization (DFE). DFE correction is needed when the channel is distosrted.

**Enable ← DFE**
Enables the use of the decision feedback equalization.

Remote command:
BUS<m>:USBThree:DFE:ENBLe on page 2166

**Estimate coefficients ← DFE**
Enables the use of estimate coefficients.

Remote command:
BUS<m>:USBThree:DFE:ESTCoeff on page 2166

**Coefficient ← DFE**
Sets the value of the coefficient.

Remote command:
BUS<m>:USBThree:DFE:COEFficient on page 2166

**Number of Taps ← DFE**
Sets a number of taps for the decision feedback filter.

Remote command:
BUS<m>:USBThree:DFE:TCONt on page 2167

#### 13.19.2.3 Configuring USB 3.1 signals

For configuration, assign the lines to the input channels and define the active states and the logical thresholds.

1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, set "Protocol"="USB3.1".
4. Select the "Protocol type".
5. Switch back to the "Setup" tab.

6. Set the logical thresholds: Either enter user-defined values directly in the "Threshold" fields, or set the thresholds to the middle reference levels with "Auto thresholds".
7. Enable "Decode".

For details on configuration settings, see Chapter 13.19.2.1, "USB 3.1 configuration settings", on page 844.

## 13.19.3 USB 3.1 trigger

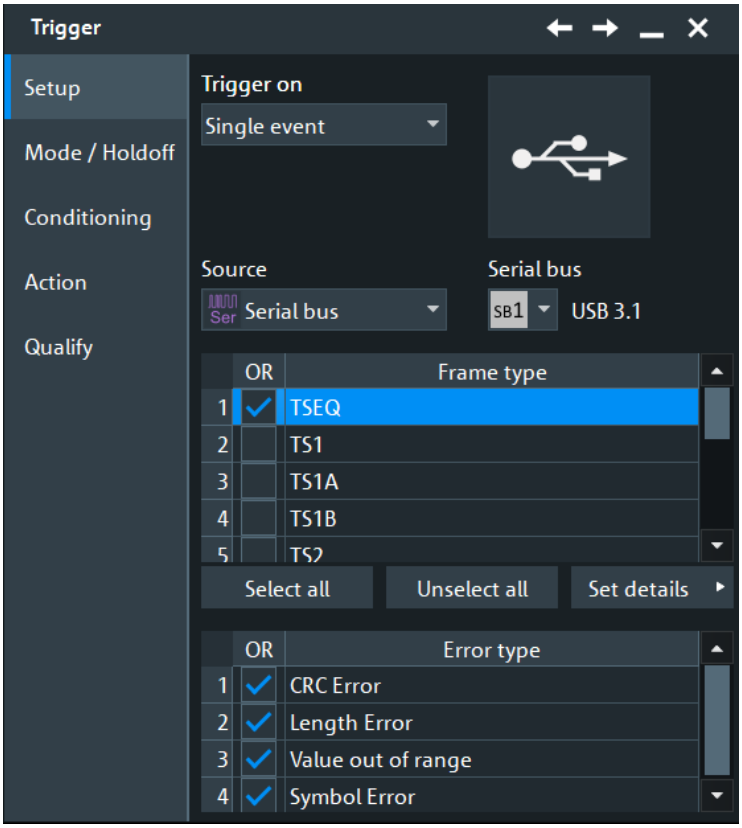
If you need information on how to get started with triggering on USB 3.1 signals, see Chapter 13.19.3.2, "Triggering on USB 3.1", on page 851. Otherwise proceed with the USB 3.1 trigger settings.

If the high-speed serial pattern trigger (option R&S RTP-K140/K141) is installed, USB 3.1 Gen 1 uses it for achieving faster triggering.

### 13.19.3.1 USB 3.1 trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"

In this section, all trigger settings for USB 3.1 are described. The user interface of the instrument guides you through the trigger setup.



Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Protocol type**
Selects the protocol type for the USB 3.1 analysis.

**Frame type**
Selects the frame type for USB 3.1 analysis. For the available frame types, see "Frame types" on page 837.

You can define individual checking parameters for the fields listed in the "Set details" dialog.

The trigger instant is the last criterion that is fulfilled.

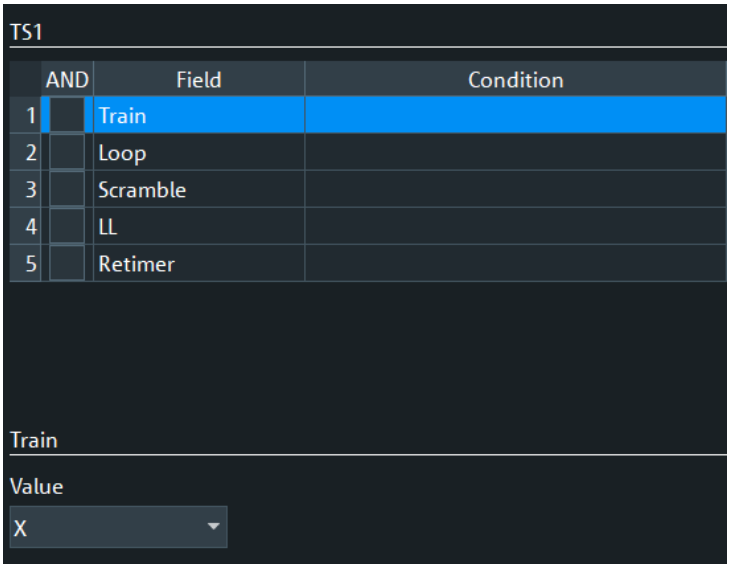
Remote command:
`TRIGger<m>:USBThree:FRENable` on page 2181
`TRIGger<m>:USBThree:FRAMe<n>:ENABle` on page 2181

**Error types: Item name, Enable**
The table lists the error types you can trigger on: "CRC Error", "Length Error", "Value out of range", "Symbol Error". Enable any error that you want to trigger on in the "OR" column.

Remote command:
`TRIGger<m>:USBThree:ERENable` on page 2181
`TRIGger<m>:USBThree:ERRor<n>:ENABle` on page 2181

**Set details**
Select the frame type first, and then open the "Set details" dialog.

In this dialog to define the individual parameters for the fields of the selected frame.




**Field name / Condition Summary / Enable ← Set details**
This table lists the field numbers and names in the selected frame together with a summary of the user settings of checking conditions for each field and a checkbox to enable the checking.

It also shows the user settings of checking conditions for each field and a checkbox to enable the checking.

Select a field in the table to specify the checking conditions for this field in the "Set details" dialog.

For an overview of frames and fields, see Frame types.

Remote command:
`TRIGger<m>:USBThree:FIENable` on page 2181
`TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:ENABle` on page 2181

**Data**
Defines for the selected field, how a data check is executed.

"Condition" Sets the comparison condition to a specific value or a range.

"Min" Specifies the value or sets the start value of a range.

"Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger<m>:USBThree:DOPerator` on page 2182
`TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DOPerator` on page 2182
`TRIGger<m>:USBThree:DMIN` on page 2183
`TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DMIN` on page 2183
`TRIGger<m>:USBThree:DMAX` on page 2183
`TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DMAX` on page 2183

**Data BitState**
Defines the data bit state to be checked for the selected field. Permitted bit states are "1", "0" or "X" (do not care).

Remote command:
`TRIGger<m>:USBThree:BIT` on page 2183
`TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:BIT` on page 2183

**Index**
The trigger index setup consists of a comparison condition and one or two address patterns.

| | |
|---|---|
| "Condition" | Sets the comparison condition to a specific value or a range. |
| "Min" | Specifies the value or sets the start value of a range. |
| "Max" | Sets the maximum value of a range for "Condition" = "In range"/"Out of range". |

Remote command:
`TRIGger<m>:USBThree:IOPerator` on page 2184
`TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IOPerator` on page 2184
`TRIGger<m>:USBThree:IMIN` on page 2184
`TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IMIN` on page 2184
`TRIGger<m>:USBThree:IMAX` on page 2185
`TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IMAX` on page 2185

#### 13.19.3.2 Triggering on USB 3.1

Prerequisite: A bus is configured for the USB 3.1 signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Select the "Protocol Type".
5. If you trigger on errors, enable the error types you want to find.
6. Select the frame types you want to trigger on.
7. For some frame types, you can define the frame fields.
   For information on the configuration settings, see Chapter 13.19.3.1, "USB 3.1 trigger settings", on page 848.

### 13.19.4 USB 3.1 decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".

2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode Results Table

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

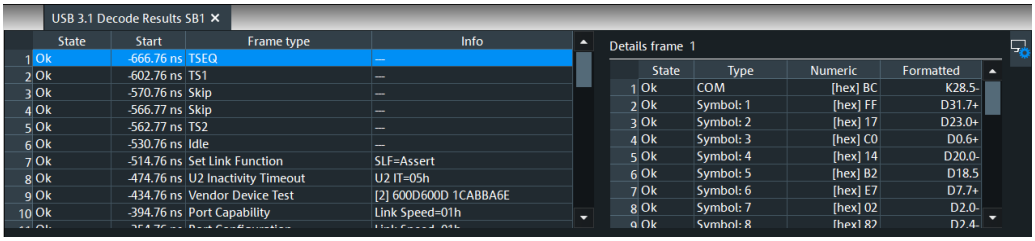



***Table 13-36: Content of the Decode results table***

| Column | Description |
|---|---|
| Frame | Frame number |
| State | Overall state of the frame, for example "OK" or "Incomplete" |
| Start | Start time of the frame |
| Frame Type | Type of the frame |
| Info | Specific frame information, for example result of the Link Speed Measurement = 5.0 Gb/s, or content of the Data Packet Payload |

Enable "Details" in the decode table to display a more detailed analysis of the selected frame.

***Table 13-37: Content of the Details frame table***

| Column | Description |
|---|---|
| State | Overall state of the field |
| Start | Start time of the field |
| Stop | Stop time of the field |
| Type | Type (name) of field |

| Column | Description |
|---|---|
| Numeric value | The numeric value of the field |
| Formatted | Formatted content of the field, for example HPSTART "F7" = end packet framing (EPF) |

**Remote commands**

Remote control commands for retrieving decode results are described in Chapter 24.17.20, "USB 3.1 (option R&S RTP-K61/K62)", on page 2162.

Remote command:

- `BUS<m>:USBThree:RESult:FCOunt?` on page 2186
- `BUS<m>:USBThree:RESult:FRAMe<n>:INFO?` on page 2186
- `BUS<m>:USBThree:RESult:FRAMe<n>:TYPE?` on page 2186
- `BUS<m>:USBThree:RESult:FRAMe<n>:STATe?` on page 2187
- `BUS<m>:USBThree:RESult:FRAMe<n>:STARt?` on page 2187
- `BUS<m>:USBThree:RESult:FRAMe<n>:STOP?` on page 2187
- `BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:TYPE?` on page 2188
- `BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:STATus?` on page 2188
- `BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:STARt?` on page 2188
- `BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:STOP?` on page 2189
- `BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:VAL?` on page 2189
- `BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:FVAL?` on page 2189

## 13.19.5 Search on decoded USB 3.1 data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search source to "SerBus" for the configured protocol.

For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

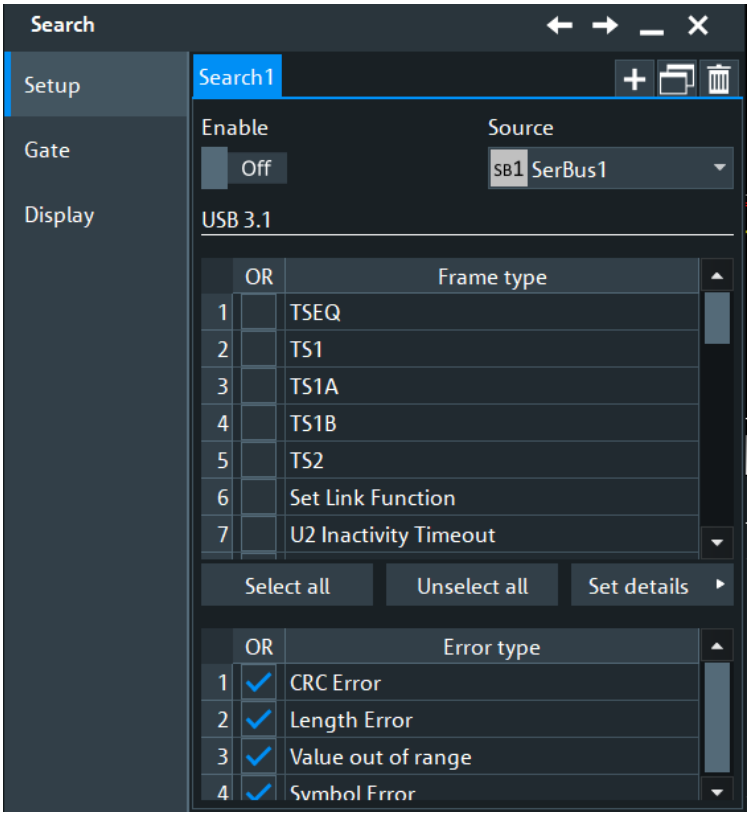
If you need information on how to get started with searching USB 3.1 data, see Chapter 13.19.5.3, "Searching USB 3.1 data", on page 855. Otherwise proceed with the USB 3.1 search setup.

### 13.19.5.1 USB 3.1 search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"

**Search criteria**

Use the "Search criteria" dialog to define the search type. If you search for frame content, define the frame type in which data is to be searched.




You can define individual search parameters for the fields of the selected frame in the table below the "Search criteria". To specify these parameters, select a field in the table and define the data and/or index operators and values, or the bit state.

For a description of how to set the search conditions, see Chapter 13.19.3.1, "USB 3.1 trigger settings", on page 848.

Remote command:

`SEARch:TRIGger:USBThree:ERENable` on page 2190
`SEARch:TRIGger:USBThree:ERRor<m>:ENABle` on page 2190
`SEARch:TRIGger:USBThree:FRENable` on page 2191
`SEARch:TRIGger:USBThree:FRAMe<m>:ENABle` on page 2191
`SEARch:TRIGger:USBThree:FIENable` on page 2191
`SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:ENABle` on page 2191
`SEARch:TRIGger:USBThree:DOPerator` on page 2192
`SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DOPerator` on page 2192
`SEARch:TRIGger:USBThree:DMIN` on page 2193
`SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DMIN` on page 2193
`SEARch:TRIGger:USBThree:DMAX` on page 2193
`SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DMAX` on page 2193
`SEARch:TRIGger:USBThree:BIT` on page 2194
`SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:BIT` on page 2194

SEARch:TRIGger:USBThree:IOPerator on page 2194
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IOPerator on page 2194
SEARch:TRIGger:USBThree:IMIN on page 2195
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMIN on page 2195
SEARch:TRIGger:USBThree:IMAX on page 2195
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMAX on page 2195

#### 13.19.5.2 USB 3.1 search results

To get search results, "Enable" the search in the "Control" section of the "Search Setup" dialog. You can minimize, shift or close the search dialog to better see the "Search Results" table.

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands:

- SEARch:RESult:USBThree:FCOunt? on page 2197
- SEARch:RESult:USBThree:FRAMe<m>:TYPE? on page 2196
- SEARch:RESult:USBThree:FRAMe<m>:INFO? on page 2196
- SEARch:RESult:USBThree:FRAMe<m>:STATe? on page 2197
- SEARch:RESult:USBThree:FRAMe<m>:STARt? on page 2197
- SEARch:RESult:USBThree:FRAMe<m>:STOP? on page 2198
- SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:TYPE? on page 2198
- SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:STATus? on page 2198
- SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:STARt? on page 2199
- SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:STOP? on page 2199
- SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:VAL? on page 2199
- SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:FVAL? on page 2200

#### 13.19.5.3 Searching USB 3.1 data

Prerequisite: A serial bus is configured for the USB 3.1 signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".

3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Tap "Source" and select the serial bus that is set to USB 3.1 (e.g. "SB1", unless already selected).

   The search dialog for USB3 protocol analysis is opened.
5. Specify search criteria according to Chapter 13.19.5.1, "USB 3.1 search setup", on page 853.
6. To acquire a waveform, press [Single].

   The R&S RTP performs a USB 3.1 decode according to the thresholds and protocol settings of the associated serial bus source (here in our example SB1).
7. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog:

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

## 13.20 USBPD (option R&S RTP-K63)

R&S RTP-K63 is a firmware option that enables the R&S RTP to analyze Universal Serial Bus Power Delivery (USBPD) signals.

For analysis, USBPD encoded signals can be triggered, decoded and searched.

This chapter describes:

- The USB power delivery protocol....856
- USBPD configuration....858
- USBPD trigger....861
- USBPD decode results....865
- Search on decoded USBPD data....867

### 13.20.1 The USB power delivery protocol

The requirements on the USB have changed in the last years with the need of providing power through the USB port additionally to the data transfer. The USBPD specification aims to define standard for optimizing the power usage through the USB for the needs of the users.

**USBPD characteristics**

Main characteristics of USBPD are:

- Power direction is not fixed
- Negotiation of required power amount between devices

- Alternate modes can be defined through vendor defined messages, which allows for USB connector pins to be used for purposes other than USB

### Message types

In the USBPD protocol, a power delivery connection can be made between a port that supplies power (source) and a port that consumes power (sink). They communicate with each other through messages. The USBPD specification defines three message types:

- Control messages: 16-bit messages used to control the messages between the port partners or transfer messages with no extra data. A control message consists of a message header and a CRC.
- Data messages: 48 bit to 240 bit messages used to transfer information between port partners. A data message consists of a message header and several data objects. The information that a data object carries is defined by the message type of the message header, see Table 13-38.
- Extended messages: can have a different length up to the defined maximum length of an extended message. It is used to transfer information between port partners. The information that the extended message carries is defined by the message type of the message header, see Table 13-38.

### Frame packet types

All frame types are listed below. The frames listed above consist of individual sets of fields. Some frames only contain one field, others are much longer.

The frame types are sorted according to the message type.

***Table 13-38: Frame packet types***

| SCPI | Description | Message type |
|---|---|---|
| ALRT | Alert message | Data |
| BATT | Battery status | Data |
| BIST | Built in self-test | Data |
| RQST | Request | Data |
| SINK | Sink capabilities message | Data |
| SRC | Source capabilities message | Data |
| VEND | Vendor defined message | Data |
| CTRL | Control | Control |
| DATA | Data | Data |
| LOWP | Low power | Low power |
| TEST | Test frame | Test |
| RESet | Reset frame | Reset |
| XBAC | Battery capabilities | Extended |
| XFRS | Firmware update response | Extended |

| SCPI | Description | Message type |
|---|---|---|
| XFRQ | Firmware update request | Extended |
| XGBC | Get battery cap | Extended |
| XGMI | Get manufacturer info | Extended |
| XGBS | Get battery status | Extended |
| XMFI | Manufacturer info | Extended |
| XMSG | Message | Extended |
| XSRC | Sources capabilities message | Extended |
| XSRS | Security response | Extended |
| XSRQ | Security request | Extended |
| XSTA | Status | Extended |

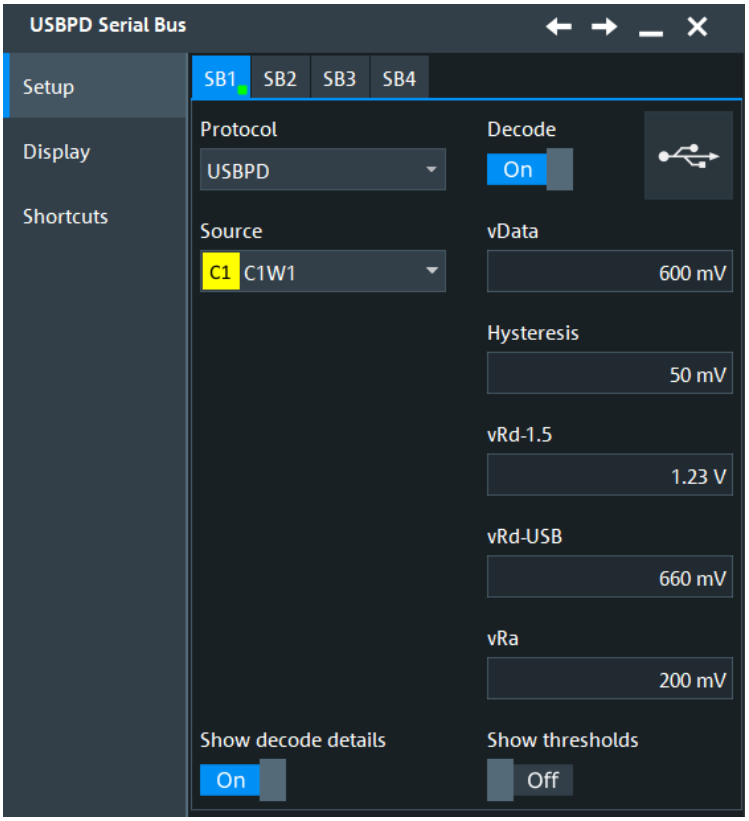
More information on the USBPD protocol, including all specifications down to the field contents, is available in the "Universal Serial Bus Power Delivery Specification" documentation. Refer to the online resources at www.usb.org.

## 13.20.2 USBPD configuration

### 13.20.2.1 USBPD configuration settings

Access: [Bus] > "USBPD" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Data**
Defines the source settings for the data signal. You can select analog channels "C1" to "C4", and also active math and reference waveforms as source. If you want to trigger on the serial bus, analog channel sources are required.

Remote command:
`BUS<m>:USBPd:SOURce` on page 2201

**vData**
Sets the threshold value of the data.

Remote command:
`BUS<m>:USBPd:THReshold` on page 2201

**Hysteresis**
Sets a value for the hysteresis of the data.

Remote command:
`BUS<m>:USBPd:HYSTeresis` on page 2201

**Current advertisment thresholds**
The signal level provides information about the current advertisement between the bursts. These thresholds determine the levels at which the current advertisement modes are defined.

"vRd-1.5" Sets the threshold at USB Type-C current of 1.5 A.

"vRd-USB" Sets the threshold at default USB Type-C current.

"vRa" Sets the threshold for the low current.

Remote command:
BUS<m>:USBPd:THRBottom on page 2202
BUS<m>:USBPd:THRMid on page 2202
BUS<m>:USBPd:THRTop on page 2202

**Show decode detail**
If enabled, the data words are broken down into subframes. If not enabled the data words are displayed as 32-bit data words.

If the "Show decode detail" is enabled, you cannot do a search and trigger on USBPD frames.

Remote command:
BUS<m>:USBPd:DETail on page 2201

**Show thresholds**
If enabled, the threshold levels are displayed in the diagram.

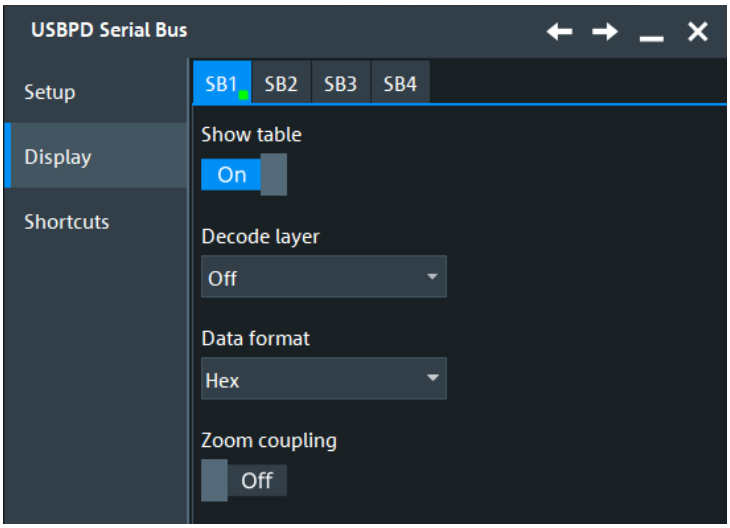
Remote command:
BUS<m>:THReshold on page 1638

### 13.20.2.2 Display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

To enhance the decode possibilities of the USBPD protocol, you can use an additional setting in the "Display" tab: "Decode layer".




**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the end results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

Available are:
- "Edges"
- "Bits"
- "4b5b Symbols"

#### 13.20.2.3 Configuring the USBPD signals

For configuration, assign the lines to the input channels and define the logical thresholds and the hysteresis.

1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, set "Protocol"="USBPD".
4. Select the data source.
5. Enter the "Thresholds".
6. Enable "Decode".

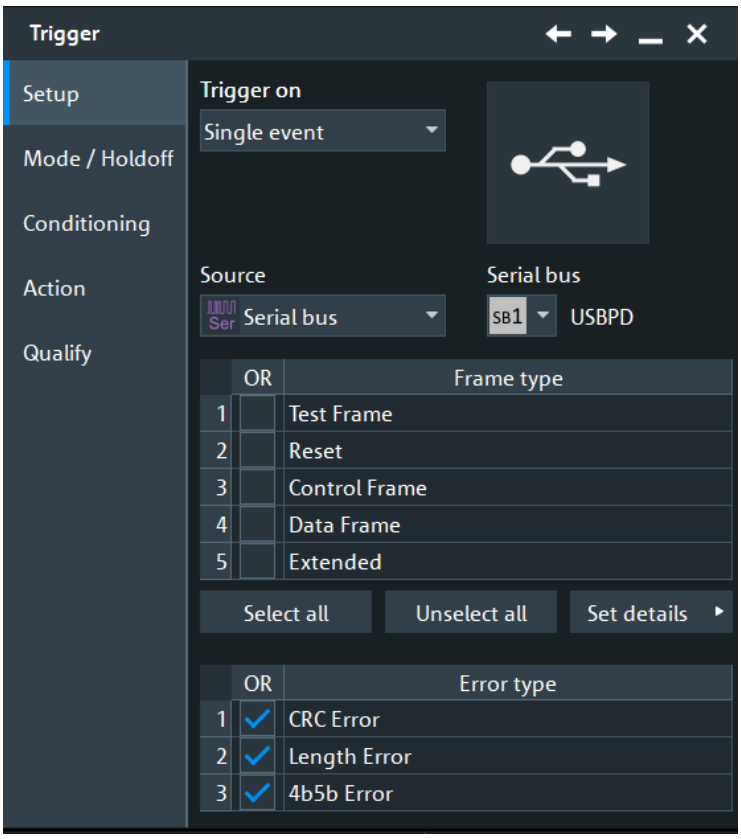
### 13.20.3 USBPD trigger

If you need information on how to get started with triggering on USBPD signals, see Chapter 13.20.3.2, "Triggering on USBPD", on page 864. Otherwise proceed with the USBPD trigger settings.

#### 13.20.3.1 USBPD trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"

In this section, all trigger settings for USBPD are described. The user interface of the instrument guides you through the trigger setup.




Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Frame Type**
Selects the frame type for the USBPD analysis. For the available packet types, see "Frame packet types" on page 857.

You can define individual checking parameters for the fields listed of the frame.

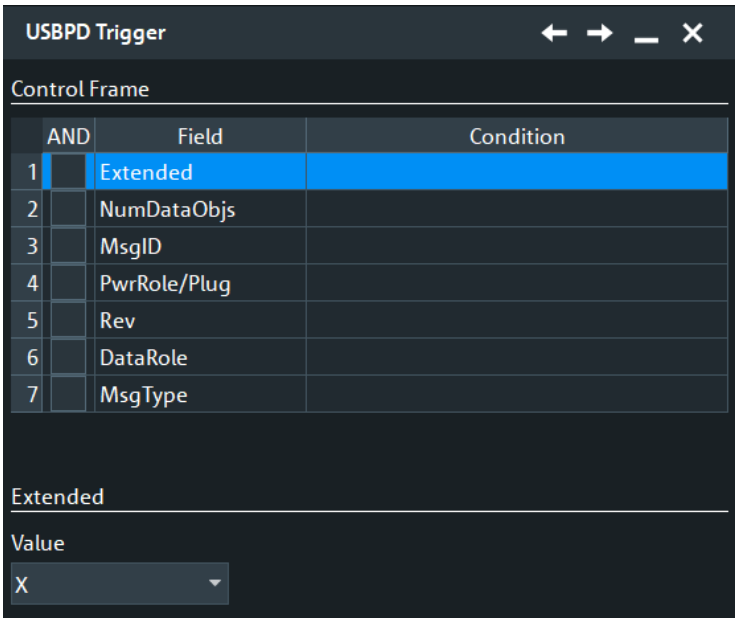
To specify these parameters, select a field from this list and open "Set details". You can define the data and/or index operators and values, or the bit state.

The trigger instant is the last criterion that is fulfilled.

Remote command:
`TRIGger<m>:USBPd:FRENable` on page 2205
`TRIGger<m>:USBPd:FRAMe<n>:ENABle` on page 2205

**Set details**
Select the frame type first, and then open the "Set details" dialog.

In this dialog to define the individual parameters for the fields of the selected frame.




**Enable/ Field name/ Condition ← Set details**
This table lists the field numbers and names in the selected frame together with a summary of the user settings of checking conditions for each field and a checkbox to enable the checking.

Select a field in the table to specify the checking conditions for this field in the "Data", "BitState" and/or "Index" dialog (whichever applies). The condition is only applied, and the "Condition Summary" is only shown in the table, if "Enable" is checked.

For an overview of frames and fields, see "Frame packet types" on page 857.

Remote command:
`TRIGger<m>:USBPd:FIENable` on page 2205
`TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:ENABle` on page 2205

**Error > Enable /Item**
The table lists the error types you can trigger on: "CRC Error", "Length Error", "4b5b Error". Enable any error that you want to trigger on in the "OR" column.

Remote command:
`TRIGger<m>:USBPd:ERENable` on page 2205
`TRIGger<m>:USBPd:ERRor<n>:ENABle` on page 2205

**Data**
Sets the data to be triggered on. The trigger data setup consists of a comparison condition and one or two data patterns.

"Condition" Sets the comparison condition to a specific value or a range.

"Min" Specifies the value or sets the start value of a range.

"Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
TRIGger<m>:USBPd:DOPerator on page 2207
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DOPerator on page 2207
TRIGger<m>:USBPd:DMAX on page 2207
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DMAX on page 2207
TRIGger<m>:USBPd:DMIN on page 2207
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DMIN on page 2207

**Data BitState**
Defines the bit state to be checked for the selected field. Permitted bit states are "1", "0" or "X" (do not care).

Remote command:
TRIGger<m>:USBPd:BIT on page 2206
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:BIT on page 2206

**Index**
Defines for the selected field, how an index check is executed.

"Condition" Defining a specific index or an index range requires to set the operator to one of the following conditions: equal, in range.

"Min" Specifies the index or sets the start value of an index range.

"Max" Sets the the end value of an index range, if "Condition" is set to `INRange`.

Remote command:
TRIGger<m>:USBPd:IOPerator on page 2209
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IOPerator on page 2209
TRIGger<m>:USBPd:IMAX on page 2208
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IMAX on page 2208
TRIGger<m>:USBPd:IMIN on page 2208
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IMIN on page 2208

### 13.20.3.2 Triggering on USBPD

Prerequisite: A bus is configured for the USBPD signal to be analyzed.

In order to be able to trigger on an USBPD data, the "Show decode detail" field in the "Configuration" tab of the protocol setup should be disabled.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Tap "Serial bus" and select the serial bus that is set to USBPD.

5. If you trigger on errors, enable the error types you want to find.
6. Select the frame types you want to trigger on.
7. For some frame types, you can define the frame fields.
   For information on how to proceed with the configuration settings, see Chapter 13.20.3.1, "USBPD trigger settings", on page 861.

### 13.20.4 USBPD decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Serial Bus" dialog > "Setup" tab, enable "Decode".
2. In the "Serial Bus" dialog > "Display" tab, select additional result display settings: "Show table" and "Show binary".
3. If required, enable "Zoom coupling"

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

**Decode Results Table**

The example in Figure 13-34 shows a simulated USBPD message. Among the long line of decoded frames, the zoom has selected a "Control" frame.

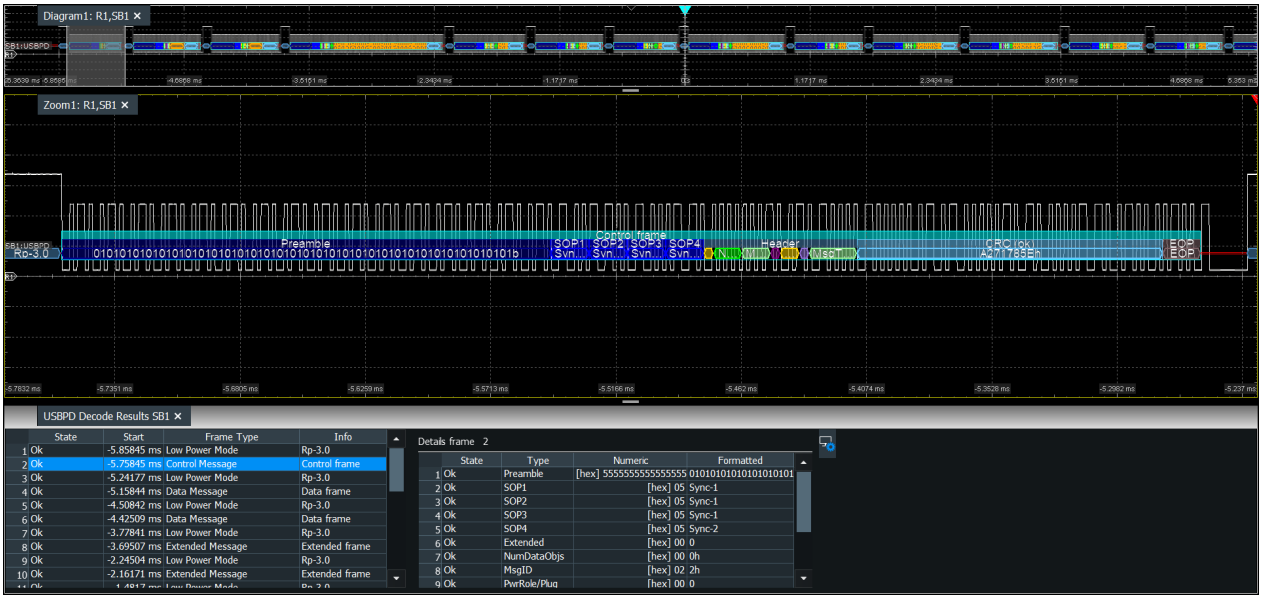



*Figure 13-34: USBPD protocol: decoded signal*

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

*Table 13-39: Content of the Decode results table*

| Column | Description |
|---|---|
| State | Overall state of the frame |
| Start | Time of frame start |
| Frame Type | Type of the frame |
| Info | Information about the frame |

*Table 13-40: Content of the Details frame table*

| Column | Description |
|---|---|
| State | Overall state of the field |
| Start | Start time of the field |
| Type | Type (name) of field |
| Numeric value | The numeric value of the field |
| Formatted | Formatted content of the field |

### Remote commands

Remote commands are described in Chapter 24.17.21.3, "Decode results", on page 2209.

## 13.20.5 Search on decoded USBPD data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search source to "SerBus" for the configured protocol.

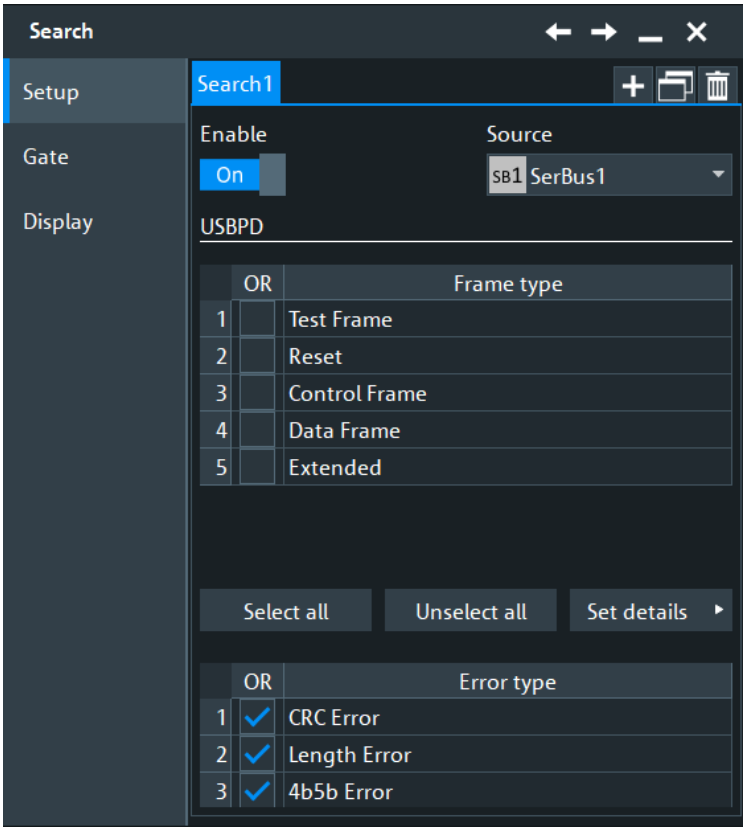
For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

If you need information on how to get started with searching USBPD data, see Chapter 13.20.5.3, "Searching USBPD data", on page 869. Otherwise proceed with the USBPD search setup.

### 13.20.5.1 USBPD search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"

**Search**
In the "Search" dialog to define the frame type in which data is to be searched.



You can define individual search parameters for individual fields in the "Set details" dialog. To specify these parameters, select a field in this table and define the data and/or index operators and values, or the bit state.

For a description of how to set the search conditions, see Chapter 13.20.3.1, "USBPD trigger settings", on page 861.

Remote command:
SEARch:TRIGger:USBPd:ERENable on page 2215
SEARch:TRIGger:USBPd:ERRor<m>:ENABle on page 2215
SEARch:TRIGger:USBPd:BIT on page 2217
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:BIT on page 2217
SEARch:TRIGger:USBPd:DMAX on page 2217
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMAX on page 2217
SEARch:TRIGger:USBPd:DMIN on page 2218
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMIN on page 2218
SEARch:TRIGger:USBPd:DOPerator on page 2218
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DOPerator on page 2218
SEARch:TRIGger:USBPd:IMAX on page 2219
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMAX on page 2219
SEARch:TRIGger:USBPd:IMIN on page 2219
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMIN on page 2219
SEARch:TRIGger:USBPd:IOPerator on page 2220
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IOPerator on page 2220

### 13.20.5.2 USBPD search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog to better see the "Search Results" table.

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands:

- SEARch:RESult:USBPd:FCOunt? on page 2220
- SEARch:RESult:USBPd:FRAMe<m>:DATA? on page 2221
- SEARch:RESult:USBPd:FRAMe<m>:TYPE? on page 2225
- SEARch:RESult:USBPd:FRAMe<m>:INFO? on page 2223
- SEARch:RESult:USBPd:FRAMe<m>:STATe? on page 2224
- SEARch:RESult:USBPd:FRAMe<m>:STARt? on page 2224
- SEARch:RESult:USBPd:FRAMe<m>:STOP? on page 2225
- SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:TYPE? on page 2223

- SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:STATus? on page 2221
- SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:STARt? on page 2222
- SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:STOP? on page 2222
- SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:VAL? on page 2223
- SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:FVAL? on page 2221

#### 13.20.5.3 Searching USBPD data

Prerequisite: A serial bus is configured for the USBPD signal to be decoded and analyzed.

To be able to search for a USBPD data, the "Show decode detail" field in the "Configuration" tab of the protocol setup should be disabled. If "Show decode detail" is enabled, the "Enable" search button is disabled.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".
3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Tap "Source" and select the serial bus that is set to USBPD (e.g. "SerBus1", unless already selected).

   The search dialog for USBPD protocol analysis is opened.
5. Specify search criteria according to Chapter 13.20.5.1, "USBPD search setup", on page 867.
6. To acquire a waveform, press [Single].

   The R&S RTP performs a USBPD decode according to the thresholds and protocol settings of the associated serial bus source (here in our example SB1).
7. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog:

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

## 13.21 SpaceWire (option R&S RTP-K65)

The SpaceWire is a communication network standard used for spacecrafts. It is based on the IEEE 1355 standard of comunications and coordinated by the European Space Agency (ESA).

- SpaceWire basics ..... 870
- SpaceWire configuration ..... 871
- SpaceWire trigger ..... 874
- SpaceWire decode results ..... 877
- Search on decoded SpaceWire data ..... 878


### 13.21.1 SpaceWire basics

The SpaceWire links are a Point-to-point (P2P) connection between a node and another node or a rooter. The link is full-duplex bidirectional serial data link.

The SpaceWire has two types of characters:

- Data characters containing a parity bit, a data control flag and eight bits of data.

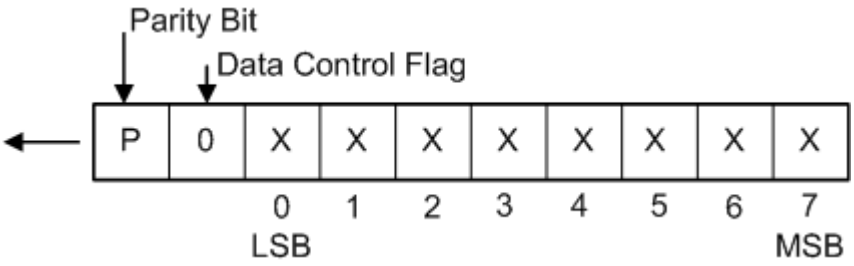



*Figure 13-35: SpaceWire data characters*

- Control characters containing a parity-bit, a data-control flag and the 2-bit control code. The data control flag is set to 1 and indicates that this is a control character.

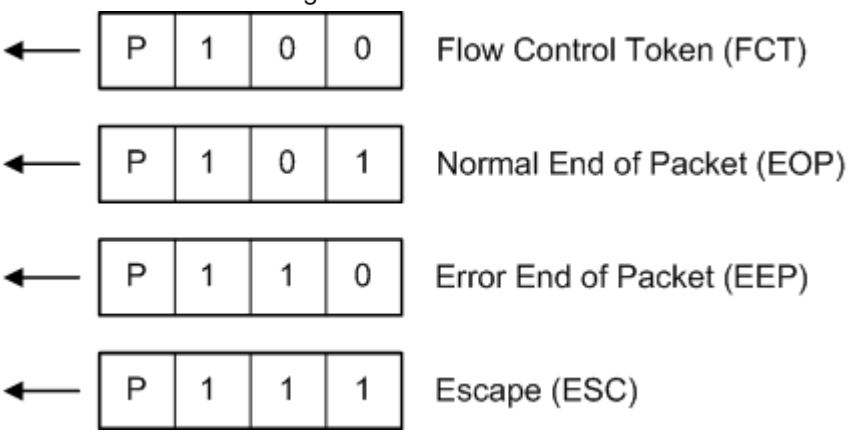



*Figure 13-36: SpaceWire control characters*

Also there are two control codes:

- NULL code consisting of an Escape (ESC) and a Flow Control Token (FCT)
- Time Code consisting of an ESC followed by a single data character

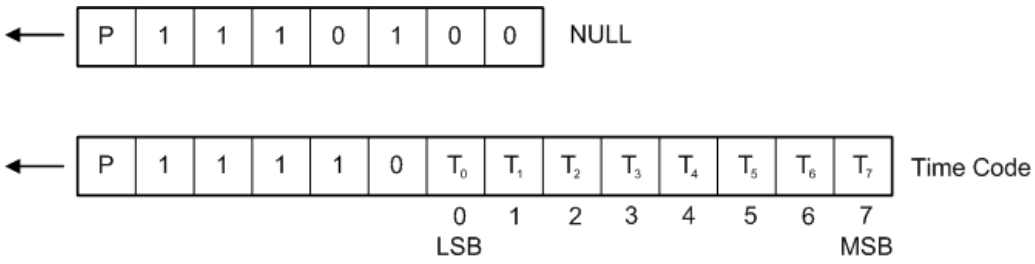


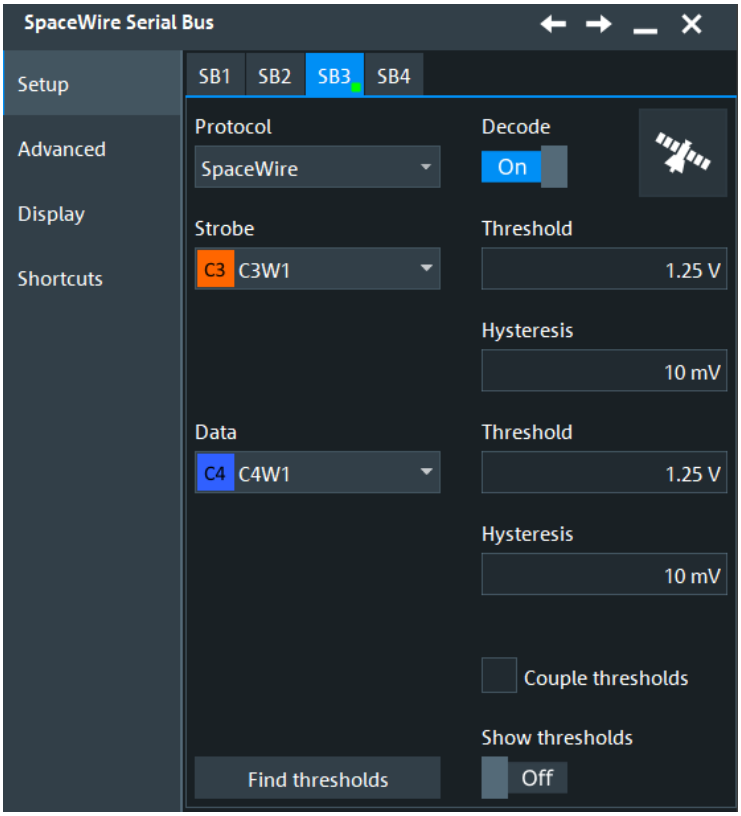

*Figure 13-37: SpaceWire control codes*

## 13.21.2 SpaceWire configuration

### 13.21.2.1 SpaceWire configuration settings

Access: [Bus] > "SpaceWire" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Strobe**
Selects the source for the strobe signal.

Remote command:
BUS<m>:SWIRe:STRBe:SOURce on page 2228

**Data**
Selects the source for the data signal.

Remote command:
BUS<m>:SWIRe:DATA:SOURce on page 2226

**Threshold setup**
Sets the threshold value for the strobe/data signal.

"Threshold" Enter individual values for each line directly in the fields.

"Couple thresholds" Enables coupling, i.e. the same threshold and hysteresis value is used for the strobe and data signal.

Remote command:
BUS<m>:SWIRe:COUPling on page 2228
BUS<m>:SWIRe:STRBe:THReshold on page 2228
BUS<m>:SWIRe:DATA:THReshold on page 2227
BUS<m>:SWIRe:PRESet on page 2229
BUS<m>:SETReflevels on page 1638

**Hysteresis**
Sets a value for the hysteresis of the strobe/data signal.

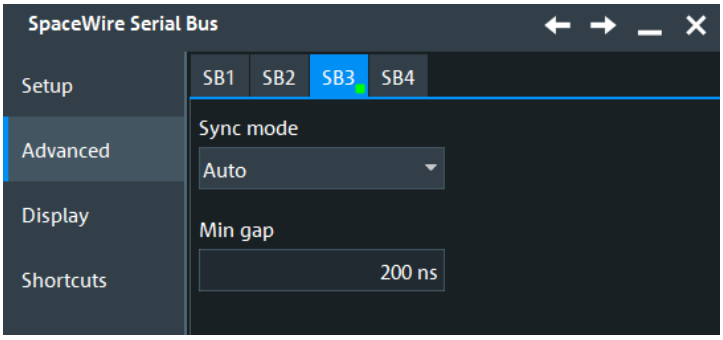
Remote command:
BUS<m>:SWIRe:STRBe:HYSTeresis on page 2227
BUS<m>:SWIRe:DATA:HYSTeresis on page 2226

**Show thresholds**
If enabled, the threshold levels are displayed in the diagram.

Remote command:
BUS<m>:THReshold on page 1638

### 13.21.2.2 Advanced settings

Access: [Bus] > "Advanced" tab.



**Bit Position**
Sets the bit position, the align position for the manual synchronization mode. This can be useful when parity errors exist in the signal, and parity check is the main indicator for the decoder to do packet alignment.

Remote command:
`BUS<m>:SWIRe:BPOSition` on page 2226

**Min Gap**
SpaceWire can have idle phases where strobe and data signals are not being sent. These "gaps" are identified to resume decoding after this idle time.

"Min Gap" sets the minimum duration of a gap. Any inactivity greater than this time is interpreted as a gap and lead to a resynchronization to the signal.

Remote command:
`BUS<m>:SWIRe:MGAP` on page 2227

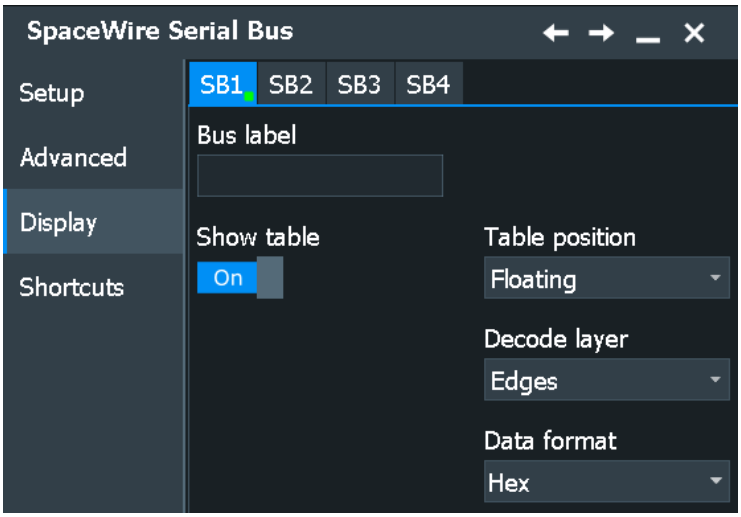
**Sync Settings**
Sets the mode for the synchronization of the signal. In the auto mode, the decoder automatically does the packet align. In the manual mode, you can set the align point manually with the "Bit Position" setting.

Remote command:
`BUS<m>:SWIRe:SYSLect` on page 2228

### 13.21.2.3 Display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.




**Decode Layer**
Selects the decode layer.

Decoding is performed in several steps, and the end results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

Available are:

- "Off"
- "Edges"
- "Binary"

#### 13.21.2.4 Configuring the SpaceWire signals

For configuration, assign the lines to the input channels and define the logical thresholds and the hysteresis.

1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, set "Protocol"="SpaceWire".
4. Select the source and polarity for the strobe and data signals.
5. Enter the "Threshold" and the "Hysteresis" for the strobe and data signals.
6. Set the "Sync Settings" and the "Bit Position" if necessary.
7. Enable "Decode".

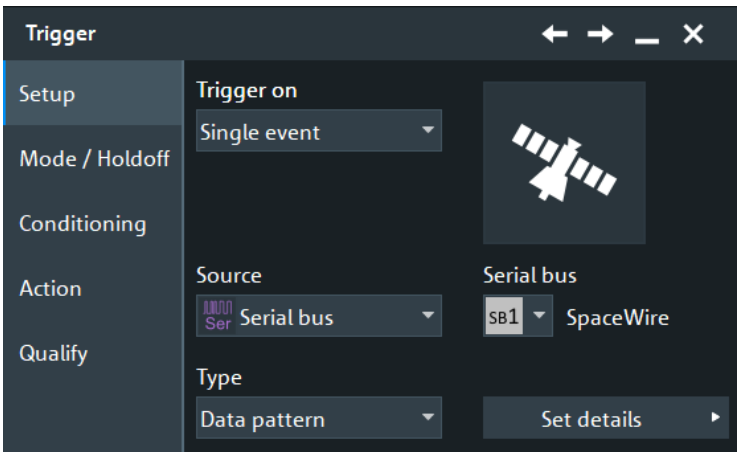
### 13.21.3 SpaceWire trigger

#### 13.21.3.1 SpaceWire trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"

Make sure that:

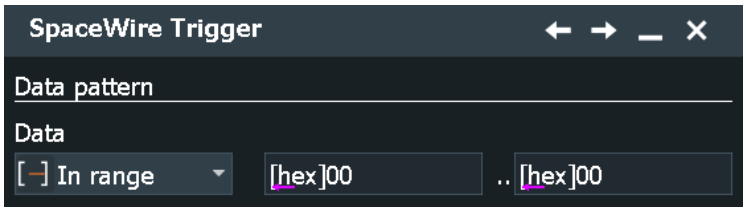
- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.




**Type**
Selects the trigger type for the SpaceWire analysis.

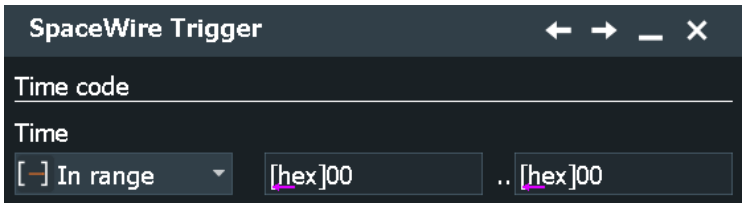
"Control frame" Sets the trigger to the selected control type frame.

"Data Pattern" Sets the trigger to a defined data pattern or pattern range.




"NULL frame" Sets the trigger to a null frame, a frame without usable data.

"Time Code" Sets the trigger to a time-code control code. You can define the data pattern of the time code to be triggered on.




"Errors" Triggers on an enabled error type.

Remote command:
`TRIGger<m>:SWIRe:TYPE` on page 2232

**Trigger Control Type**
Triggers on a specific control type character.

"ANY" Any control type character

"FCT" Flow control token character

"EOP" Normal end of packet character

"EEP" Error end of packet character

Remote command:
`TRIGger<m>:SWIRe:CTYPe` on page 2229

**Data (Time Code)**
Sets the specified data type for the time code to be triggered on. The trigger data setup consists of a comparison condition and one or two data patterns.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"Data Min/ Data" Specifies the value or sets the start value of a range.

"Data Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger<m>:SWIRe:TIME:CONDition` on page 2231
`TRIGger<m>:SWIRe:TIME:MAX` on page 2232
`TRIGger<m>:SWIRe:TIME:MIN` on page 2232

**Data (Data Pattern)**
Sets the specified data type for the data pattern to be triggered on. The data type setup consists of the condition and one or two data patterns.

In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Condition" Sets the comparison condition to a specific value or a range.

"Data Min/ Data" Specifies the value or sets the start value of a range.

"Data Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
`TRIGger<m>:SWIRe:DATA:CONDition` on page 2230
`TRIGger<m>:SWIRe:DATA:MAX` on page 2230
`TRIGger<m>:SWIRe:DATA:MIN` on page 2230

**Parity Error**
Checks the parity of every frame and triggers if the parity is even.

Remote command:
`TRIGger<m>:SWIRe:ERRor:PARity` on page 2231

**ESC Error**
Triggers on an escape error.

Remote command:
`TRIGger<m>:SWIRe:ERRor:ESC` on page 2231

### 13.21.3.2 Triggering on SpaceWire

Prerequisite: A bus is configured for the SpaceWire signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and select the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Select the serial bus that is set to SpaceWire.
5. Select the "Trigger Type" to be used for SpaceWire protocol analysis.
6. To refine the trigger settings, configure additional settings, which are available for some trigger types.
   For details, see Chapter 13.21.3.1, "SpaceWire trigger settings", on page 874.

## 13.21.4 SpaceWire decode results

When the configuration of the serial bus is complete, the signal can be decoded:

1. In the "Protocol" dialog > "Setup" tab, enable "Decode".
2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode results table

The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

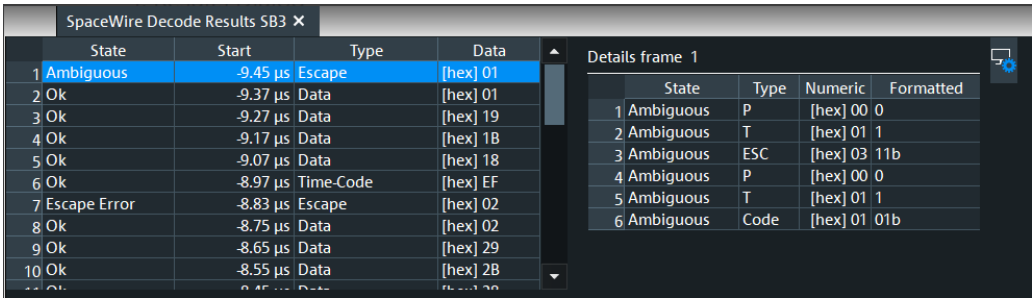


*Table 13-41: Content of the Decode results table*

| Column | Description |
|---|---|
| State | Overall state of the frame |
| Start | Time of frame start in relation to the trigger point |
| Stop | Time of frame stop in relation to the trigger point |
| Type | Frame type |
| Data | Data value |

*Table 13-42: Content of the Details frame table*

| Column | Description |
|---|---|
| State | State of the field |
| Type | Field type |
| Numeric | Hexadecimal value of the field |
| Formatted | Formatted content of the field |

**Tools**

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

**Remote commands**

Remote commands to retrieve decode results are described in Chapter 24.17.22.3, "Decode results", on page 2232.

## 13.21.5 Search on decoded SpaceWire data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

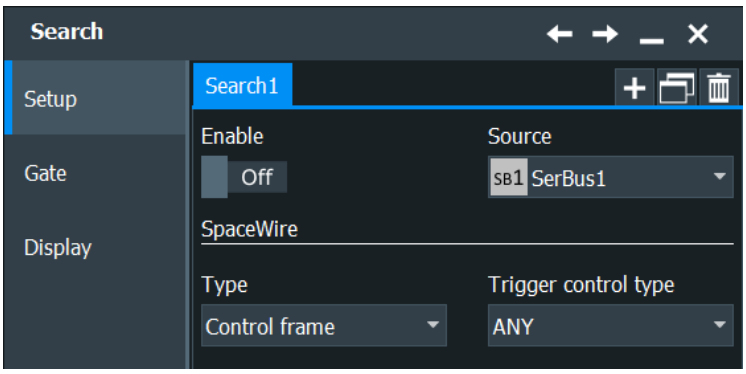
Before you can start the search, you have to configure the bus correctly and acquire decoded data.

To search on decoded data, set the search source to "SerBus" for the configured protocol.

For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

### 13.21.5.1 SpaceWire search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"




**Type**
Sets the search type for the SpaceWire analysis.

"Control frame" Searches for the selected control type frame.

"Data Pattern" Searches for a defined data pattern or pattern range.

"NULL frame" Searches for a null frame, a frame without usable data.

"Time Code" Searches for a time-code control code. You can define the data pattern of the time code to be searched for.

"ERRORs" Searches for an enabled error type.

Remote command:
`SEARch:TRIGger:SWIRe:TYPE` on page 2241

**Trigger Control Type**
Searches for a specific control type character.

"ANY" Any control type character

"FCT" Flow Control Token character

"EOP" Normal End of Packet character

"EEP" Error End of Packet character

Remote command:
`SEARch:TRIGger:SWIRe:CTYPe` on page 2238

**Data (Time Code)**
Sets the specified data type for the time code to be searched for. The data type setup consists of the condition and one or two data patterns.

"Condition" Defines the operator to set a specific data type ("Equal" or "Not equal") or a data type range.

"Data Min/ Data" Defines the bit pattern of the data pattern.
In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
SEARch:TRIGger:SWIRe:TIME:CONDition on page 2240
SEARch:TRIGger:SWIRe:TIME:MAX on page 2240
SEARch:TRIGger:SWIRe:TIME:MIN on page 2241

**Data (Data Pattern)**
Sets the specified data type for the data pattern to be searched for. The data type setup consists of the condition and one or two data patterns.

"Condition" Defines the operator to set a specific data type ("Equal" or "Not equal") or a data type range.

"Data Min/ Data" Defines the bit pattern of the data pattern.
In binary format, use the following characters: 1; 0; or X (do not care). To enter the pattern in other formats use the bit pattern editor, see Chapter 13.1.7, "Bit pattern editor", on page 524.

"Data Max" The second data pattern is required to specify a range with conditions "In range" and "Out of range".

Remote command:
SEARch:TRIGger:SWIRe:DATA:CONDition on page 2238
SEARch:TRIGger:SWIRe:DATA:MAX on page 2239
SEARch:TRIGger:SWIRe:DATA:MIN on page 2239

**Parity Error**
Checks the parity of every frame and searches for even parity.

Remote command:
SEARch:TRIGger:SWIRe:ERRor:PARity on page 2239

**ESC Error**
Searches for an escape error.

Remote command:
SEARch:TRIGger:SWIRe:ERRor:ESC on page 2239

### 13.21.5.2 SpaceWire search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

Remote commands:

- SEARch:RESult:SWIRe:FCOunt? on page 2241
- SEARch:RESult:SWIRe:FRAMe<m>:DATA? on page 2242
- SEARch:RESult:SWIRe:FRAMe<m>:STARt? on page 2242

- SEARch:RESult:SWIRe:FRAMe<m>:STATe? on page 2242
- SEARch:RESult:SWIRe:FRAMe<m>:STOP? on page 2243
- SEARch:RESult:SWIRe:FRAMe<m>:TYPE? on page 2243

#### 13.21.5.3 Searching SpaceWire

Prerequisite: A serial bus is configured for the SpaceWire signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".
3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Tap "Source" and select the serial bus that is set to SpaceWire (e.g. "SB1", unless already selected).
5. Specify search criteria according to Chapter 13.21.5.1, "SpaceWire search setup", on page 878.
6. To acquire a waveform, press [Single].

   The R&S RTP performs a SpaceWire decode according to the thresholds and protocol settings of the associated serial bus source.
7. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog:

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

## 13.22 PCIe (option R&S RTP-K72/K73)

R&S RTP-K72 is a firmware option that enables the R&S RTP to analyze Peripheral Component Interconnect Express (PCIe) signals encoded by the PCIe standard, generation 1 and 2.

With option R&S RTP-K73, you can also analyze PCIe standard, generation 3 signals.

For analysis, PCIe encoded signals can be triggered, decoded and searched.

- The PCIe protocol ..... 882
- PCIe configuration ..... 884
- PCIe digital signal processing (DSP) settings ..... 888

- PCIe trigger....892
- PCIe Gen 1/2 decode results....903
- PCIe Gen 1/2 search....904

## 13.22.1 The PCIe protocol

The PCIe is a high-speed serial computer expansion bus standard.

The communication between two PCIe devices is performed through logical connections called links. Each link consists of several lanes. The lanes contain one differential signaling pair for receiving data and the other for transmitting it.

### PCIe logical layers

The PCIe has three logical layers:

- Transaction layer: assembles and disassembles transaction layer packets (TLPs). TLP transfer information like read and write and some event types.
- Data link layer: used for link management and data integrity. Error detection and correction are also done in this layer. The data link layer produces data link layer packets (DLLP).
- Physical layer: includes the circuitry for the interface operation.

### PCIe Packet Types

The communication between the layers is done through packets. All packet types that can be decoded by the R&S RTP are listed below.

| SCPI name | Description | Type |
|---|---|---|
| MRD32 / MRD64 | Memory Read Request for 32 bit/ 64-bit addresssing packet format | TLP |
| MRDLK32 / MRDLK64 | Memory Read Request-Locked for 32 bit/ 64-bit addresssing packet format | TLP |
| MWR32 / MWR64 | Memory Write Request for 32 bit/ 64-bit addresssing packet format | TLP |
| IORD | I/O Read Request | TLP |
| IOWR | I/O Write Request | TLP |
| CFGRD0 / CFGRD1 | Configuration Read Type 0/1 | TLP |
| CFGWR0 / CFGWR1 | Configuration Write Type 0/1 | TLP |
| TCfgRd / TCfgWr | Deprecated TLP Type | TLP |
| MSG | Message Request | TLP |
| MSGD | Message Request with data payload | TLP |
| CPL | Completion without Data | TLP |
| CPLD | Completion with Data | TLP |

| SCPI name | Description | Type |
|---|---|---|
| CPLLK | Completion for Locked Memory Read without Data | TLP |
| CPLDLK | Completion for Locked Memory Read | TLP |
| FA32 / FA64 | Fetch and Add AtomicOp Request for 32 bit/ 64-bit addresssing packet format | TLP |
| SWP32 / SWP64 | Unconditional Swap AtomicOp Request for 32 bit/ 64-bit addresssing packet format | TLP |
| CAS32 / CAS64 | Compare and Swap AtomicOp Request for 32 bit/ 64-bit addresssing packet format | TLP |
| LPRFX | Local TLP Prefix | TLP |
| EPRFX | End-End TLP Prefix | TLP |
| EDB | | TLP |
| ACK | Acknowledgement | DLLP |
| NAK | Negative acknowledgment | DLLP |
| PMEL1 / PMEL23 | Power management Enter L1/ L23 | DLLP |
| PMASRL1 | Power management Active State Request L1 | DLLP |
| PMRA | Power management request acknowledgment | DLLP |
| VENDS | Vendor Specific | DLLP |
| IFC1P / IFC2P | Initialization flow control 1/2 posted requests | DLLP |
| IFC1NP / IFC2NP | Initialization flow control 1/2 non posted requests | DLLP |
| IFC1CPL / IFC2CPL | Initialization flow control 1/2 completions | DLLP |
| UPDFCP | Update flow control 1/2 posted requests | DLLP |
| UPDFCNP | Update flow control non posted requests | DLLP |
| UPDFCCPL | Update flow control completions | DLLP |
| SKPOS | SKP ordered set | Ordered Set |
| TS1OS / TS2OS | Training sequence 1/2 ordered set | Ordered Set |
| FTSOS | Fast training sequence ordered set | Ordered Set |
| EIOS | Electrical idle ordered set | Ordered Set |
| EIEOS | Electrical idle exit ordered set | Ordered Set |
| COMPL | Compliance pattern | Ordered Set |
| MCOMPL | Modified Compliance pattern | Ordered Set |
| DCOMPL | Delay Compliance | Ordered Set |

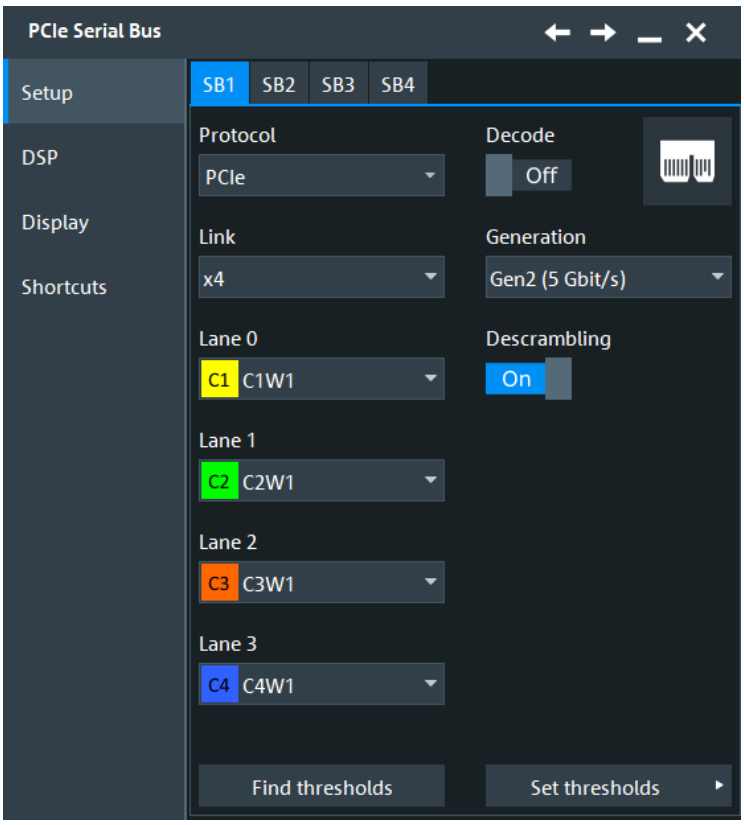
| SCPI name | Description | Type |
|---|---|---|
| SDS | Start of data sequence OS | Ordered Set |
| MRIFC1 / MRIFC2 | Multi-root initialization flow control 1/2 | Multi-root DLLP |
| MRUPDFC | Multi-root update flow control | Multi-root DLLP |
| MRINIT | Multi-root initialization | Multi-root DLLP |
| MRRESET | Multi-root reset | Multi-root DLLP |
| IDL | Logical idle | Framing token |
| EDB | End bad | Framing token |
| EDS | End of data stream | Framing token |
| SDAT Sync data | Sync data | |
| SCTR Sync control | Syn Control | |
| PMUX | Protocol multiplexing | |

## 13.22.2 PCIe configuration

### 13.22.2.1 PCIe configuration settings

Access: [Bus] > "PCIe" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Generation**
Selects the generation of the PCIe technology.

| | |
|---|---|
| "PCIe Gen1 (2.5 Gbit/s) " | Selects the PCIe generation 1.0, standard introduced in 2003. It has a transfer rate of 2.5 Gbit/s. |
| "PCIe Gen2 (5 Gbit/s) " | Selects the PCIe generation 2.0, standard introduced in 2007. It has a transfer rate of 5 Gbit/s. |
| "PCIe Gen3 (8 Gbit/s) " | Selects the PCIe generation 3.0, standard introduced in 2010. It has a transfer rate of 8 Gbit/s. |

Remote command:
`BUS<m>:PCIE:GEN` on page 2245

**Link**
Selects the link width, the number of lanes that are used for the transmission of the data.

Remote command:
`BUS<m>:PCIE:LNKW` on page 2245

**Descrambling**
Enables descrambling of the data.

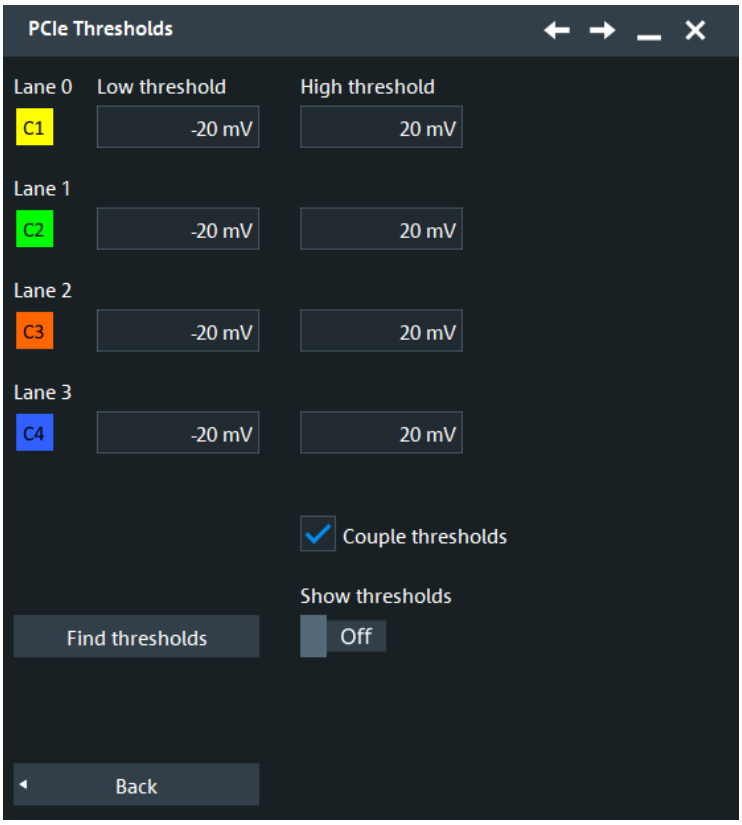
Remote command:
`BUS<m>:PCIE:DSCRambling` on page 2244

**Lane 0/1/2/3**
Select the signal sources for the logical lanes.

Usually, the source is one of the analog channels. Reference and math waveforms are available as source if the trigger source is one of the analog channels but not the serial bus.

Remote command:
`BUS<m>:PCIE:LZER:SOURce` on page 2245
`BUS<m>:PCIE:LONE:SOURce` on page 2246
`BUS<m>:PCIE:LTWO:SOURce` on page 2246
`BUS<m>:PCIE:LTHRee:SOURce` on page 2246

**Thresholds**
Sets the threshold value for digitization of signals for each line.




**Couple thresholds ← Thresholds**
Enables the same threshold value for all lanes.

**Find thresholds ← Thresholds**
Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

Remote command:
`BUS<m>:SETReflevels` on page 1638

**Low threshold ← Thresholds**
Sets the lower threshold value for the respective lane.

Remote command:
BUS<m>:PCIE:LONE:THRLow on page 2246
BUS<m>:PCIE:LTHRee:THRLow on page 2246
BUS<m>:PCIE:LTWO:THRLow on page 2246
BUS<m>:PCIE:LZER:THRLow on page 2246

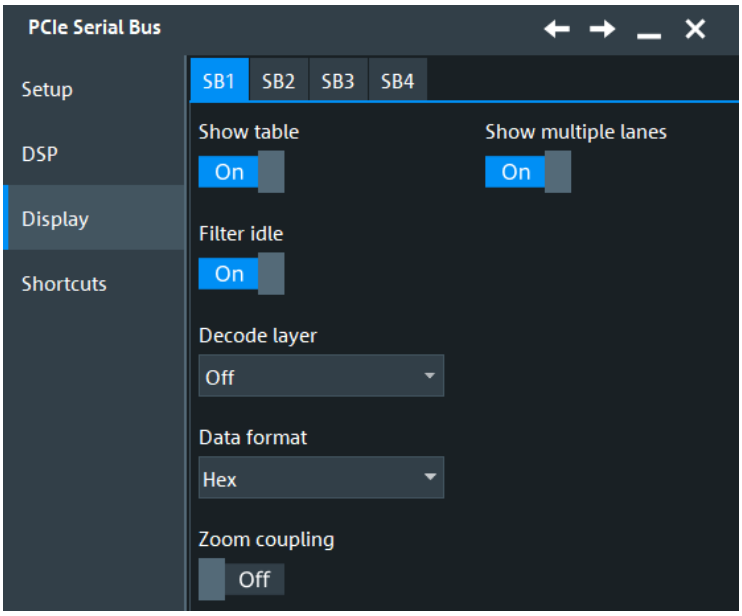
**High threshold ← Thresholds**
Sets the high threshold value for the respective lane.

Remote command:
BUS<m>:PCIE:LONE:THRHigh on page 2246
BUS<m>:PCIE:LTHRee:THRHigh on page 2246
BUS<m>:PCIE:LTWO:THRHigh on page 2246
BUS<m>:PCIE:LZER:THRHigh on page 2246

### 13.22.2.2 Display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.




**Decode layer**
Selects the decode layer.

Decoding is performed in several steps, and the end results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

Available are:
- "Off"
- "Edges"

- "Binary"

**Filter Idle**
Enables the filtering of idle packets.

Remote command:
`BUS<m>:PCIE:FIDLe` on page 2244

**Show multiple lanes**
Displays the result in multiple lanes.

Remote command:
`BUS<m>:PCIE:SMLanes` on page 2246

#### 13.22.2.3 Configuring PCIe Gen 1/2

For configuration, you assign the line to the input channel, set the threshold, the bitrate, and the sync symbol.

For details on configuration settings, see Chapter 13.22.2.1, "PCIe configuration settings", on page 884.

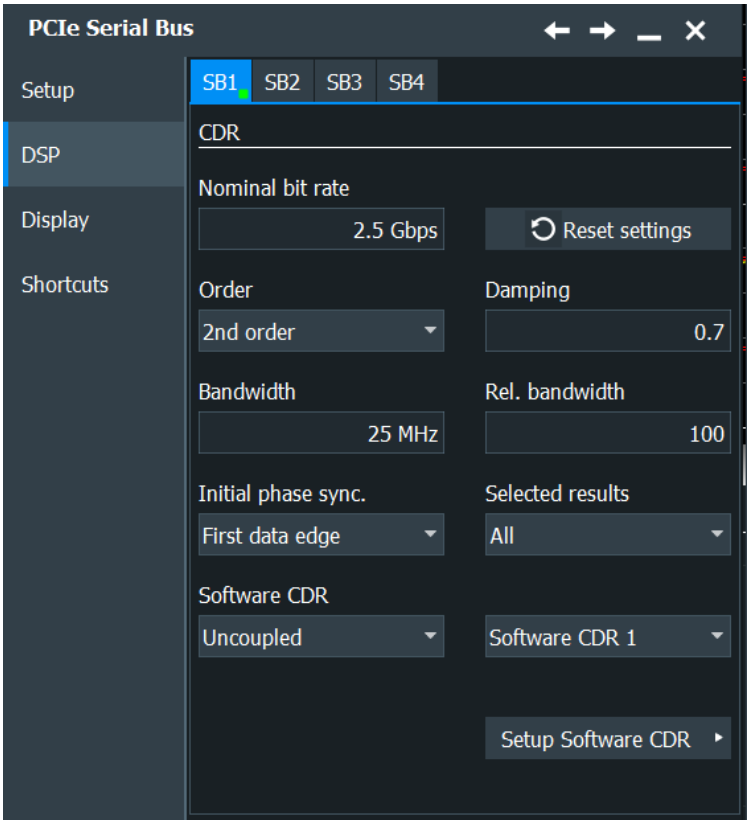
1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up.
3. In the "Setup" tab, select the protocol: "PCIe".
4. Tap the "Generation" button and select the PCIe protocol.
5. Tap the "Link" button and chose the link width.
6. Select the source for of the signal for each lane.
7. Check the threshold settings for each lane. Adjust the values if necessary.

### 13.22.3 PCIe digital signal processing (DSP) settings

#### 13.22.3.1 PCIe Gen 1/2 CDR configuration settings

The process of clock data recovery (CDR) generates a reference clock from a high-speed serial data stream that is sent without a dedicated clock signal. The generated clock signal matches the frequency and is aligned to the phase of the data stream. The recovered clock can be used to sample the data stream and to obtain the sequence of transmitted bits.

Access: [Bus] > "Setup" tab > "Protocol = PCIe" = "Generation = Gen1/Gen2" > "DSP" tab




**Nominal bit rate**
Sets the quiescent frequency of the PLL. It corresponds to the data rate of the data stream from which the clock is to be recovered.

Remote command:
`BUS<m>:CDR:BITRate` on page 2247

**PLL settings**
Phase-locked loop parameters are listed below.

**Note**: Nomial bit rate, bandwidth and relative bandwidth are interacting settings. Modifying one parameter also changes one of the dependent parameters.

"Order" Sets the order of the PLL: first or second order. PLL of higher order can compensate for more complex jitter behavior.

"Bandwidth" Sets the PLL bandwidth. It defines the part of the spectrum that the PLL can follow during synchronization. The PLL bandwidth is usually defined by the transmission standard.

"Rel. bandwidth" Sets the relative bandwidth, that is the ratio of the nominal bit rate to the PLL bandwidth.

"Damping" Sets the damping factor, which is only relevant for second order PLL.

Remote command:
`BUS<m>:CDR:PLL:ORDer` on page 2248
`BUS<m>:CDR:PLL:BWIDth` on page 2247

BUS<m>:CDR:PLL:RELBwidth on page 2248
BUS<m>:CDR:PLL:DAMPing on page 2247

**Initial phase sync.**
Defines the phase reference for the first clock edge.

"First sample" The first clock edge matches the first sample of the waveform at the left border of the display.

"First data edge" The first clock edge matches the first edge of the data signal.

Remote command:
BUS<m>:CDR:SYNC on page 2248

**Selected results**
The PLL requires some time to synchronize to the phase of the data stream. You can select when the CDR algorithm returns clock edges:

"After initial sync." The clock edges of the synchronization time are discarded; results are gathered after initial synchronization of the CDR. Thus, meaningful TIE measurement results can be obtained.

"All" All clock edges are used.

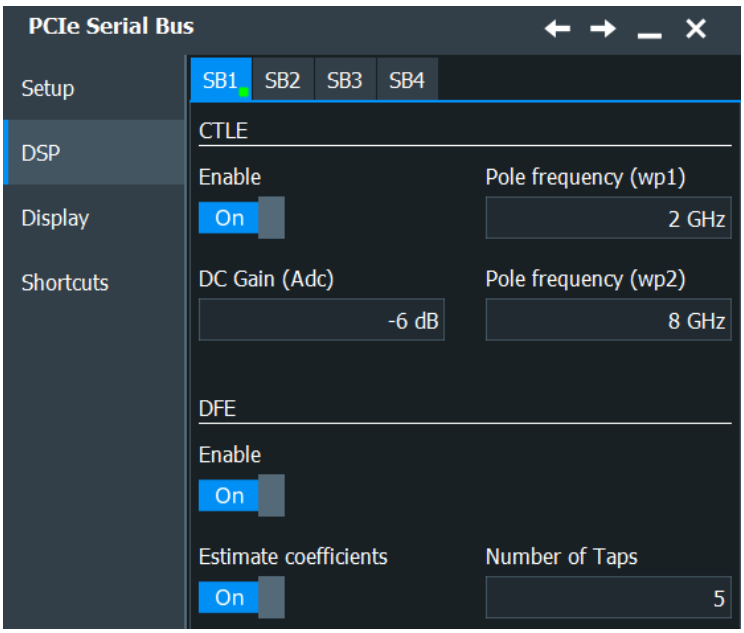
Remote command:
BUS<m>:CDR:RESults on page 2248

**Software CDRs**
Couples the protocol CDR to the selected software CDR. For more information about it see, Chapter 18.3, "Clock data recovery", on page 1067.

### 13.22.3.2 PCIe Gen 3 settings

Access: [Bus] > "Setup" tab > "Protocol = PCIe" = "Generation = Gen3" > "DSP" tab

At this tab, you can enable and define the equalization techniques used for the equalization at the reciever side of the signal.




**CTLE**
Continuous time linear equalization (CTLE) is an equalization that can be used for improving the signal performance. It boosts the high frequency noise and the signal. Thus it helps reduce the inter-symbol interference (ISI) on the reciever side.

The transfer function for behavioral CTLE is defined as follows:

$$H(s) = \omega_{P2}\frac{s + \omega_{P1} * A_{DC}}{(s + \omega_{P1}) * (s + \omega_{P2})}$$

***Figure 13-38: Transfer function for behavioral CTLE***

$\omega_{P1}=2\pi f_{P1}$ = Pole one frequency
$\omega_{P2}=2\pi f_{P2}$ = Pole two frequency

The PCI Express® Base Specification Revision 3.0, defines 1st order CTLE with fixed low frequency (LF) and high frequency (HF) poles for behavioral Rx equalization.

**Enable ← CTLE**
Enables the CTLE equalization.

Remote command:
`BUS<m>:CTLE:ENBLe` on page 2164

**Pole frequency (wp1)/(wp2) ← CTLE**
Sets the pole 1/2 frequency for the transfer function of the CTLE algorithm.

Remote command:
`BUS<m>:CTLE:P1FReq` on page 2165
`BUS<m>:CTLE:P2FReq` on page 2165

**DC Gain ($A_{DC}$) ← CTLE**
Sets the adjustable DC gain. This value can be chosen according to the channel loss characteristics. For low-loss channels, you can set a value close to 0 dB.

Remote command:
`BUS<m>:CTLE:DCGain` on page 2164

**DFE**
Defines the settings of the decision feedback equalization (DFE). DFE correction is needed when the channel is distorted.

**Enable ← DFE**
Enables the use of the decision feedback equalization.

Remote command:
`BUS<m>:PCIE:DFE:ENBLe` on page 2251

**Estimate coefficients ← DFE**
Enables the use of estimate coefficients. The coefficients will

Remote command:
`BUS<m>:PCIE:DFE:AUTO` on page 2251

**Coefficient ← DFE**
Sets the value of the coefficient.

Remote command:
`BUS<m>:PCIE:DFE:COEFficient` on page 2251

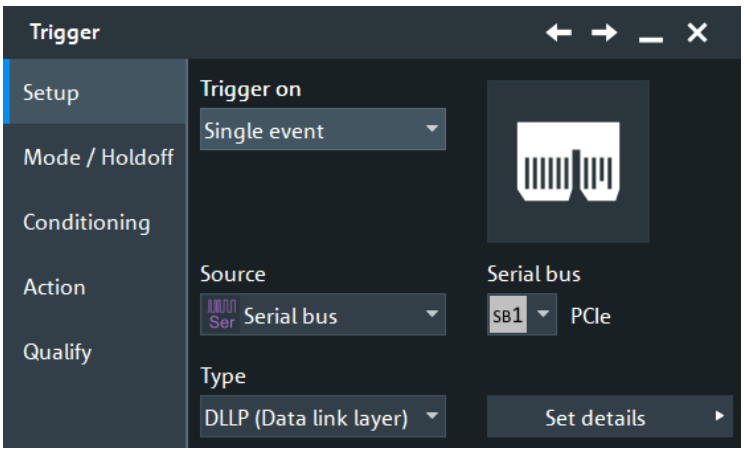
**Number of Taps ← DFE**
Sets a number of taps for the decision feedback filter.

Remote command:
`BUS<m>:PCIE:DFE:TCONt` on page 2251

## 13.22.4 PCIe trigger

### 13.22.4.1 PCIe trigger settings

Access: [Bus] > "Shortcuts" tab > "Setup trigger"




Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Trigger level**
Defines the trigger level for PCIe 3.

Remote command:
`TRIGger<m>:PCIE:TRGLevel` on page 2253

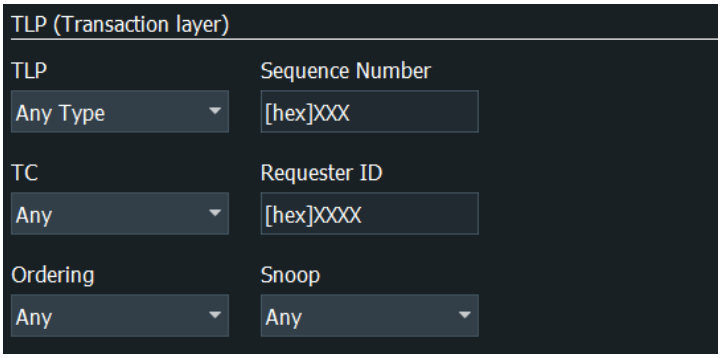
**Trigger type PCI express**
Selects the trigger type for the PCI express analysis.

| | |
|---|---|
| "TLP (Transaction layer)" | Sets the trigger on a defined type of the transaction layer packet. |
| "DLLP (Data link layer)" | Sets the trigger on a defined type of the data link layer packet. |
| "Ordered Sets" | Sets the trigger on a defined ordered set. |
| "Error Condition" | Sets the trigger on a selection of error conditions. |

Remote command:
`TRIGger<m>:PCIE:TYPE` on page 2263

**TLP (Transaction layer)**
Specifies the trigger conditions for the transaction layer type trigger.

**TLP ← TLP (Transaction layer)**
Triggers on a transaction type.




"Any type" Triggers on any transaction type.

"Mem" Triggers on memory transaction type used for data transfers to memory-mapped locations.

"IO" Triggers on I/O transaction type used for data transfers to I/O-mapped locations.

"Cfg" Triggers on configuration transaction type used to set up the device functions.

"Msg" Triggers on message transaction type used for transmitting diverse messages.

"Cpl" Triggers on completion without data.

"FetchAdd" Triggers on fetch and add atomic op request.

"Swap" Triggers on unconditional swap atomic op request.

"CAS" Triggers on compare and swap atomic op request transaction type.

Remote command:
`TRIGger<m>:PCIE:TLP:TYPE` on page 2262

**Sequence Number ← TLP (Transaction layer)**
Triggers on a sequence number, which indicates if any TLPs have been lost.

Remote command:
`TRIGger<m>:PCIE:TLP:SNUM` on page 2262

**TC ← TLP (Transaction layer)**
Triggers on a traffic class which is mapped on a specific virtual channel. The TC label is transmitted unmodified through the fabric.

Remote command:
`TRIGger<m>:PCIE:TLP:TCHN` on page 2262

**Ordering ← TLP (Transaction layer)**
Triggers on an ordering. The ordering can be relaxed or strong.

Remote command:
`TRIGger<m>:PCIE:TLP:ORDE` on page 2261

**Snoop ← TLP (Transaction layer)**
Triggers on a snoop state.

Remote command:
TRIGger<m>:PCIE:TLP:SNOO on page 2262

**Requester ID ← TLP (Transaction layer)**
Triggers on a requester ID, consisting of the requester's bus number, device number, and function number.

Remote command:
TRIGger<m>:PCIE:TLP:REID on page 2261

**Device ID ← TLP (Transaction layer)**
Triggers on a device ID, a field that can be assigned a value by the vendor for classifying root complex register blocks (RCRB). The Device ID is only valid for Configuration transaction. It is composed of 8-bit us number, 5-bit device number and a 3-bit function number.

Remote command:
TRIGger<m>:PCIE:TLP:DEID on page 2260

**Read/Write ← TLP (Transaction layer)**
Triggers on a read and/ or write state.

Remote command:
TRIGger<m>:PCIE:TLP:MERW on page 2260

**Address Type ← TLP (Transaction layer)**
Triggers on an address type. The address type can be 32 bit or 64 bit.

Remote command:
TRIGger<m>:PCIE:TLP:ADRT on page 2259

**Type ← TLP (Transaction layer)**
Triggers on a configuration type.

Remote command:
TRIGger<m>:PCIE:TLP:CFGT on page 2260

**Routing ← TLP (Transaction layer)**
Triggers on a selected message routing type.

Remote command:
TRIGger<m>:PCIE:TLP:MSGR on page 2261

**Message Code ← TLP (Transaction layer)**
Triggers on a specific message code.

Remote command:
TRIGger<m>:PCIE:TLP:MSGC on page 2261

**Status ← TLP (Transaction layer)**
Triggers on a completion status.

Remote command:
TRIGger<m>:PCIE:TLP:CPLS on page 2260

**Completer ID ← TLP (Transaction layer)**
Triggers on a completer ID, consisting of the completer's bus number, device number, and function number.

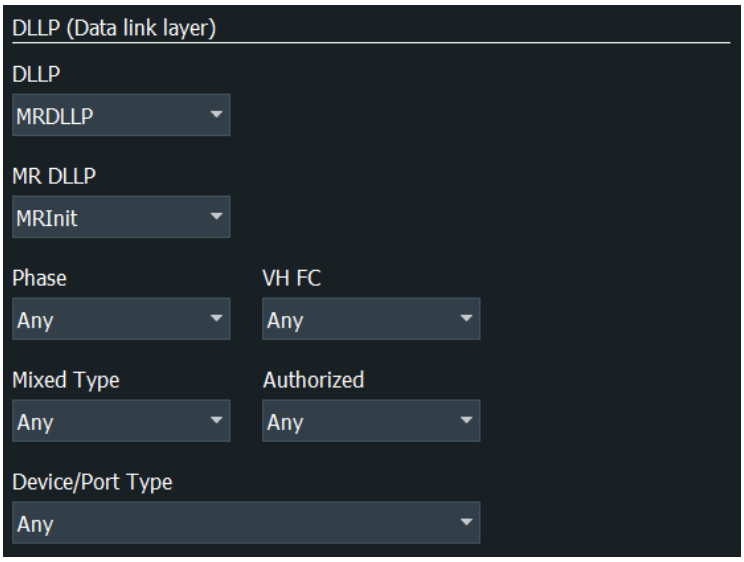
Remote command:
`TRIGger<m>:PCIE:TLP:CPID` on page 2260

**DLLP Type**
Sets the trigger to the type of DLLP. You can refine the settings for the different types according to their specifications.

Remote command:
`TRIGger<m>:PCIE:DLLP:TYPE` on page 2256

**DLLP Any Type conditions**
Sets the trigger to any type of DLLP.

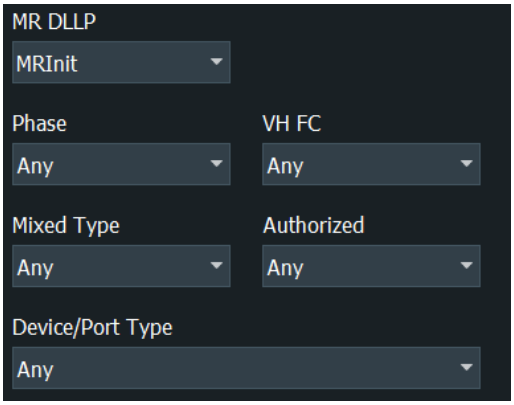
**DLLP MRDLLP conditions**
Specifies the conditions for the multi-root trigger.




**MR DLLP ← DLLP MRDLLP conditions**
Triggers on a multi-root type for the data link layer.

Remote command:
`TRIGger<m>:PCIE:DLLP:MULT:TYPE` on page 2255

**MRInit ← DLLP MRDLLP conditions**
Specifies the conditions for the multi-root initialization protocol trigger.




**Phase ← MRInit ← DLLP MRDLLP conditions**
Triggers on a phase state.

Remote command:
`TRIGger<m>:PCIE:DLLP:MULT:PHAS` on page 2254

**VH FC ← MRInit ← DLLP MRDLLP conditions**
Triggers on a virtual hierarchies flow control presence.

Remote command:
`TRIGger<m>:PCIE:DLLP:MULT:VHFC` on page 2255

**Mixed Type ← MRInit ← DLLP MRDLLP conditions**
Triggers on a mixed type presence.

Remote command:
`TRIGger<m>:PCIE:DLLP:MULT:MIXT` on page 2254

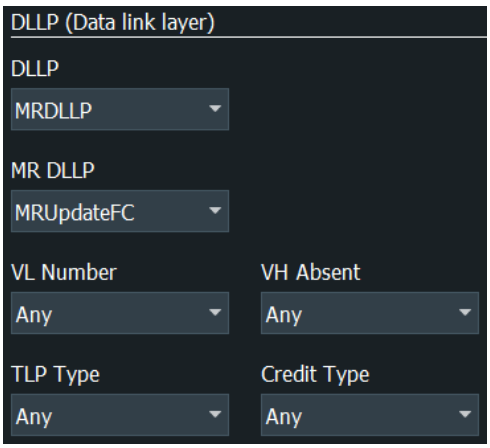
**Authorized ← MRInit ← DLLP MRDLLP conditions**
Triggers on an authorized state presence.

Remote command:
`TRIGger<m>:PCIE:DLLP:MULT:AUTH` on page 2253

**Device/Port Type ← MRInit ← DLLP MRDLLP conditions**
Triggers on an device/port type.

Remote command:
`TRIGger<m>:PCIE:DLLP:MULT:DVPT` on page 2254

**MRInitFC1/InitFC2/UpdateFC ← DLLP MRDLLP conditions**
Specifies the conditions for the multi-root initialization flow control protocol trigger.




**VL Number ← MRInitFC1/InitFC2/UpdateFC ← DLLP MRDLLP conditions**
Triggers on a virtual link (VL) number.

Remote command:
`TRIGger<m>:PCIE:DLLP:MULT:VLNR` on page 2256

**VH Absent ← MRInitFC1/InitFC2/UpdateFC ← DLLP MRDLLP conditions**
Triggers on absent virtual hierarchies (VH).

Remote command:
`TRIGger<m>:PCIE:DLLP:MULT:HABS` on page 2254

**TLP Type ← MRInitFC1/InitFC2/UpdateFC ← DLLP MRDLLP conditions**
Triggers on a transaction layer type for a multi-root data link layer trigger type.

"Any" Any transaction layer type.

"P" Posted credit.

"N" Non posted credit.

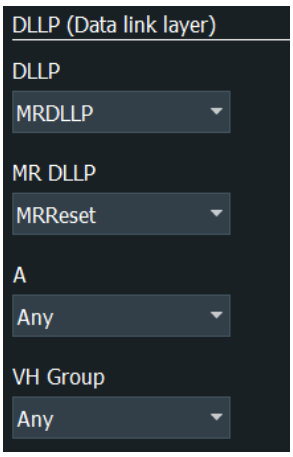
"Cpl" Completion credit.

Remote command:
`TRIGger<m>:PCIE:DLLP:MULT:TLPT` on page 2255

**Credit Type ← MRInitFC1/InitFC2/UpdateFC ← DLLP MRDLLP conditions**
Triggers on any, data or header credit type.

Remote command:
`TRIGger<m>:PCIE:DLLP:MULT:CRET` on page 2254

**MRReset ← DLLP MRDLLP conditions**
Specifies the conditions for the multi-root reset protocol trigger.




**A ← MRReset ← DLLP MRDLLP conditions**
Triggers on the value of the A bit (Ack/Request).

Remote command:
`TRIGger<m>:PCIE:DLLP:MULT:RESA` on page 2255

**VH Group ← MRReset ← DLLP MRDLLP conditions**
Triggers on the selected virtual hierarchies group.

Remote command:
`TRIGger<m>:PCIE:DLLP:MULT:VHGR` on page 2256

**DLLP ACK/NAK conditions**
Triggers on an acknowledgement (ACK)/ negative acknowledgment (NAK).

**Sequence ← DLLP ACK/NAK conditions**
Triggers on the sequence field indicating what TLPs are affected by the ACK/NAK.

Remote command:
`TRIGger<m>:PCIE:DLLP:SEQ` on page 2256

**DLLP Init FC1/InitFC2/UpdateFC conditions**
Triggers on flow control initialization/update conditions.

**Credit Type ← DLLP Init FC1/InitFC2/UpdateFC conditions**
Triggers on a credit type value.

| | |
|---|---|
| "Any" | Triggers on any credit type. |
| "P" | Triggers on posted requests. |
| "NP" | Triggers on non posted requests. |
| "CPL" | Triggers on completions. |

Remote command:
`TRIGger<m>:PCIE:DLLP:FCTL:CRET` on page 2253

**VC ID ← DLLP Init FC1/InitFC2/UpdateFC conditions**
Triggers on a virtual channel ID value.

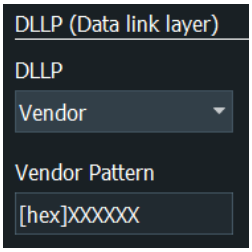
Remote command:
TRIGger<m>:PCIE:DLLP:FCTL:VCID on page 2253

**DLLP PM conditions**
Triggers on a specified power management conditions.

**PM Type ← DLLP PM conditions**
Triggers on a power management type.

Remote command:
TRIGger<m>:PCIE:DLLP:POWM on page 2256

**DLLP Vendor conditions**
Triggers on a specified vendor conditions.




**Vendor Pattern ← DLLP Vendor conditions**
Triggers on a vendor pattern.

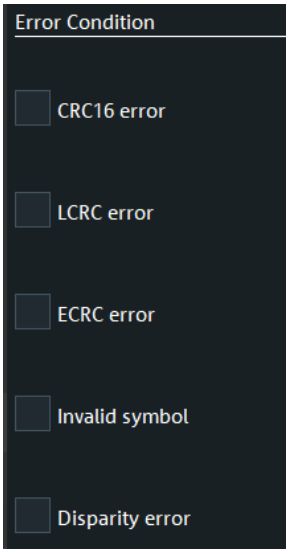
Remote command:
TRIGger<m>:PCIE:DLLP:VPAT on page 2257

**Ordered set condition**
Triggers on an ordered set.

| | |
|---|---|
| "SKP OS" | Triggers on an SKP ordered sets. |
| "Training Seq 1/2" | Triggers on a training sequence ordered set. |
| "Fast Training Seq" | Triggers on a fast training sequence ordered set. |
| "Electrical Idle OS" | Triggers on an electrical idle ordered set that must be send by the transmitter before it can enter the electrical idle. |
| "Electrical Idle Exit OS" | Triggers on an electrical idle exit ordered set. |
| "Compliance Pattern" | Triggers on a compliance pattern ordered set. |

Remote command:
TRIGger<m>:PCIE:OSET:TYPE on page 2259

**Error Condition PCIe Gen 1/2**
Triggers on enabled error condition.




"CRC16 error" Triggers on 16-bit cyclic redundancy check (CRC) errors.

"LCRC error" Triggers on link cyclic redundancy check (LCRC) errors.

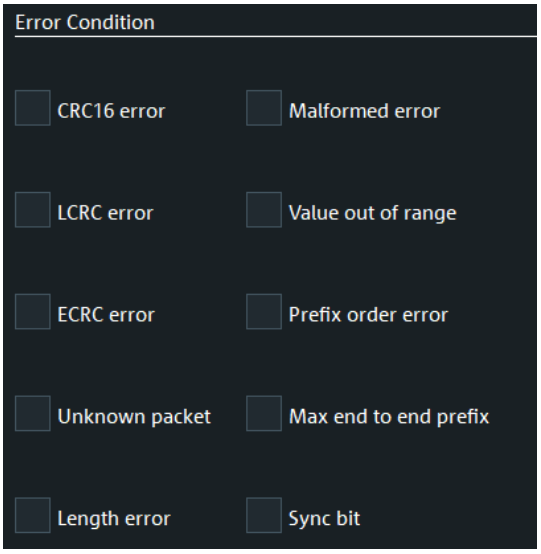
"ECRC error" Triggers on end-to-end cyclic redundancy checksum (ECRC) errors.

"Invalid symbol" Triggers on invalid symbol errors.

"Disparity error" Triggers on disparity errors.

Remote command:
`TRIGger<m>:PCIE:ERRC:CRC` on page 2257
`TRIGger<m>:PCIE:ERRC:DISP` on page 2257
`TRIGger<m>:PCIE:ERRC:ECRC` on page 2257
`TRIGger<m>:PCIE:ERRC:INVP` on page 2257
`TRIGger<m>:PCIE:ERRC:LCRC` on page 2258

**Error Condition PCIe Gen 3**
Triggers on enabled error condition.




"CRC16 error" Triggers on 16-bit cyclic redundancy check (CRC) errors.

"LCRC error" Triggers on link cyclic redundancy check (LCRC) errors.

"ECRC error" Triggers on end-to-end cyclic redundancy checksum (ECRC) errors.

"Unknown packet" Triggers on an unknown packet.

"Length error" Triggers on a length error , for frames that do not have the specified length or data.

"Malformed error" Triggers on a malformed TLP: a TLP that violates specific TLP formation rules that are defined in the specification.

"Value out of range" Triggers, if a value of a field is out of the range defined by the PCIe standard.

"Prefix order error" Triggers on a wrong prefix order.

"Max end to end prefix" Triggers on an end-end prefix error. The maximum number of permitted end-end TLP prefixes is 4.

"Sync bit" Triggers on a sync bit.

Remote command:
`TRIGger<m>:PCIE:ERRC:LENGth` on page 2258
`TRIGger<m>:PCIE:ERRC:MALFormed` on page 2258
`TRIGger<m>:PCIE:ERRC:MXPFix` on page 2258
`TRIGger<m>:PCIE:ERRC:PFORder` on page 2258
`TRIGger<m>:PCIE:ERRC:SYCBit` on page 2258
`TRIGger<m>:PCIE:ERRC:UKNP` on page 2259
`TRIGger<m>:PCIE:ERRC:VOORange` on page 2259

#### 13.22.4.2 Triggering on PCIe Gen 1/2

Prerequisite: A bus is configured for the PCIe Gen 1/2 signal to be analyzed.

For the basic trigger settings, proceed in the following way:

1. Press the [Bus] key and select the "Shortcuts" tab.
2. Press "Setup trigger".
3. Tap the "Source" button and select the "Serial bus" trigger source.
4. Tap "Serial bus" and select the serial bus that is set to PCIe Gen 1/2.
5. Tap "Trigger Type PCI express" and select the trigger type to be used for PCIe Gen 1/2 protocol analysis.
6. Depending on the selected trigger type, more setup conditions can be specified.

For information on how to proceed with the configuration settings, see Chapter 13.22.4.1, "PCIe trigger settings", on page 892.

## 13.22.5 PCIe Gen 1/2 decode results

When the configuration of the serial bus is complete, the signal can be decoded:

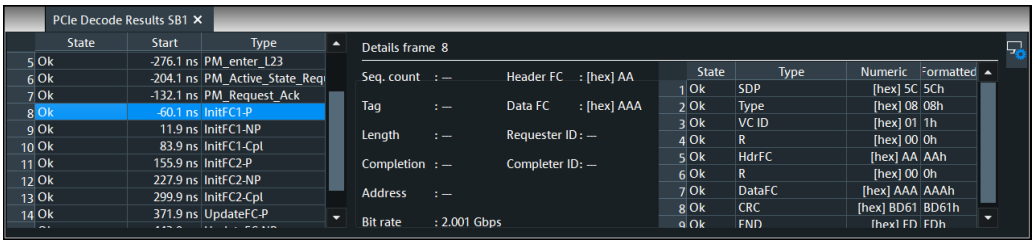
1. In the "Serial Bus" dialog > "Setup" tab, enable "Decode".
2. In the "Serial Bus" dialog > "Display" tab, select additional result display settings: "Show table" and "Show binary".
3. If required, enable "Zoom coupling"

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode results table




The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

*Table 13-43: Content of the Decode results table*

| Column | Description |
|---|---|
| State | Overall state of the packet |
| Start | Start time of the packet |
| Type | Type of packet |

*Table 13-44: Content of the Details frame table*

| Column | Description |
|---|---|
| Sequence count | Sequence count |
| Tag | Tag of the packet |
| Length | Length of packet |
| Completion | Completion status field value |
| Address | Value of the address |
| Bit rate | The bit rate of the packet |
| Header FC | Credit value of the header |
| Data FC | Credit value of the payload data |
| Requester ID | Requester ID |
| Completer ID | Completer ID |
| State | State of the field |
| Type | Field type |
| Numeric | Hexadecimal value of the field |
| Formatted | Formatted content of the field |

**Tools**

Press "Display settings" in the "Decode Results" table to access the "Display" tab.

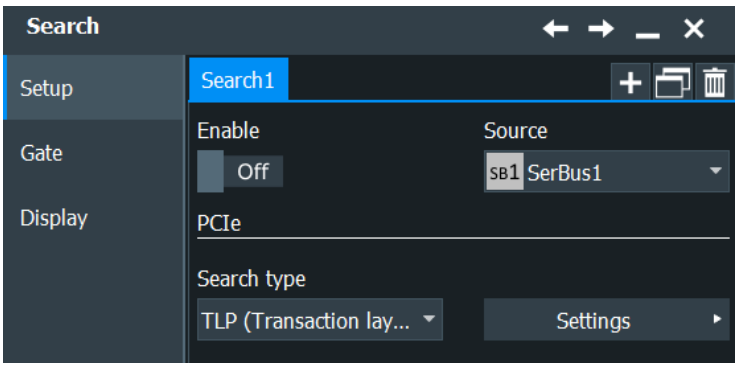
**Remote commands**

Remote commands to retrieve decode results are described in Chapter 24.17.23.5, "Decode results", on page 2263.

## 13.22.6 PCIe Gen 1/2 search

### 13.22.6.1 PCIe Gen 1/2 search settings

Access: [Bus] > "Shortcuts" tab > "Setup search"




Make sure that:

- In the [Bus] > "Setup" tab:
  - The correct "Protocol" is selected.
  - The data sources of the serial bus are channel signals.
- In the "Trigger" > "Setup" tab:
  - The "Trigger on" = "Single event"
  - The trigger "Source" = "Serial bus".
  - The correct "Serial bus" is selected.

**Serial bus**
Selects the serial bus to be triggered on. Make sure to select the correct bus before you enter the settings.

To trigger on a serial bus, the signals sources must be channel signals. If the data or clock source is a math or reference waveform, you cannot trigger on that bus.

Remote command:
`TRIGger<m>:SOURce:SBSelect` on page 1642

**Protocol**
Defines the protocol type of the selected serial bus.

Remote command:
`BUS<m>:TYPE` on page 1637

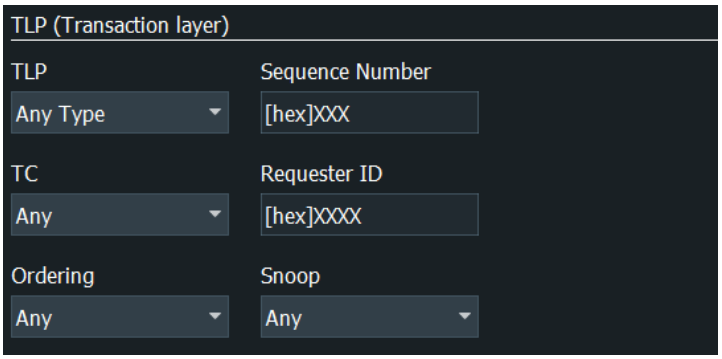
**Searches for PCI express**
Selects the search type for the PCI express analysis.

| | |
|---|---|
| "TLP (Transaction layer)" | Sets the search on a defined type of the transaction layer packet. |
| "DLLP (Data link layer)" | Sets the search on a defined type of the data link layer packet. |
| "Ordered Sets" | Sets the search on a defined ordered set. |
| "Error Condition" | Sets the search on a selection of error conditions. |

Remote command:
`SEARch:TRIGger:PCIE:TYPE` on page 2286

**TLP (Transaction layer)**
Specifies the search conditions for the transaction layer type search.

**TLP ← TLP (Transaction layer)**
Searches for a transaction type.




| | |
|---|---|
| "Any type" | Searches for any transaction type. |
| "Mem" | Searches for memory transaction type used for data transfers to memory-mapped locations. |
| "IO" | Searches for I/O transaction type used for data transfers to I/O-mapped locations. |
| "Cfg" | Searches for configuration transaction type used to setup the device functions. |
| "Msg" | Searches for message transaction type used for transmitting diverse messages. |
| "Cpl" | Searches for completion without data. |
| "FetchAdd" | Searches for fetch and add atomic op request. |
| "Swap" | Searches for unconditional swap atomic op request. |
| "CAS" | Searches for compare and swap atomic op request transaction type. |

Remote command:
`SEARch:TRIGger:PCIE:TLP:TYPE` on page 2286

**Sequence Number ← TLP (Transaction layer)**
Searches for a sequence number, which indicates if any TLPs have been lost.

Remote command:
`SEARch:TRIGger:PCIE:TLP:SNUM` on page 2286

**TC ← TLP (Transaction layer)**
Searches for a traffic class which is mapped on a specific virtual channel. The TC label is transmitted unmodified through the fabric.

Remote command:
`SEARch:TRIGger:PCIE:TLP:TCHN` on page 2286

**Ordering ← TLP (Transaction layer)**
Searches for an ordering. The ordering can be relaxed or strong.

Remote command:
SEARch:TRIGger:PCIE:TLP:ORDE on page 2285

**Snoop ← TLP (Transaction layer)**
Searches for a snoop state.

Remote command:
SEARch:TRIGger:PCIE:TLP:SNOO on page 2285

**Requester ID ← TLP (Transaction layer)**
Searches for a requester ID, consisting of the requester's bus number, device number, and function number.

Remote command:
SEARch:TRIGger:PCIE:TLP:REID on page 2285

**Device ID ← TLP (Transaction layer)**
Searches for a device ID, a field that can be assigned a value by the vendor for classifying root complex register blocks (RCRB). The Device ID is only valid for Configuration transaction. It is composed of 8-bit Bus number, 5-bit Device number and a 3-bit Function number.

Remote command:
SEARch:TRIGger:PCIE:TLP:DEID on page 2284

**Read/Write ← TLP (Transaction layer)**
Searches for a read and/ or write state.

Remote command:
SEARch:TRIGger:PCIE:TLP:MERW on page 2284

**Address Type ← TLP (Transaction layer)**
Searches for an address type. The address type can be 32 bit or 64 bit.

Remote command:
SEARch:TRIGger:PCIE:TLP:ADRT on page 2282

**Type ← TLP (Transaction layer)**
Searches for a configuration type.

Remote command:
SEARch:TRIGger:PCIE:TLP:CFGT on page 2283

**Routing ← TLP (Transaction layer)**
Searches for a selected message routing type.

Remote command:
SEARch:TRIGger:PCIE:TLP:MSGR on page 2284

**Message Code ← TLP (Transaction layer)**
Searches for a specific message code.

Remote command:
SEARch:TRIGger:PCIE:TLP:MSGC on page 2284

**Status ← TLP (Transaction layer)**
Searches for a completion status.

Remote command:
SEARch:TRIGger:PCIE:TLP:CPLS on page 2283

**Completer ID ← TLP (Transaction layer)**
Searches for a completer ID, consisting of the completer's bus number, device number, and function number.

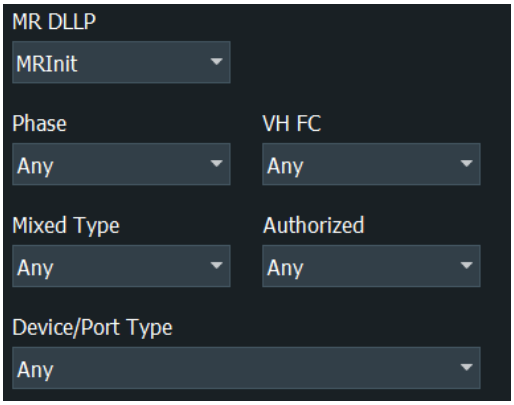
Remote command:
SEARch:TRIGger:PCIE:TLP:CPID on page 2283

**DLLP Type**
Sets the trigger to the type of DLLP. You can refine the settings for the different types according to their specifications.

Remote command:
SEARch:TRIGger:PCIE:DLLP:TYPE on page 2277

**DLLP Any Type conditions**
Sets the trigger to any type of DLLP.

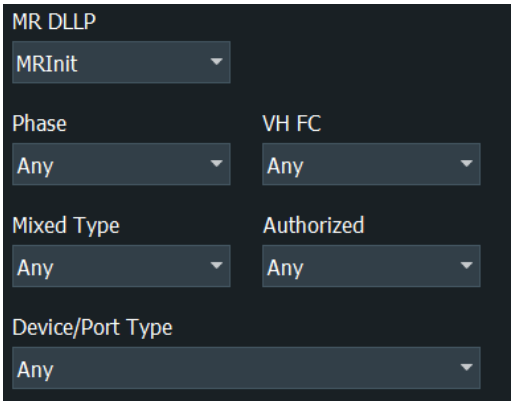
**DLLP MRDLLP conditions**
Specifies the conditions for the multi-root trigger.




**MR DLLP ← DLLP MRDLLP conditions**
Searches for a multi-root type for the data link layer.

Remote command:
SEARch:TRIGger:PCIE:DLLP:MULT:TYPE on page 2275

**MRInit ← DLLP MRDLLP conditions**
Specifies the conditions for the multi-root initialization protocol search.




**Phase ← MRInit ← DLLP MRDLLP conditions**
Searches for a phase state.

Remote command:
SEARch:TRIGger:PCIE:DLLP:MULT:PHAS on page 2274

**VH FC ← MRInit ← DLLP MRDLLP conditions**
Searches for a virtual hierarchies flow control presence.

Remote command:
SEARch:TRIGger:PCIE:DLLP:MULT:VHFC on page 2275

**Mixed Type ← MRInit ← DLLP MRDLLP conditions**
Searches for a mixed type presence.

Remote command:
SEARch:TRIGger:PCIE:DLLP:MULT:MIXT on page 2274

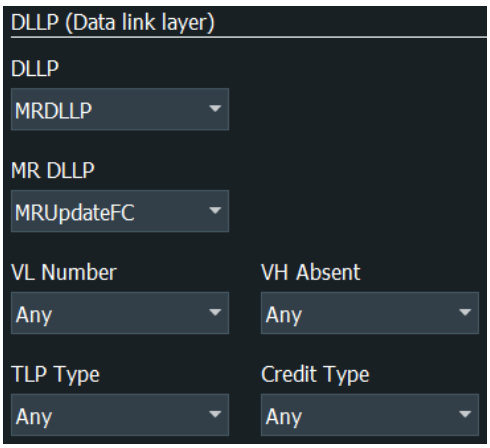
**Authorized ← MRInit ← DLLP MRDLLP conditions**
Searches for an authorized state presence.

Remote command:
SEARch:TRIGger:PCIE:DLLP:MULT:AUTH on page 2273

**Device/Port Type ← MRInit ← DLLP MRDLLP conditions**
Searches for an device/port type.

Remote command:
SEARch:TRIGger:PCIE:DLLP:MULT:DVPT on page 2273

**MRInitInit FC1/InitFC2/UpdateFC ← DLLP MRDLLP conditions**
Specifies the conditions for the multi-root initialization flow control protocol search.




**VL Number ← MRInitInit FC1/InitFC2/UpdateFC ← DLLP MRDLLP conditions**
Searches for a virtual link (VL) number.

Remote command:
`SEARch:TRIGger:PCIE:DLLP:MULT:VLNR` on page 2276

**VH Absent ← MRInitInit FC1/InitFC2/UpdateFC ← DLLP MRDLLP conditions**
Searches for absent virtual hierarchies (VH).

Remote command:
`SEARch:TRIGger:PCIE:DLLP:MULT:HABS` on page 2274

**TLP Type ← MRInitInit FC1/InitFC2/UpdateFC ← DLLP MRDLLP conditions**
Searches for a transaction layer type for a multi-root data link layer trigger type.

"Any" Any transaction layer type.

"P" Posted credit.

"N" Non posted credit.

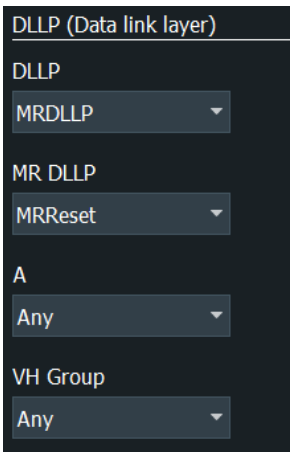
"Cpl" Completion credit.

Remote command:
`SEARch:TRIGger:PCIE:DLLP:MULT:TLPT` on page 2275

**Credit Type ← MRInitInit FC1/InitFC2/UpdateFC ← DLLP MRDLLP conditions**
Searches for any, data or header credit type.

Remote command:
`SEARch:TRIGger:PCIE:DLLP:FCTL:CRET` on page 2272

**MRReset ← DLLP MRDLLP conditions**
Specifies the conditions for the multi-root reset protocol search.




**A ← MRReset ← DLLP MRDLLP conditions**
Searches for the value of the A bit (Ack/ Request).

Remote command:
`SEARch:TRIGger:PCIE:DLLP:MULT:RESA` on page 2274

**VH Group ← MRReset ← DLLP MRDLLP conditions**
Searches for the selected virtual hierarchies group.

Remote command:
`SEARch:TRIGger:PCIE:DLLP:MULT:VHGR` on page 2276

**DLLP ACK/NAK conditions**
Searches for an acknowledgement (ACK)/ negative acknowledgment (NAK).

**Sequence ← DLLP ACK/NAK conditions**
Searches for the sequence field indicating what TLPs are affected by the ACK/NAK.

Remote command:
`SEARch:TRIGger:PCIE:DLLP:SEQ` on page 2277

**DLLP Init FC1/InitFC2/UpdateFC conditions**
Searches for flow control initialization/update conditions.

**Credit Type ← DLLP Init FC1/InitFC2/UpdateFC conditions**
Searches for a credit type value.

"Any" Searches for any credit type.
"P" Searches for posted requests.
"NP" Searches for non posted requests.
"CPL" Searches for completions.

Remote command:
`SEARch:TRIGger:PCIE:DLLP:FCTL:CRET` on page 2272

**VC ID ← DLLP Init FC1/InitFC2/UpdateFC conditions**
Searches for a virtual channel ID value.

Remote command:
SEARch:TRIGger:PCIE:DLLP:FCTL:VCID on page 2273

**DLLP PM conditions**
Searches for a specified power management conditions.

**PM Type ← DLLP PM conditions**
Searches for a power management type.

Remote command:
SEARch:TRIGger:PCIE:DLLP:POWM on page 2276

**DLLP Vendor conditions**
Searches for a specified vendor conditions.

**Vendor Pattern ← DLLP Vendor conditions**
Searches for a vendor pattern.

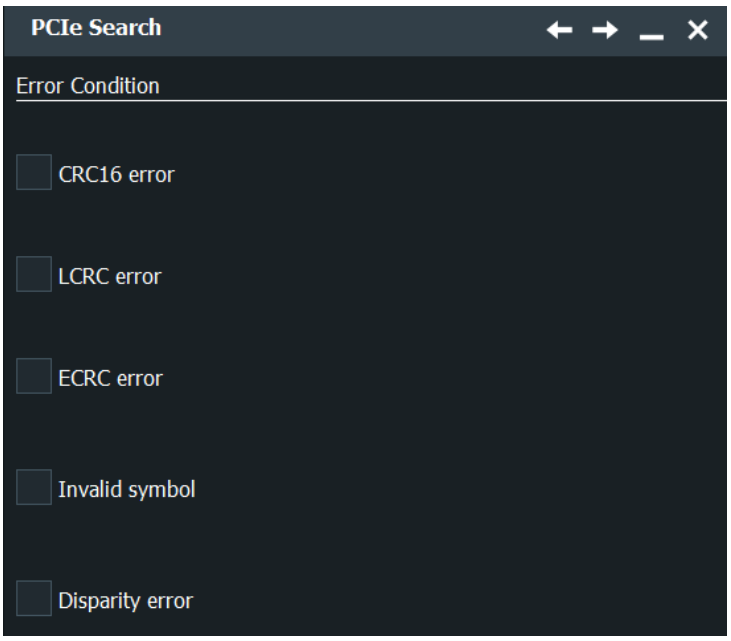
Remote command:
SEARch:TRIGger:PCIE:DLLP:VPAT on page 2277

**Ordered set condition**
Searches for the enabled ordered sets.

| | |
|---|---|
| "SKP OS" | Searches for an SKP ordered sets. |
| "Training Seq 1/2" | Searches for a training sequence ordered set. |
| "Fast Training Seq" | Searches for a fast training sequence ordered set. |
| "Electrical Idle OS" | Searches for an electrical idle ordered set that must be send by the transmitter before it can enter the electrical idle. |
| "Electrical Idle Exit OS" | Searches for an electrical idle exit ordered set. |
| "Start of data sequence OS" | Available for PCIe "Generation" = " Gen 3 (8 Gbit/s)"<br>Searches for a start of data sequence order set. |
| "Compliance Pattern" | Searches for a compliance pattern ordered set. |

Remote command:
SEARch:TRIGger:PCIE:OSET:COMP on page 2281
SEARch:TRIGger:PCIE:OSET:EIDE on page 2281
SEARch:TRIGger:PCIE:OSET:EIDL on page 2281
SEARch:TRIGger:PCIE:OSET:FTS on page 2281
SEARch:TRIGger:PCIE:OSET:SKIP on page 2282
SEARch:TRIGger:PCIE:OSET:TSONe on page 2282
SEARch:TRIGger:PCIE:OSET:TSTWo on page 2282
SEARch:TRIGger:PCIE:OSET:SDS on page 2280

**Error Condition**
Searches for enabled error conditions.




"CRC16 error" Searches for 16-bit cyclic redundancy check (CRC) errors.

"LCRC error" Searches for link cyclic redundancy check (LCRC) errors.

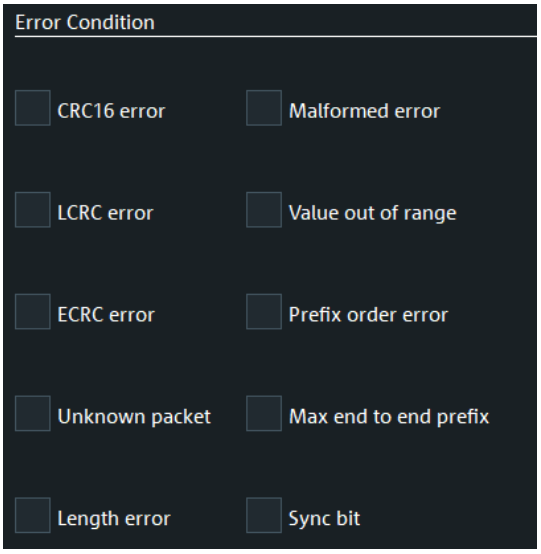
"ECRC error" Searches for end-to-end cyclic redundancy checksum (ECRC) errors.

"Invalid symbol" Searches for invalid symbol errors.

"Disparity error" Searches for disparity errors.

Remote command:
`SEARch:TRIGger:PCIE:ERRC:CRC` on page 2277
`SEARch:TRIGger:PCIE:ERRC:DISP` on page 2278
`SEARch:TRIGger:PCIE:ERRC:ECRC` on page 2278
`SEARch:TRIGger:PCIE:ERRC:INVP` on page 2278
`SEARch:TRIGger:PCIE:ERRC:LCRC` on page 2278

**Error Condition PCIe Gen 3**
Searches for enabled error condition.




"CRC16 error" Searches for 16-bit cyclic redundancy check (CRC) errors.

"LCRC error" Searches for link cyclic redundancy check (LCRC) errors.

"ECRC error" Searches for end-to-end cyclic redundancy checksum (ECRC) errors.

"Unknown packet" Searches for an unknown packet.

"Length error" Searches for a length

"Malformed error" Searches for a malformed TLP: a TLP that violates specific TLP formation rules that are defined in the specification.

"Value out of range" Triggers, if a value is out of range.

"Prefix order error" Searches for a wrong prefix order.

"Max end to end prefix" Searches for an end-end prefix error. The maximum number of permitted end-end TLP prefixes is 4.

"Sync bit" Searches for a sync bit.

Remote command:
`SEARch:TRIGger:PCIE:ERRC:LENGth` on page 2279
`SEARch:TRIGger:PCIE:ERRC:MALFormed` on page 2279
`SEARch:TRIGger:PCIE:ERRC:MXPFix` on page 2279
`SEARch:TRIGger:PCIE:ERRC:PFORder` on page 2279
`SEARch:TRIGger:PCIE:ERRC:SYCBit` on page 2280
`SEARch:TRIGger:PCIE:ERRC:UKNP` on page 2280
`SEARch:TRIGger:PCIE:ERRC:VOORange` on page 2280

### 13.22.6.2 PCIe Gen 1/2 search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

The results of the search are the same as for decoding, see Chapter 13.23.3, "DDR decode results", on page 923.

Remote Commands:

- `SEARch:RESult:PCIE:PCOunt?` on page 2293
- `SEARch:RESult:PCIE:PACKet<m>:ADDR?` on page 2287
- `SEARch:RESult:PCIE:PACKet<m>:CPID?` on page 2287
- `SEARch:RESult:PCIE:PACKet<m>:CPS?` on page 2288
- `SEARch:RESult:PCIE:PACKet<m>:DATA?` on page 2288
- `SEARch:RESult:PCIE:PACKet<m>:DFC?` on page 2288
- `SEARch:RESult:PCIE:PACKet<m>:FCOunt?` on page 2289
- `SEARch:RESult:PCIE:PACKet<m>:HFC?` on page 2290
- `SEARch:RESult:PCIE:PACKet<m>:LEN?` on page 2290
- `SEARch:RESult:PCIE:PACKet<m>:RQID?` on page 2291
- `SEARch:RESult:PCIE:PACKet<m>:SEQN?` on page 2291
- `SEARch:RESult:PCIE:PACKet<m>:STARt?` on page 2291
- `SEARch:RESult:PCIE:PACKet<m>:STATe?` on page 2291
- `SEARch:RESult:PCIE:PACKet<m>:STOP?` on page 2292
- `SEARch:RESult:PCIE:PACKet<m>:TAG?` on page 2292
- `SEARch:RESult:PCIE:PACKet<m>:TYPE?` on page 2292
- `SEARch:RESult:PCIE:PACKet<m>:FCOunt?` on page 2289
- `SEARch:RESult:PCIE:PACKet<m>:FLD<n>:FVAL?` on page 2289
- `SEARch:RESult:PCIE:PACKet<m>:FLD<n>:STATus?` on page 2289
- `SEARch:RESult:PCIE:PACKet<m>:FLD<n>:TYPE?` on page 2289
- `SEARch:RESult:PCIE:PACKet<m>:FLD<n>:VAL?` on page 2290

### 13.22.6.3 Searching for PCIe Gen 1/2

Prerequisite: A serial bus is configured for the PCIe signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and selet the "Shortcuts" tab.
2. Tap "Setup search".

3. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
4. Tap "Source" and select the serial bus that is set to PCIe (e.g. "SB1").
5. Specify the search criteria according to Chapter 13.22.6.1, "PCIe Gen 1/2 search settings", on page 904.
6. To acquire a waveform, press [Single].

   The R&S RTP performs an PCIe decode according to the thresholds and protocol settings of the associated serial bus source.
7. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog:

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

## 13.23 DDR (option R&S RTP-K91/K93 )

Double Data Rate Type 3 (DDR3) and Double Data Rate 4 (DDR4 ) are type of Synchronous Dynamic Random Access Memory (SDRAM). With option R&S RTP-K91 / R&S RTP-K93 you can perform the following analysis on DDR3/ DDR4 data:

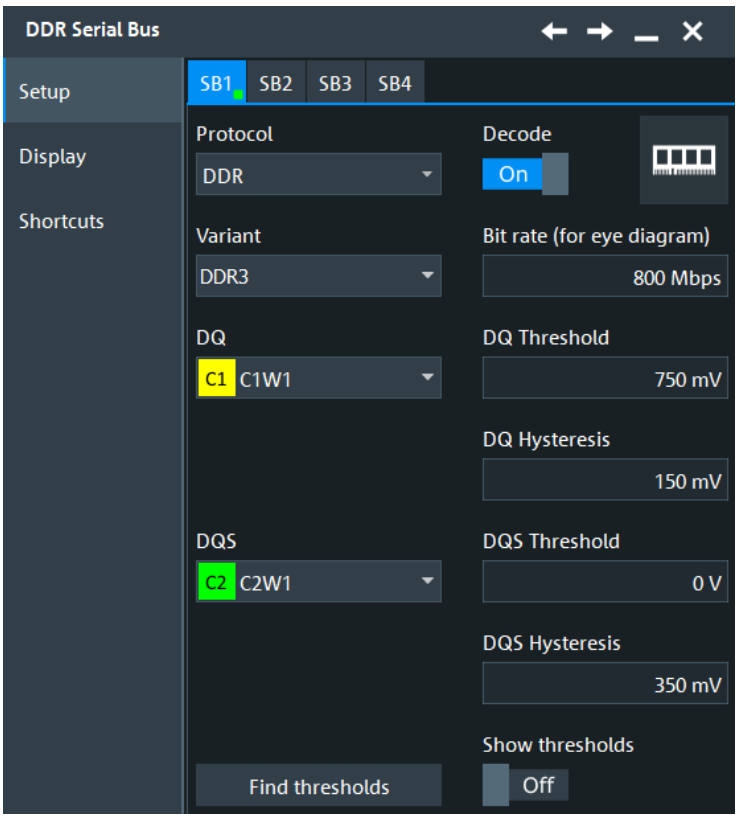
- Decode the signal.
  See:Chapter 13.23.1, "DDR configuration", on page 916
- Create and analyze an eye diagram. This helps to separate the read and write cycle.
  See Chapter 19, "Advanced eye diagram analysis (option R&S RTP-K136/K137)", on page 1072.
- Together with the R&S ScopeSuite software you can perform a compliance test on the signal.
  For details refer to:
  - "DDR3 Compliance Tests, Test Procedures Manual"
  - "DDR4 Compliance Tests, Test Procedures Manual"

### 13.23.1 DDR configuration

#### 13.23.1.1 DDR configuration settings

Access: [Bus] > "DDR" > "Setup" tab

Make sure that the tab of the correct serial bus is selected.




See also: Chapter 13.1.1, "Setup - general settings", on page 516

**Variant**
Selects the DDR standard.

"DDR3" Requires option R&S RTP-K91.

"DDR4" Requires option R&S RTP-K93.

Remote command:
`BUS<m>:DDR:STANdard` on page 2294

**DQ**
Sets the channel for the data output signal.

You can select analog channels "C1" to "C4", and also active math and reference waveforms as source.

Remote command:
`BUS<m>:DDR:DATA:SOURce` on page 2294

**DQS**
Sets the channel for the data strobe signal.

You can select analog channels "C1" to "C4", and also active math and reference waveforms as source.

Remote command:
`BUS<m>:DDR:STRBe:SOURce` on page 2295

**Threshold**
Sets the threshold value for the digitization of the DQ/DQS lane.

There are two ways to set the thresholds:
- "Threshold"
  Enter individual values for each line directly in the fields.
- "Find thresholds"
  Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

Remote command:
`BUS<m>:DDR:DATA:THReshold` on page 2294
`BUS<m>:DDR:STRBe:THReshold` on page 2295
`BUS<m>:SETReflevels` on page 1638

**Hysteresis**
Sets a value for the hysteresis of the respective lane.

Remote command:
`BUS<m>:DDR:DATA:HYSTeresis` on page 2294
`BUS<m>:DDR:STRBe:HYSTeresis` on page 2295

**Bit rate (for eye diagram)**
Declares the nominal bit rate of the data signal for the DDR eye diagram analysis.

See also, Range UI, Position UI.

Remote command:
`BUS<m>:DDR:BITRate` on page 2296

**Show thresholds**
If enabled, the threshold levels are displayed in the diagram.

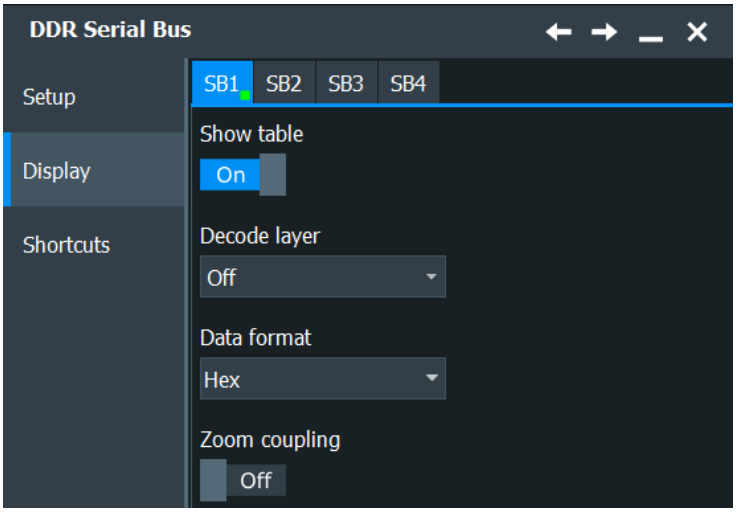
Remote command:
`BUS<m>:THReshold` on page 1638

#### 13.23.1.2 Display settings

Access: [Bus] > "Display" tab.

For common display settings, see Chapter 13.1.3, "Display", on page 518.

To enhance the decode possibilities of the DDR protocol, you can use an additional setting in the "Display" tab: "Decode layer".




**Decode layer**
Selects the decode layer.

Decoding is performed in several layers, and the end results are presented in the decode table. The decode layer selects an interim step for which the decoding result is shown in the honeycomb display.

| | |
|---|---|
| "Edges" | All edges of each lane in stage 1 decoding. |
| "Bits" | All bits of each lane in stage 1 decoding. |
| "Word" | All words of each lane in stage 1 decoding. |

### 13.23.1.3 Configuring the DDR signals

For configuration, assign the lanes to the input channels and define the logical thresholds and the hysteresis.

1. Press the [Bus] key on the front panel.
2. Select the tab of the bus you want to set up, for example "SB1".
3. In the "Setup" tab, set "Protocol"="DDR".
4. Select the DDR "Variant" you want to configure.
5. Optionally, you can enter a "Bus label" on the "Display" tab.
6. Set the channels for the "DS" and "DQS".
7. Set the "Threshold" and "Hysteresis" values.
8. Enable "Decode".

## 13.23.2 Search on decoded DDR data

Using the search functionality, you can find various events in the decoded data. You can find the same events that you can trigger on, and even many more, since several event types can also be combined.

Before you can start the search, you have to configure the bus correctly and acquire decoded data.

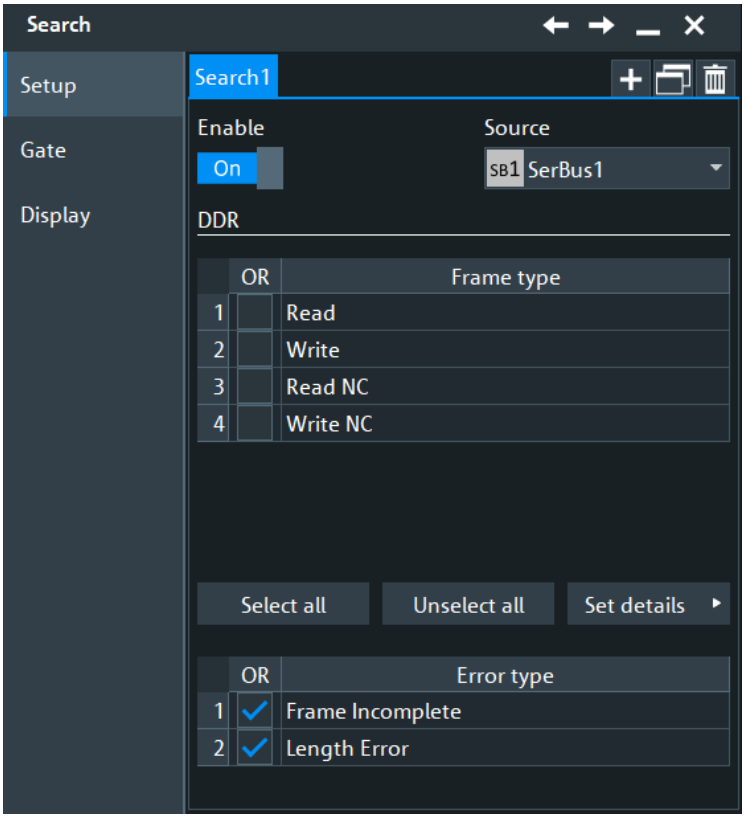
To search on decoded data, set the search source to "SerBus" for the configured protocol.

For general information on how to handle the search functionality, see Chapter 11, "Search functions", on page 449.

If you need information on how to get started with searching for DDR data, see Chapter 13.23.2.3, "Searching for DDR data", on page 923. Otherwise proceed with the DDR search setup.

### 13.23.2.1 DDR search setup

Access: [Bus] > "Shortcuts" tab > "Setup search"




In this dialog, you can define the search type for DDR.

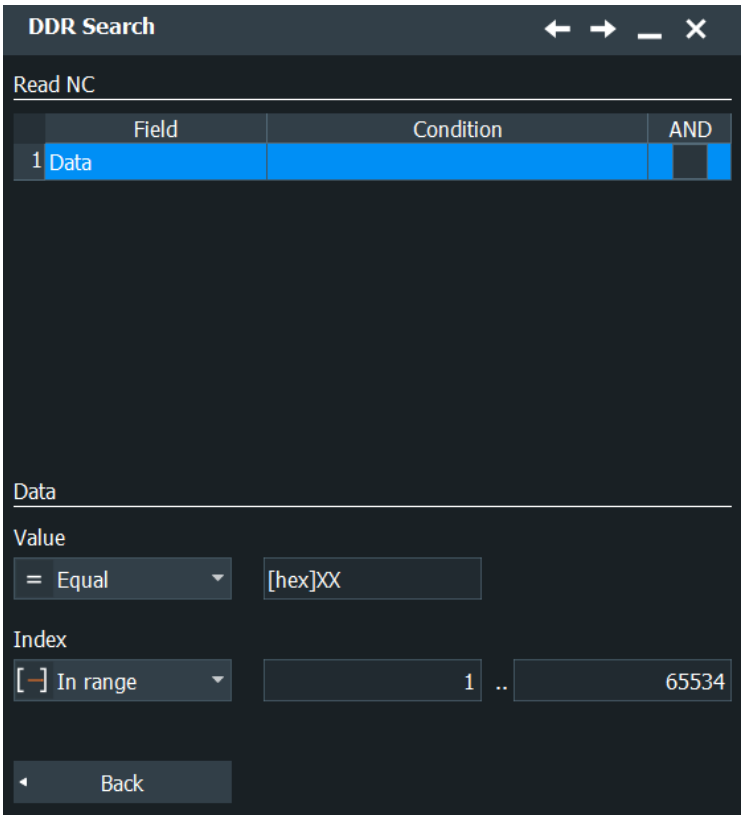
**DDR search**
You can select one or more frames to be searched for. Available are "Read", "Write, ", "Read NC" and "Write NC ". You can also define the data and/or index operators and values for the field, see Set details.

Remote command:
SEARch:TRIGger:DDR:FRENable on page 2302
SEARch:TRIGger:DDR:FRAMe<m>:ENABle on page 2303

**Error type**
You can search for "Frame Incomplete" and "Length error".

Remote command:
SEARch:TRIGger:DDR:ERRor<m>:ENABle on page 2301
SEARch:TRIGger:DDR:ERENable on page 2301

**Set details**
In this dialog, you can define the individual search parameters for the fields of the selected frame. Select the frame type first, and then open the "Set details" dialog.




Remote command:
SEARch:TRIGger:DDR:FIENable on page 2301

**Value ← Set details**
Sets the data value of the field to be searched for. The search data setup consists of a comparison condition and one or two data patterns.

"Condition" Sets the comparison condition to a specific value or a range.

"Min" Specifies the value or sets the start value of a range.

"Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
SEARch:TRIGger:DDR:DMAX on page 2302
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMAX on page 2302
SEARch:TRIGger:DDR:DMIN on page 2302
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMIN on page 2302
SEARch:TRIGger:DDR:DOPerator on page 2303
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DOPerator on page 2303

**Index ← Set details**
The search index setup consists of a comparison condition and one or two address patterns.

"Condition" Sets the comparison condition to a specific value or a range.

"Min" Specifies the value or sets the start value of a range.

"Max" Sets the maximum value of a range for "Condition" = "In range"/"Out of range".

Remote command:
SEARch:TRIGger:DDR:IMAX on page 2304
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMAX on page 2304
SEARch:TRIGger:DDR:IMIN on page 2304
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMIN on page 2304
SEARch:TRIGger:DDR:IOPerator on page 2305
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IOPerator on page 2305

### 13.23.2.2 DDR search results

To get search results, "Enable" the search. You can minimize, shift or close the search dialog.

The search results are listed in the search result table and marked in the waveform by blue lines.

For an introduction to search results, see:

- Chapter 11.1.2, "Search results", on page 450
- Chapter 11.4.2, "Display zoom settings", on page 470

The results of the search are the same as for decoding, see Chapter 13.23.3, "DDR decode results", on page 923.

Remote commands:

- SEARch:RESult:DDR:FCOunt? on page 2305
- SEARch:RESult:DDR:FRAMe<m>:FLD<n>:FVAL? on page 2306
- SEARch:RESult:DDR:FRAMe<m>:FLD<n>:STARt? on page 2306
- SEARch:RESult:DDR:FRAMe<m>:FLD<n>:STATus? on page 2306
- SEARch:RESult:DDR:FRAMe<m>:FLD<n>:STOP? on page 2307

- SEARch:RESult:DDR:FRAMe<m>:FLD<n>:VAL? on page 2307
- SEARch:RESult:DDR:FRAMe<m>:NWRDs? on page 2308
- SEARch:RESult:DDR:FRAMe<m>:STARt? on page 2308
- SEARch:RESult:DDR:FRAMe<m>:STATe? on page 2308
- SEARch:RESult:DDR:FRAMe<m>:STOP? on page 2309
- SEARch:RESult:DDR:FRAMe<m>:TYPE? on page 2309

#### 13.23.2.3 Searching for DDR data

Prerequisite: A serial bus is configured for the DDR signal to be decoded and analyzed.

The search for events is set up in the following way:

1. Press the [Bus] key and select the "Shortcuts" tab.
2. If the dialog box does not contain a search entry, tap the "Add" icon to create one, as described in "To create a user-defined search" on page 464.
3. Tap "Source" and select the serial bus that is set to DDR (e.g. "SB1", unless already selected).

   The search dialog for DDR protocol analysis is opened.
4. Specify search criteria according to Chapter 13.23.2.1, "DDR search setup", on page 920.
5. To acquire a waveform, press [Single].

   The R&S RTP performs a DDR decode according to the thresholds and protocol settings of the associated serial bus source (here in our example SB1).
6. To start searching the acquired waveform for specific events, tap "Enable" in the search setup dialog:

   The R&S RTP displays the "Search Results" box that lists the detected events.

For information on how to configure the search results presentation and how to navigate the search results, see also "To display search zoom windows" on page 472.

### 13.23.3 DDR decode results

When the configuration of the serial bus is complete, the signal can be decoded:

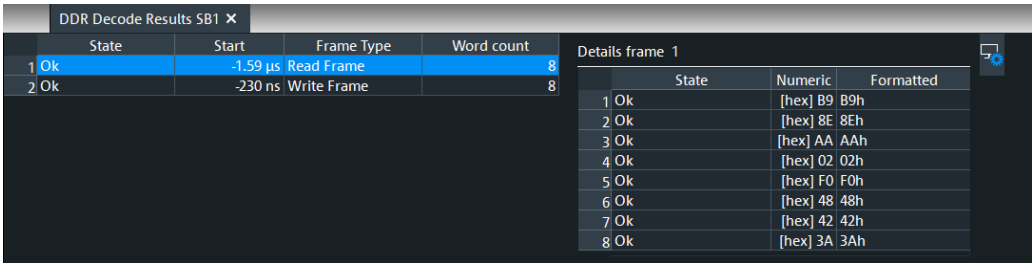
1. In the "Protocol" dialog > "Setup" tab, enable "Decode".
2. In the "Protocol" dialog > "Display" tab, select additional result display settings: "Show decode table" and "Show binary signals".

For a description of the display settings, see also Chapter 13.1.3, "Display", on page 518.

The instrument captures and decodes the signal according to the standard definition and the configuration settings.

The color-coding of the various protocol sections and errors simplifies the interpretation of the visual display. The decode information condenses or expands, depending on the horizontal scale. Various data formats are available to show the result values.

### Decode Results Table




The results are shown in two tables:

- "Decode results": contains information about all decoded frames
- "Details frame": contains more detailed information about the selected frame in the "Decode results" table.

***Table 13-45: Content of the Decode result table***

| Column | Description |
|---|---|
| Frame | Frame count |
| State | Overall state of the frame indicating, for example, if the frame is valid or invalid |
| Start | Time of frame start in relation to the trigger point |
| Frame Type | Frame identifier specifying the data or control frame name |
| Word count | Total number of words in the frame |

***Table 13-46: Content of the Details frame table***

| Column | Description |
|---|---|
| State | State of the field |
| Numeric | Hexadecimal value of the field |
| Formatted | Formatted content of the field |

### Remote commands

Remote commands are described in Chapter 24.17.24.2, "Decode results", on page 2296.

## 13.23.4 Eye diagram setup

This chapter describes the configuration of eye diagrams.

The configuration of an eye diagram mainly consists of the selection and setup of the input source and timing reference source. You can also adjust display settings and restrict the segments that contribute to the eye generation by various means.

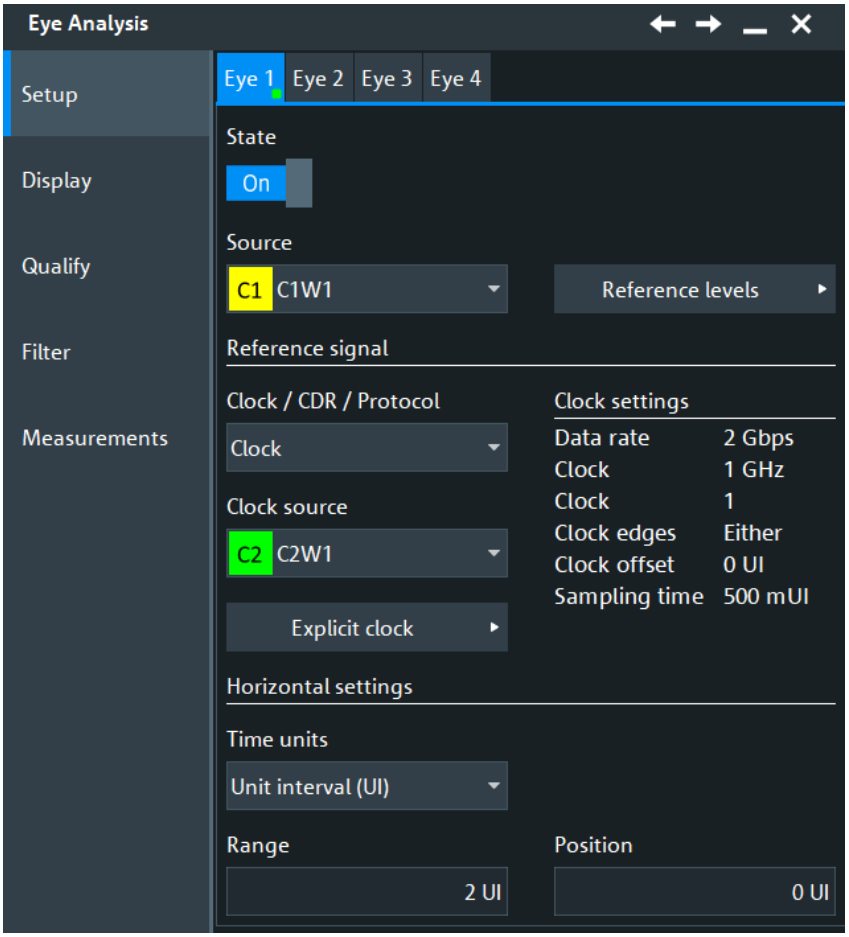
You can configure and analyze up to four independent eye diagrams in parallel.

The "Eye Diagram" dialog box provides four tabs with all settings that are required to configure eye diagrams.

### 13.23.4.1 Basic setup

Access: [Apps]> "Analysis" > "Eye Analysis" > "Setup" tab

In the "Setup" tab, you configure the source, the timing reference, and horizontal scale settings of the eye diagram. The vertical scale of the eye diagram is coupled to the vertical scale of the data source.




You can configure and analyze up to four independent eye diagrams in parallel (vertical subtabs Eye 1 to Eye 4). Make sure to select the correct eye diagram subtab on the left before you adjust the settings.

State....926
Source....926
Reference signal....926

└ Clock / CDR / Protocol ..... 926
└ Clock ..... 926
└ CDR ..... 927
└ Protocol ..... 927
Horizontal settings ..... 928
└ Time units ..... 928
└ Range, Position ..... 928
└ Range UI, Position UI ..... 928
Explicit clock ..... 929
└ Vertical reference level ..... 929
└ Clock frequency ..... 929
└ Estimate clock frequency ..... 929
└ Clock multiplier ..... 930
└ Clock edge ..... 930
└ Data rate ..... 930
└ Clock offset ..... 930
└ Estimate clock offset ..... 930
└ Sampling time ..... 930

**State**
Activates or deactivates the eye diagram.

Remote command:
`EYE<m>[:STATe]` on page 2413

**Source**
Selects the waveform from which the eye diagram is generated (data source). You can select analog channels, reference and math waveforms.

If the source waveform is minimized or hidden, no measurement results for the eye diagram are shown.

Select the same channel for the DDR serial bus protocol configuration "DQ" data line and the input for the eye diagram.

Remote command:
`EYE<m>:SOURce` on page 2413

**Reference signal**
In this section you can define the reference signal used for the eye diagram.

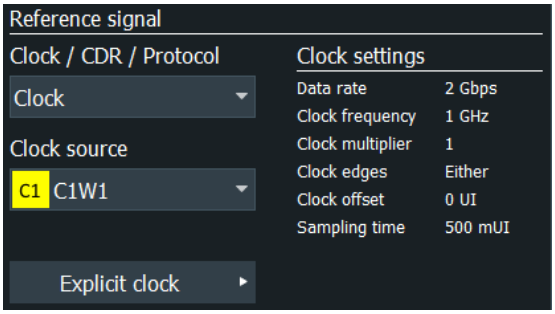
**Clock / CDR / Protocol ← Reference signal**
Selects the type of waveform that is used to obtain the timing information required to slice the data source waveform.

Remote command:
`EYE<m>:TIMReference[:SOURce]` on page 2418

**Clock ← Reference signal**
If a clock signal is available, the instrument can use it directly to set the segmentation timestamps. The signal has to be provided on an additional channel.

With "Clock source" you can select the source of the clock signal.

See also Explicit clock.

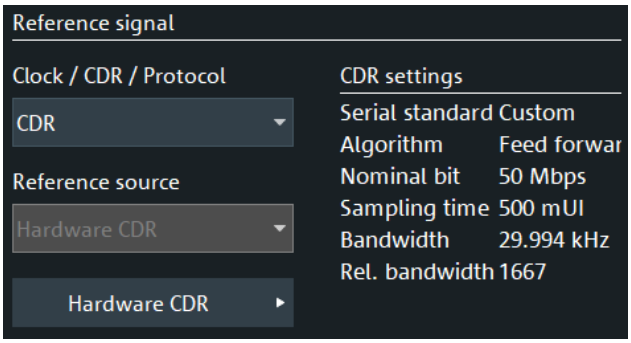



Remote command:
EYE<m>:TIMReference:CLOCk:SOURce on page 2418

**CDR ← Reference signal**
Available, if Type = "CDR".

Uses a hardware clock data recovery. You can also load a reference waveform that contains CDR information in the "Menu" > "Save/Recall" > "Recall" > "Waveform" dialog.

See Hardware CDR trigger.

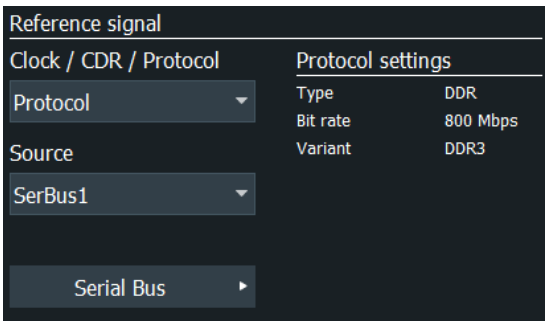



Remote command:
REFCurve<m>:CDR:ALGorithm? on page 1398
REFCurve<m>:CDR:BITRate? on page 1398
REFCurve<m>:CDR:BWIDth? on page 1399
REFCurve<m>:CDR:RELBwidth? on page 1399
REFCurve<m>:CDR:SAMTime? on page 1399

**Protocol ← Reference signal**
If "Protocol" is selected as the reference type signal, the DDR source signal settings are used for the timing information.

You can select the "Source" of the protocol signal. You can only select the protocols that are enabled.

Also some protocol-specific settings are displayed: "Type", "Bit rate", "Variant".




Remote command:
EYE<m>:TIMReference:PROTocol:SELect on page 2418

**Horizontal settings**
Defines the horizontal settings.

**Time units ← Horizontal settings**
Sets the method to define the horizontal scale of the eye diagram: as absolute time, or in unit intervals. You can use unit interval, if the nominal frequency of the clock signal is known.

Indirectly, the horizontal scale determines the size of one waveform slice. The waveform slice must be fully contained in the waveform acquisition to contribute to the eye diagram.

The time units value, effect the following parameters:

- The units of the "Eye diagram".
- Setup tab of the cursor measurements on "Eye diagram".
- Parameters of the eye measurement: "Eye width".
- Masks tests: user and eye
- Histogram: horizontal Histogram setup.
- "Settings" > "Appearance" > "Annotations" > "X position".

Remote command:
EYE<m>:HORizontal[:UNIT] on page 2414

**Range, Position ← Horizontal settings**
Available for Time units = "Absolute time"

Define the horizontal scale in absolute time.

"Range" Time range that is covered by the eye diagram.

"Position" Place of the zero point in the diagram, in seconds. The zero point is the alignment point on which the slice timestamps are superimposed.

Remote command:
EYE<m>:HORizontal:ABSolute:RANGe on page 2414
EYE<m>:HORizontal:ABSolute:POSition on page 2414

**Range UI, Position UI ← Horizontal settings**
Available for Time units = "Unit interval (UI)"

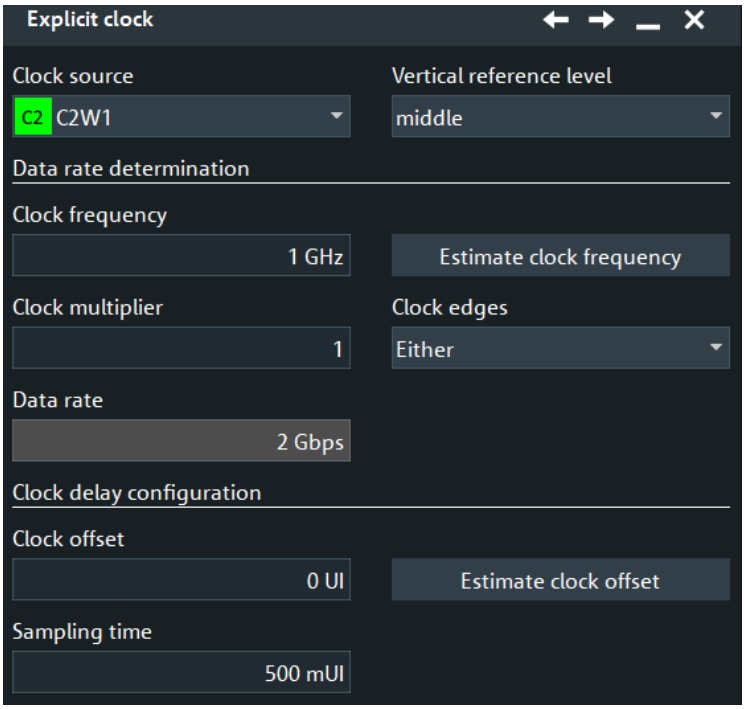
Define the horizontal scale in unit intervals. A unit interval typically corresponds to a bit in the input data signal.

"Range UI" Number of unit intervals that are visible in the eye diagram.

"Position UI" Place of the zero point in the diagram, in unit intervals. The zero point is the alignment point on which the segment timestamps are superimposed.

Remote command:
EYE<m>:HORizontal:UINTerval:RANGe on page 2415
EYE<m>:HORizontal:UINTerval:POSition on page 2415

**Explicit clock**
Opens a dialog to define the setting of the explicit clock for "Clock / CDR / Protocol" = "Clock".




**Vertical reference level ← Explicit clock**
Sets the reference level to define the timestamps for slicing the data waveform. A timestamp is set each time when the selected slope of the timing waveform crosses the selected reference level.

Remote command:
EYE<m>:TIMReference:CLOCk:LEVel on page 2416

**Clock frequency ← Explicit clock**
Sets the clock frequency of the explicit clock signal. If this value is unknown, you can use "Estimate clock frequency".

Remote command:
EYE<m>:TIMReference:CLOCk:FREQuency on page 2416

**Estimate clock frequency ← Explicit clock**
Calculates an estimate value for the clock frequency.

Remote command:
EYE<m>:TIMReference:CLOCk:FREStimation on page 2416

**Clock multiplier ← Explicit clock**
Sets a value for the clock multiplier, the ratio of an internal clock rate to the externally supplied clock.

Remote command:
EYE<m>:TIMReference:CLOCk:FACTor on page 2415

**Clock edge ← Explicit clock**
The clock edge sets the point in time when the state of the data signal is checked:

"Positive" Rising edge, a positive voltage change.

"Negative" Falling edge, a negative voltage change.

"Either" Both the rising and the falling edge.

Remote command:
EYE<m>:TIMReference:CLOCk:EDGE on page 2415

**Data rate ← Explicit clock**
Displays the calculated data rate. It is calculated as follows:

*"Data rate" = "Clock frequency" * "Clock multiplier"*X*

Where *X* is:

- 1 for "Clock edge"= "Positive" or "Negative"
- 2 for "Clock edge"= "Either"

Remote command:
ADVJitter<m>:SIGNal:DATRate? on page 2383

**Clock offset ← Explicit clock**
Sets a clock offset. If this value is unknown, you can use "Estimate clock offset".

Remote command:
EYE<m>:TIMReference:CLOCk:OFFSet on page 2417

**Estimate clock offset ← Explicit clock**
Calculates an estimate value for the clock offset.

Remote command:
EYE<m>:TIMReference:CLOCk:OFEStimation on page 2417

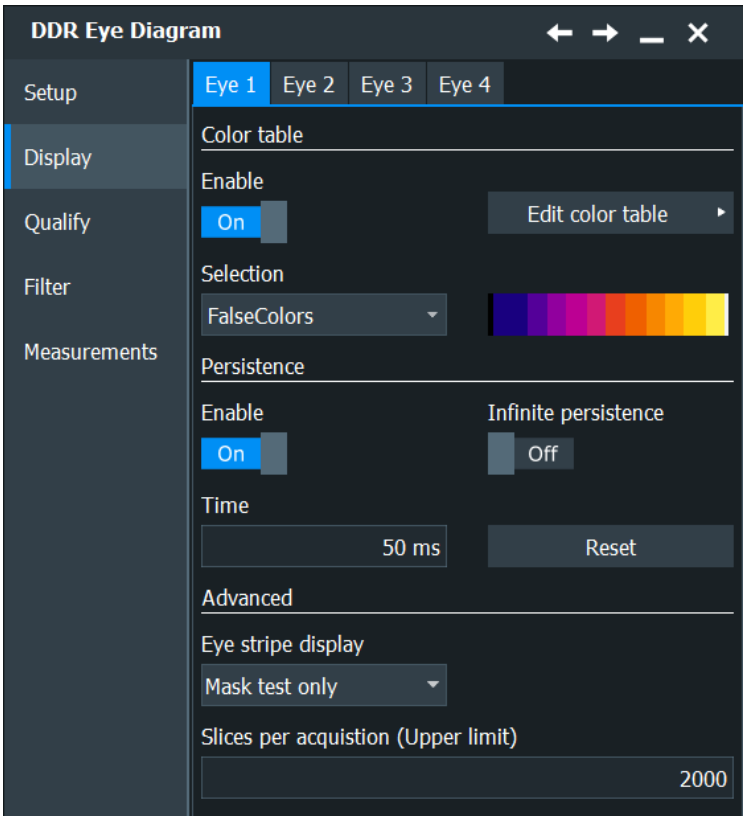
**Sampling time ← Explicit clock**
Sets a sampling time for the explicit clock signal, an offset for the clock edge in relation to the bit start. The clock edge sets the beginning of the unit interval. The sampling time is a number between 0 and 1. Value 0 sets the clock edge to the beginning of the bit period; value 0.5 sets the clock edge to the middle of the bit period.

Remote command:
EYE<m>:TIMReference:CLOCk:SAMTime on page 2417

### 13.23.4.2 Display settings

Access: [Apps]> "Analysis" > "DDR Eye" > "Display" tab

In the "Display" tab, you define the appearance of the individual eye diagrams.




You can configure and analyze up to four independent eye diagrams in parallel (vertical subtabs Eye 1 to Eye 4). Make sure to select the correct eye diagram subtab on the left before you adjust the settings.

**Color table**
Like any other waveform, you can show the eye waveform in solid color, or apply a color table.

For a description of the color table settings, see Chapter 4.3.2, "Color tables", on page 79.

**Persistence**
Each eye diagram has its specific persistence settings, independent of the general waveform persistence.

The persistence settings influence the way that the "Wave Count" and "Event Count" statistics on eye measurements are calculated:

- If "Enable" and "Infinite persistence" are disabled:
  - "Wave Count": for every image used in an eye measurement, sums up the number of acquisitions that contributed to the advanced eye image.
  - "Event Count": the number of UIs of images with successful eye measurement.

- If "Infinite persistence" is enabled:
  - "Wave Count": the number of acquisitions that contributed to the last advanced eye image used in an eye measurement.
  - "Event Count": the number of UIs of the last advanced eye image with successful eye measurement.

See also: Chapter 8.2.12.1, "Statistics", on page 384.

**Enable ← Persistence**
If enabled, each new data point in the diagram area remains on the screen for the duration that is defined using Time, or as long as Infinite persistence is selected.

If disabled, the waveform points are displayed only for the current acquisition.

Remote command:
`EYE<m>:DISPlay:PERSistence:STATe` on page 2420

**Infinite persistence ← Persistence**
If infinite persistence is enabled, each new waveform point remains on the screen until this option is disabled. Use infinite persistence to display rare events in the signal.

Remote command:
`EYE<m>:DISPlay:PERSistence:INFinite` on page 2420

**Time ← Persistence**
Sets a time factor that controls how long the waveform points fade away from the display. Thus, the R&S RTP emulates the persistence of analog phosphor screens.

Remote command:
`EYE<m>:DISPlay:PERSistence:TIME` on page 2420

**Reset ← Persistence**
Resets the display, removing persistent all waveform points.

Remote command:
`EYE<m>:DISPlay:PERSistence:RESet` on page 2420

**Eye stripe display**
Defines when the eye stripe is shown: always, never, or during mask tests.

The eye stripe is shown at the bottom of the main source waveform. It shows the part of the waveform that contributes to the eye. When a mask test is performed on the eye diagram, the eye stripe shows the position of the mask violations in red color.

Remote command:
`EYE<m>:DISPlay:MODE` on page 2419

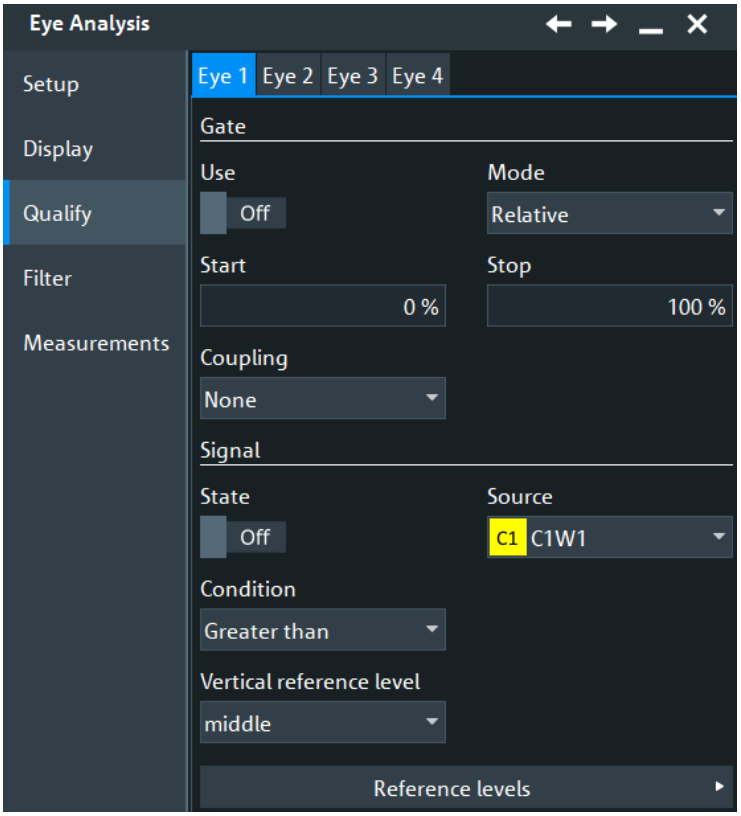
**Slices per acquisition**
Sets the number of waveform slices for a single acquisition.

Remote command:
`EYE<m>:MSLices` on page 2419

### 13.23.4.3 Qualify settings

Access: [Apps]> "Analysis" > "DDR Eye" > "Qualify" tab

Using the qualification settings, you can restrict the slices that contribute to the eye diagram. The restriction can be in horizontal direction (gate), or in vertical direction using the reference level.




You can configure and analyze up to four independent eye diagrams in parallel (vertical subtabs Eye 1 to Eye 4). Make sure to select the correct eye diagram subtab on the left before you adjust the settings.

**Gate**
Restricts the contributing slices in horizontal direction. Only the timestamps within the defined horizontal interval qualify for eye generation.

To apply the gate settings, enable the "State".

The gate definition is the same as gate definition for measurements, but the gates are independent. You can define the gate in one of these ways:

- Enter absolute time values for gate start and gate end: "Coupling mode = None" and "Mode = Absolute". Set "Start" and "Stop" time.
- Enter relative values for gate start and gate end: "Coupling mode = None" and "Mode = Relative". Set "Relative start" and "Relative stop" in percent.
- If a cursor measurement is active, you can use the vertical cursor lines as gate limits: "Coupling mode = Cursor". Select the cursor set.
- If a zoom is active, you can use the zoom limits as gate: "Coupling mode = Zoom". Select the zoom.

Remote command:
EYE<m>:QUALify:GATE[:STATe] on page 2421
EYE<m>:QUALify:GATE:COUPling on page 2421
EYE<m>:QUALify:GATE:MODE on page 2422
EYE<m>:QUALify:GATE:ABSolute:STARt on page 2422
EYE<m>:QUALify:GATE:ABSolute:STOP on page 2422
EYE<m>:QUALify:GATE:RELative:STARt on page 2422
EYE<m>:QUALify:GATE:RELative:STOP on page 2422
EYE<m>:QUALify:GATE:CURSor on page 2423
EYE<m>:QUALify:GATE:ZDIagram on page 2423

**Signal**
Sets a waveform-based qualification for the eye generation. The voltage level of a waveform is compared with a threshold, and only if this condition is fulfilled, the slices contribute to the eye generation.

**State ← Signal**
Applies the signal qualification.

Remote command:
EYE<m>:QUALify:SIGNal[:STATe] on page 2424

**Source ← Signal**
Selects the waveform to be compared.

Remote command:
EYE<m>:QUALify:SIGNal:SOURce on page 2424

**Condition ← Signal**
Selects the comparison condition for the waveform.

Remote command:
EYE<m>:QUALify:SIGNal:CONDition on page 2424

**Vertical reference level ← Signal**
Selects the threshold. You can use the lower, middle, or upper reference level. These levels are the reference levels that are defined for measurements.

See Chapter 8.2.4, "Reference levels", on page 332.

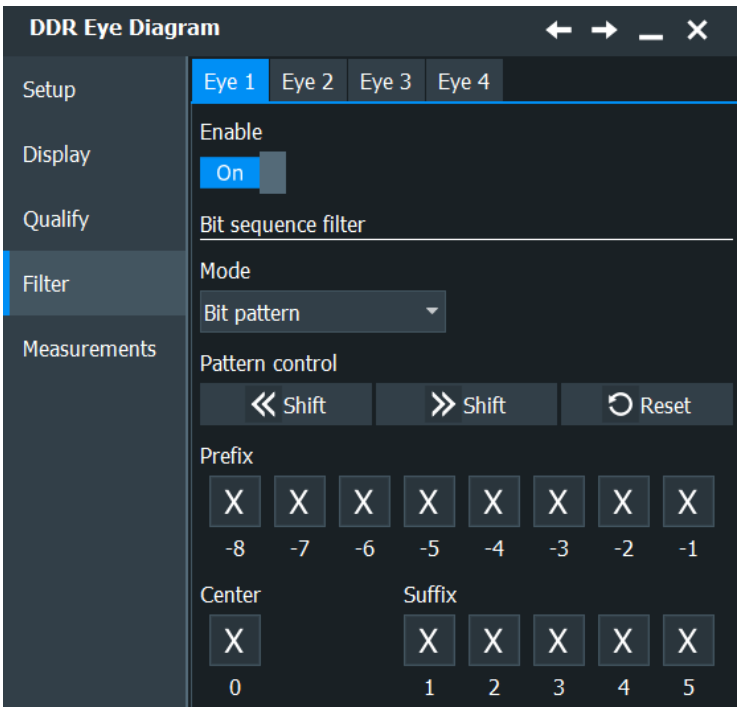
Remote command:
EYE<m>:QUALify:SIGNal:LSELect on page 2425

**Reference levels**
Opens the reference level dialog. For details, see Chapter 8.2.4, "Reference levels", on page 332.

### 13.23.4.4 Filter settings

Access: [Apps]> "Analysis" > "DDR Eye" > "Filter" tab

Filter settings allow you to select waveform slices for eye generation dependent on the bit status.




You can configure and analyze up to four independent eye diagrams in parallel (vertical subtabs Eye 1 to Eye 4). Make sure to select the correct eye diagram subtab on the left before you adjust the settings.

**Enable**
Applies the filter settings.

Remote command:
`EYE<m>:FILTer:STATe` on page 2425

**Mode**
Selects which bits contribute to the eye generation.

| | |
|---|---|
| "All bits" | All bits contribute to the eye diagram, same as "State" = off. |
| "Level transition" | Only bits after a signal transition contribute (0 to 1, or 1 to 0). |
| "Constant level" | Only bits without a preceding signal transition contribute (0 to 0, or 1 to 1). |
| "Bit pattern" | Only bits contribute that match the defined pattern before and after the center eye. Use the bit pattern, for example, to measure the effect of data-dependent jitter on the eye diagram. |

Remote command:
`EYE<m>:FILTer:BPATtern:MODE` on page 2426

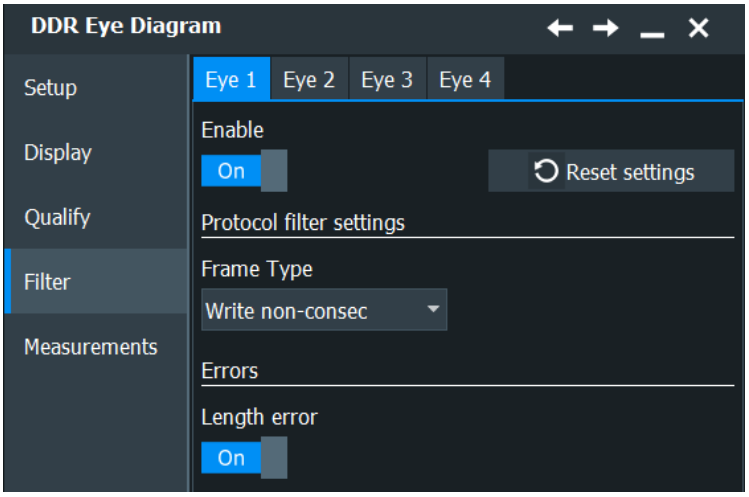
**Pattern**
Defines the bit pattern if "Mode" = "Bit pattern". You can set up to 8 bits before and 5 bits after the center eye (reference bit).

Remote command:
EYE<m>:FILTer:BPATtern:PREFix<1..8> on page 2426
EYE<m>:FILTer:BPATtern:CENTer on page 2426
EYE<m>:FILTer:BPATtern:SUFFix<1..5> on page 2426

**Pattern control**
Shift the pattern bit by bit to the left or to the right, and reset to the default (X for all bits).

Remote command:
EYE<m>:FILTer:BPATtern:SLEFt on page 2426
EYE<m>:FILTer:BPATtern:SRIGht on page 2427
EYE<m>:FILTer:BPATtern:RESet on page 2427

**Protocol filter settings**
This setting is only available, if the Source is set to "SBx" and the "DDR" serial bus analysis is enabled for the same bus.




**Reset settings ← Protocol filter settings**
Resets the current filter settings.

**Frame Type ← Protocol filter settings**
Selects which frame types contribute to the eye diagram, write frames, read frames or any of them.

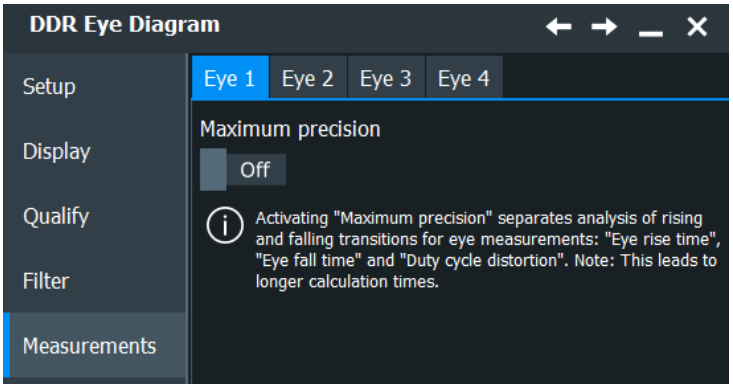
Remote command:
EYE<m>:FILTer:DDR:FTYPe on page 2427

**Length Error ← Protocol filter settings**
If enabled, the length errors contribute to the eye diagram.

Remote command:
EYE<m>:FILTer:DDR:LENerror on page 2427

#### 13.23.4.5 Measurements settings

Access: [Apps]> "Analysis" > "DDR Eye" > "Measurement" tab




**Maximum precision**
If this function is enabled, the rising and falling transitions for the eye measurements "Eye rise time", "Eye fall time" and "Duty cycle distortion" are separated. This leads to longer calculation times.

See also: Chapter 8.2.6, "Eye diagram measurements", on page 355.

Remote command:
`EYE<m>:MEASurement:MAXPrecision` on page 2419

# 14 Mixed signal option (MSO, R&S RTP-B1)

The Mixed Signal Option R&S RTP-B1 adds logic analyzer functions to the classical oscilloscope functions. Using the MSO option, you can analyze and debug embedded systems with mixed-signal designs that use analog signals and correlated digital signals simultaneously.

The Mixed Signal Option provides 16 digital channels grouped in two logic probes (pods) with 8 channels each. The instrument ensures that analog and digital waveforms are time-aligned and synchronized so that critical timing interactions between analog and digital signals can be displayed and tested. The automatic alignment compensates the skew between the probe connectors of the analog channels and the probe boxes of the digital channels.

Logic probes provided by Rohde & Schwarz are recognized by the instrument. The information on each recognized probe is shown under "Menu" "Settings" > "Maintenance" > "Hardware" > "Advanced" > "Logic Probes".

## 14.1 Digital channels and parallel buses

Each digital channel can be displayed on the screen and used as trigger source. Digital channels can be grouped and displayed as a parallel bus. Up to four parallel buses can be configured; and two bus types are supported: clocked bus and unclocked bus. The clocked bus is available only on parallel bus 1 and 2. Each digital channel can be assigned to one *active* parallel bus only, the instrument disables conflicting buses automatically.

You can display each bus and use it as trigger source, as well. For each active parallel bus, the corresponding signal icon appears on the signal bar and indicates the assigned digital channels. Individual digital channels do not have a signal icon.

If one or more parallel buses are active, the roll mode is not available.

### 14.1.1 Parallel buses - configuration

Access: "Menu" > "Logic" > "Setup" tab

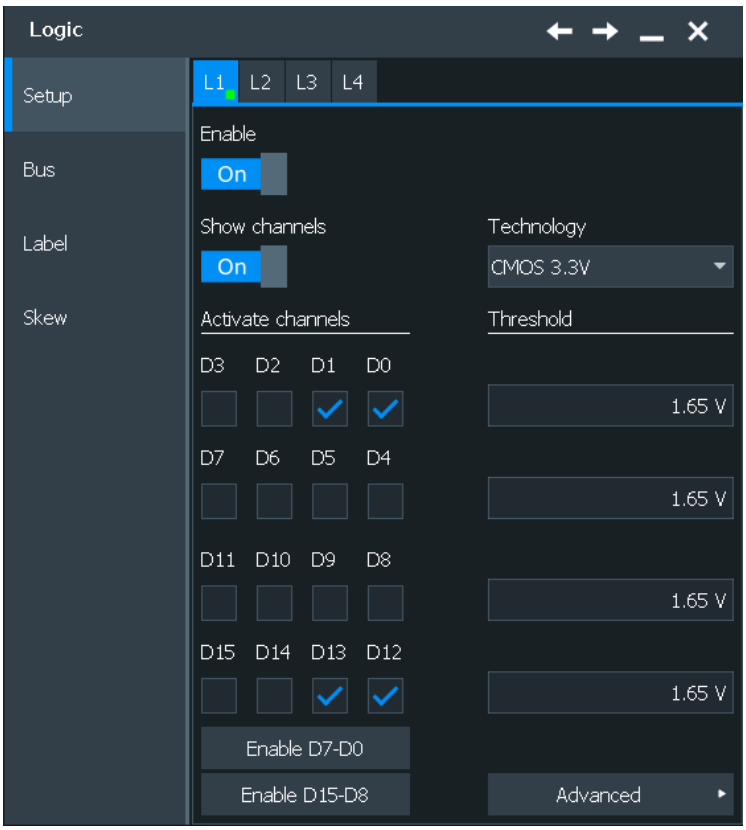
Digital channels can be displayed individually, and they can be grouped and displayed as a parallel bus. You can configure and enable up to 4 parallel buses. Each digital channel can be assigned to one *active* parallel bus only, the instrument disables conflicting buses automatically.

For clocked buses, you can display the decoded data in a result box.

If you have configured several parallel buses and you want to modify the settings, make sure that the tab of the correct bus is selected, and disable the bus before you change the settings.

#### 14.1.1.1 Setup

Access: "Menu" > "Logic" > "Setup" tab




**Enable bus**
Enables the selected parallel bus. The corresponding signal icon appears on the signal bar.

If another *active* bus already uses the same digital channel, the instrument disables the other bus and shows a message.

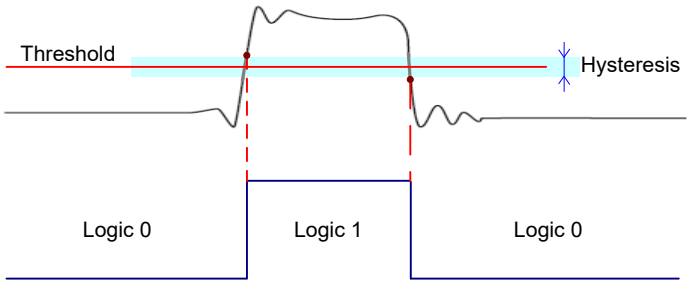
Remote command:
`BUS<m>:PARallel:STATe` on page 2313

**Show channels**
If enabled, the selected digital channels are shown in the diagram. Each channel is displayed as a logic signal.

Remote command:
`BUS<m>:PARallel:DISPlay:SHDI` on page 2317

**Technology, Threshold**
Sets the logical threshold. For each sample, the instrument compares the input voltage with the threshold value. If the input voltage is above the threshold, the signal state "1" is stored. Otherwise, the signal state "0" is stored if the input voltage is below the threshold.

To avoid the change of signal states due to noise, a hysteresis is considered.




By default, same threshold and hysteresis value are used for all digital channels and all parallel buses: "Couple thresholds" is enabled.

You can also set different thresholds for the individual channel groups. As long as the buses are disabled, you can set different thresholds for each bus. Active buses use the same threshold and hysteresis values, the settings of the last activated bus take effect.

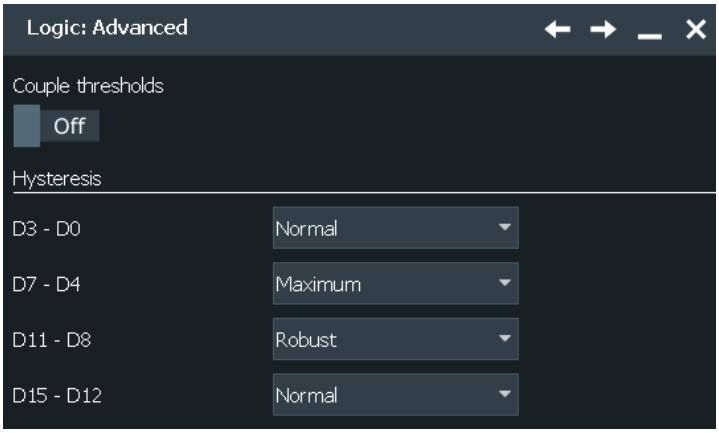
The range of threshold levels and the minimum voltage swing is given in the data sheet.

"Threshold" Enter the value directly in the field.

"Technology" Selects the threshold voltage for various types of integrated circuits from a list and applies it to all digital channels. The value is set to "Manual" if a user-defined threshold was entered directly.

Remote command:
`BUS<m>:PARallel:TECHnology` on page 2315 (all buses)
`DIGital<m>:TECHnology` on page 2310 (bus1)
`BUS<m>:PARallel:THReshold<n>` on page 2314 (all buses)
`DIGital<m>:THReshold` on page 2311 (bus1)

**Advanced**
Opens a dialog for advanced threshold and hysteresis settings. You can define them additional to the "Threshold" and "Technology".




**Couple thresholds ← Advanced**
Sets the threshold and the hysteresis for all digital channels and all buses to the same value.

Remote command:
BUS<m>:PARallel:THCoupling on page 2315 (all buses)
DIGital<m>:THCoupling on page 2311 (bus1)

**Hysteresis ← Advanced**
Defines the size of the hysteresis for the respective channels.

"Normal" The instrument sets a small value suitable for the signal and its settings. Use this setting for clean signals.

"Maximum" The instrument sets the maximum value that is possible and useful for the signal and its settings. Use this setting for noisy signals.

"Robust" Sets different hysteresis values for falling and rising edges to avoid an undefined state of the trigger system. Use this setting for very noisy signals.

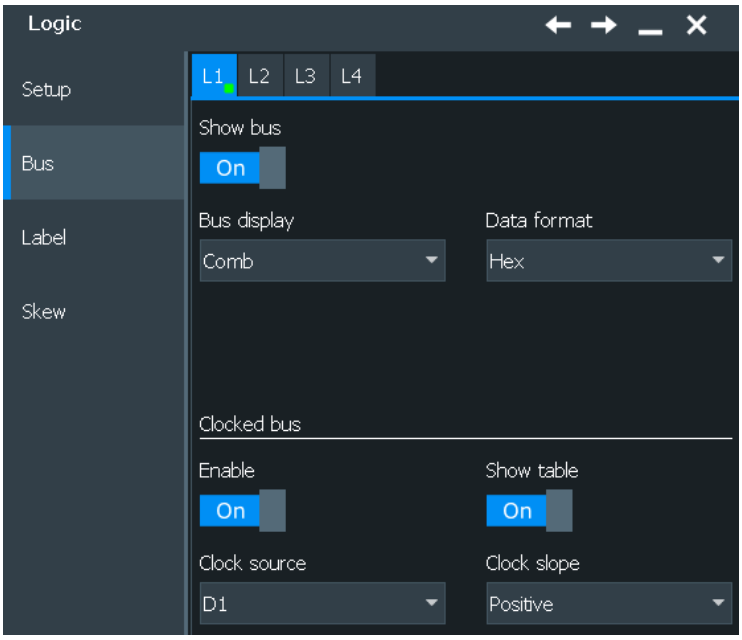
Remote command:
BUS<m>:PARallel:HYSTeresis<n> on page 2315 (all buses)
DIGital<m>:HYSTeresis on page 2311 (bus1)

**Activate channels**
Selects the state of the respective bus channel.

**Enable/Disable D7-D0, D15-D18**
The buttons select or deselect all digital channels of a pod at once.

### 14.1.1.2 Bus

Access: "Menu" > "Logic" > "Bus" tab



**Show bus**
If enabled, the resulting bus signal and bus values are displayed in the diagram. Select the presentation type for the bus signal with "Bus display" on page 942.

Remote command:
BUS<m>:PARallel:DISPlay:SHBU on page 2318

**Bus display**
Defines how the parallel bus is displayed.

"Comb" Displays the decoded bus signal with bus values. When at least one digital channel changes its value, the bus value changes too.

"Analog" Displays the bus values as signal amplitudes, similar to an analog waveform. Thus, a quasi-analog waveform is created.

Remote command:
BUS<m>:PARallel:DISPlay:BTYP on page 2318

**Data format**
Sets the data format of bus values, which are displayed in the decode table and on the comb bus display. Available formats are: Hex, ASCII, octal, binary, signed, and unsigned.

Signed and Unsigned are integer data types with maximum 16-bit length. Unsigned is used for positive integers. Signed is used for positive and negative integers.

If the target file format is BIN, you can save only signed and unsigned binary data. The data format "Signed" writes signed data; all other formats are saved as unsigned binary data.

Remote command:
BUS<m>:PARallel:DATA:FORMat on page 2317

**Clocked bus**
If a bus is a clocked bus, one of the digital channels serves as clock of the bus.

For an unclocked bus, the logical state of the bus is determined for each sample. For a clocked bus, the logical state is determined only at the specified clock edges.

The settings are only available for "Bus1" and "Bus2".

**Enable ← Clocked bus**
Enable this option for a clocked bus.

Remote command:
BUS<m>:PARallel:CLON on page 2318

**Show table ← Clocked bus**
The decode table is only available for clocked buses to check the data words.

If enabled, a results box opens with decoded values of the bus signal and its time. Each clock edge corresponds to one row in the table.

Remote command:
BUS<m>:PARallel:DECTable<n>:SHOW on page 2320
BUS<m>:PARallel:DECTable<n>:COUNt? on page 2320
BUS<m>:PARallel:DECTable<n>:DATA? on page 2320

**Clock source ← Clocked bus**
Selects the digital channel used as clock.

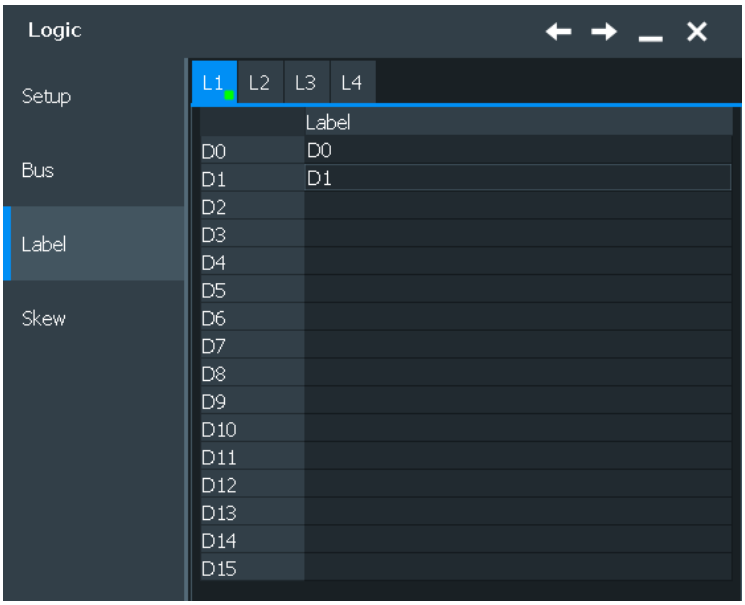
Remote command:
`BUS<m>:PARallel:CLOCk` on page 2319

**Clock slope ← Clocked bus**
Selects the slope of the clock signal at which all digital channels of the bus are analyzed.

Remote command:
`BUS<m>:PARallel:CLSLope` on page 2319

#### 14.1.1.3 Label settings

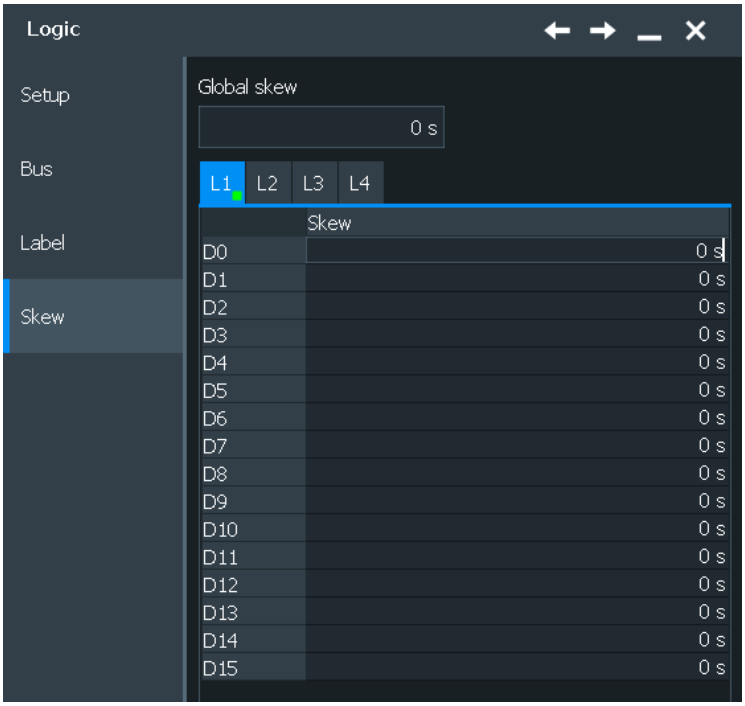
Access: "Menu" > "Logic" > "Label" tab




In this tab, you can enter a name for each digital channel. The name is displayed in the diagram.

#### 14.1.1.4 Skew settings

Access: "Menu" > "Logic" > "Skew" tab




**Global skew**
Sets a general delay for all digital channels. The resulting skew of a digital channel is the sum of the general "Global skew" and the individual "Skew".

Remote command:
`BUS<m>:PARallel:DESoffset` on page 2316

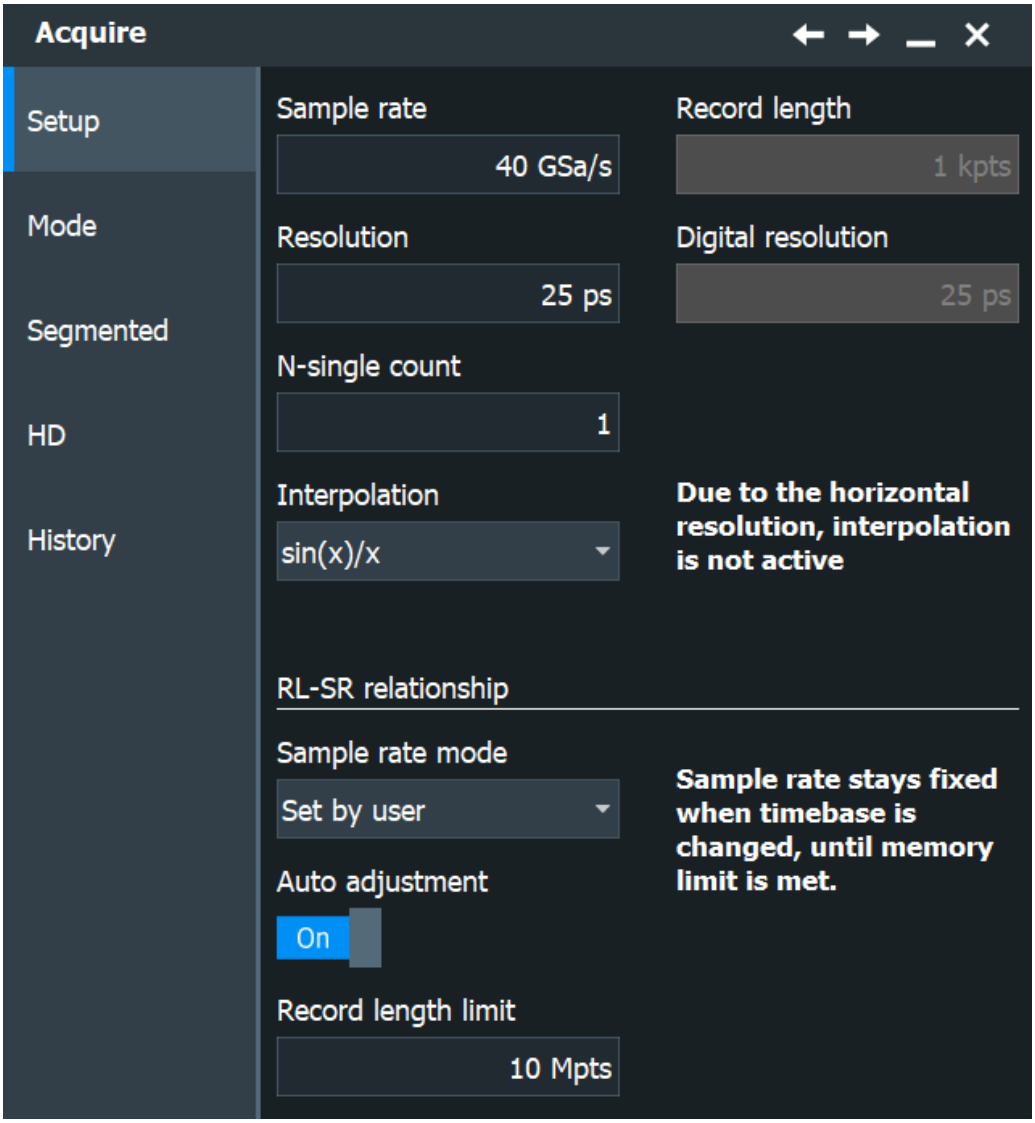
**Skew**
Sets an individual delay for each digital channel to time-align it with other digital channels. The skew value compensates delays that are known from the circuit specifics or caused by the different length of cables. The skew between the probe boxes of the digital channels and the probe connectors of the analog channels is automatically aligned by the instrument.

Remote command:
`BUS<m>:PARallel:BIT<n>:DESKew` on page 2316 (all buses)
`DIGital<m>:DESKew` on page 2312 (Bus1)

## 14.1.2 Digital resolution

Access: "Menu" > "Acquire"

If an MSO option is installed and at least one digital channel is active, additional information appears on the "Setup" tab of the "Acquire" dialog box.




**Dig. resolution**
Shows the current digital resolution of the digital channels. The maximum digital record length is always 200 MSa per digital channel. This number is independent of the installed memory.

Remote command:
`ACQuire:DRESolution?` on page 2321

### 14.1.3 Using digital probes

**NOTICE**

**Ensuring accurate measurement results**

The mixed-signal-option (MSO, R&S RTP-B1) with connected probe leads is considered as a test probe, according to EN 61326-2-1, clause 5.2.4.101. Therefore, the measurements are sensitive to electromagnetic interference. Consider additional shielding methods to avoid interference.

Consider the following guidelines for good probing practices:

- The ground lead from each digital channel group (D15–D8 and D7–D0) should be attached to the ground of the DUT if any channel within the group is being used for data capture. The ground lead improves signal fidelity to the oscilloscope, ensuring accurate measurements.
- For high-speed timing measurements (rise time < 3 ns), each digital channel probe should use its own ground.

1. Connect the digital probe cable to any of the MSO connectors on the rear panel of the instrument as shown on the Documentation Card delivered with the digital probe.
2. Connect the ground lead on each set of channels (each pod) with a probe grabber.
3. Connect a grabber to one of the probes leads.
4. Connect the grabber to a node in the circuit that you want to test.
5. For high-speed signals, connect a ground lead to the probe lead. Connect the ground lead to ground in the device under test.
6. Repeat these steps until you have connected all points of interest.

### 14.1.4 Configuring digital channels and parallel buses

The configuration of a parallel bus includes the selection and setup of the digital channels, the configuration of the bus display, and, if necessary, the clock configuration.

For a detailed description of the settings, see Chapter 14.1.1, "Parallel buses - configuration", on page 938.

1. Open "Menu" > "Logic" > "Setup".
2. Enable the digital channels to be displayed and included in the bus.
   To enable or disable all channels of a pod at once, tap "Enable D7-D0" or "Enable D15-D8".

   Enabling one or more channels also enables the display of the signals - "Show channels", and enables the parallel bus. If another active bus already uses the same digital channel, the instrument disables this bus and shows a message.

The digital signals are shown in the diagram, and the signal icon of the parallel bus appears on the signal bar. Using this bus icon, you can minimize, arrange, and switch off the bus together with its channels in the same way as you do with any waveform.

3. Set the logical thresholds as described in Chapter 14.1.5, "Setting the logical thresholds", on page 947.
4. If the bus has a clock signal, open the "Bus" tab.
   Enable "Clocked bus" and select the "Clock source" and "Clock slope".
5. If necessary, enter a "Label" for each digital channel in the "Label" tab.
6. If necessary, enter a "Skew" value to time-align the channel in the "Skew" tab.

   Now the configuration of the parallel bus is completed.

### 14.1.5 Setting the logical thresholds

For a detailed description of the settings, see Chapter 14.1.1.1, "Setup", on page 939.
Threshold settings are the same for all *active* parallel buses.

1. Open "Menu" > "Logic" > "Setup".
2. To set the thresholds, use one of the following ways:
   a) Use the same value for all digital channels and all parallel buses:
      - In the "Logic: Advanced" dialog, enable "Coupe thresholds".
      - Open the "Setup" tab. To set the threshold, either select a predefined "Technology" value or enter a user-defined value.

   b) Set different thresholds for the individual channel groups:
      - In the "Logic: Advanced" dialog, disable "Coupe thresholds".
      - Set different thresholds for each bus. Active buses use the same threshold and hysteresis values.
3. In the "Logic: Advanced" dialog, set the "Hysteresis" for each threshold to avoid the change of signal states due to noise.

## 14.2 Display

You can adjust the display of the parallel bus signals and the individual digital channels to optimize the analysis of bus data:

- Show the digital channels which are assigned to the bus, drag them to the optimal position, and scale them
- Show the decoded bus signal in different ways:
  – comb display with numeric bus values
  – analog display with bus values as amplitudes (quasi-analog waveform)

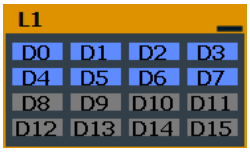
You can also drag the bus waveforms on the display and scale them.

- Show the result box of the decoded clocked bus signal

Each parallel bus is shown in a separate diagram, and the diagrams can be minimized and arranged as usual.

The signal icon indicates the activities on the digital channels even if they are not displayed in the diagram, of if the acquisition has been stopped:

- Blue: channel is low
- Green: channel is high
- Gray: channel state is changing




The display update rate of the oscilloscope is adapted to the visual perception of human eyes, and it is slower than the acquisition rate. All analog and digital waveforms that are acquired during one display update cycle are overlapped and displayed at once. Thus you can see the cumulative occurrence of binary states and edge transitions on the screen at once. Bus signals are not overlapped.

The trigger point is always visible on the display, it cannot be moved outside ("Restrict horizontal position to acquisition range" is enabled automatically).

If digital channels are active, the trigger point is always visible on the display, it cannot be moved outside.

To access and analyze one or more specific acquisitions, you can use the History Viewer in the common way.

Furthermore, you can zoom in digital signals and bus signal in the same way as in analog waveforms.

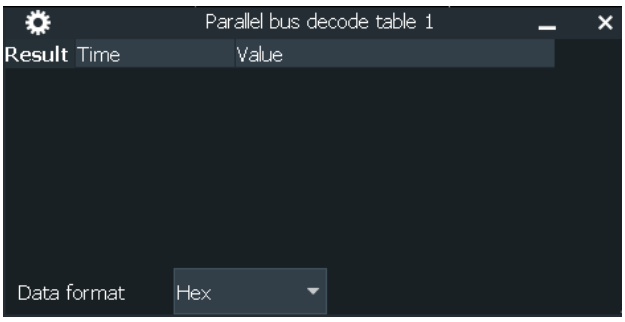
See also:

- Chapter 7.1, "Zoom", on page 261

### 14.2.1 Parallel bus - decode table

Decoding is available for clocked parallel buses.

The decode table shows the decoded data words of the bus signal and the corresponding time. Each clock edge corresponds to one row in the table. Below the table, you can select the data format of the bus values.




The results can be saved to a `.csv` or `.html` file, see Chapter 12.2.4, "Results", on page 499.

### 14.2.2 Adjusting the display of digital channels and parallel buses

The display of digital channels and parallel buses is flexible, you can adjust it to your needs by combining the following settings:

1. Enable "Show bus" if you want to display the bus signal in the diagram. Under "Bus representation", select if you want to display the decoded bus signal with bus values ("Comb"), or show the bus values as amplitudes, similar to an analog waveform ("Analog").
2. Check the signal icon of the bus to monitor the activities on the digital channels even if they are not displayed in the diagram:
   - Blue: channel is low
   - Green: channel is high
   - Gray: channel state is changing
3. In the diagram, you can change the display order of the digital channels by dragging the individual channels to the required position.
4. To adjust the line height and vertical position of all digital channels at once, tap one of the digital channels and turn the vertical [Scale] and [Position] rotary knobs. In the same way, you can move and scale the bus signal.
5. If the bus signal is displayed as quasi-analog waveform, you can double-tap the waveform to open the "Parallel buses" dialog box.
6. To switch off the display of the digital channels, disable "Show signals".

## 14.3 Trigger

For digital trigger sources are all trigger types useful that require only one trigger level as trigger condition. This level is the logical threshold. Possible trigger sources are the individual digital channels, parallel bus signals, or any logical combination of digital channels. The following trigger types are available:

*Table 14-1: Trigger types and digital trigger sources*

| Trigger type | Trigger source is | | |
|---|---|---|---|
| | Digital channel | Logic combination of digital channels | Parallel bus |
| Edge | X | X | |
| Width | X | X | |
| Timeout | X | X | |
| Setup & Hold | X | | X |
| State | | X | X |
| Pattern (with holdoff) | | X | X |
| Serial Pattern | X | X | |

In firmware versions < 5.00, the Setup & Hold was called Data2Clock trigger.

For details, see: Chapter 14.3.1, "Trigger settings for digital signals and parallel buses", on page 950.

Also, you can define trigger holdoff conditions. See Chapter 6.4, "Mode / Holdoff", on page 225.

## 14.3.1 Trigger settings for digital signals and parallel buses

Depending on the selected source, the instrument provides the appropriate trigger types and the corresponding trigger settings.

The settings in the "Setup" tab are:


- Basic trigger settings....950
- Edge....951
- Width....952
- Timeout....954
- Setup & Hold....955
- State....957
- Pattern....958
- Serial pattern....960


### 14.3.1.1 Basic trigger settings

The basic trigger settings for MSO are the trigger source and the trigger type. They are selected in the upper part of the "Trigger" dialog box.

Make sure that the trigger sequence is set to "A only".

Also, you can define trigger holdoff conditions. See Chapter 6.4, "Mode / Holdoff", on page 225.

**Source**
If the Mixed Signal Option is installed, the variety of trigger sources is enhanced with specific digital trigger sources. As a trigger source you can select:

- One of the digital channels "D0" ... "D15"

- A logic combination of digital channels: "Logic"

- One of the parallel buses "Par. bus1" ... "Par. bus4"

Remote command:
`TRIGger<m>:SOURce[:SELect]` on page 1305

**Type**
Depending on the selected source, the instrument provides the appropriate trigger types and the corresponding trigger settings. For mixed signal analysis, the following trigger types are available:

- Edge, see page 951
- Width, see page 952
- Timeout, see page 954
- Setup & Hold, see page 955
- State, see page 957
- Pattern, see page 958
- Serial pattern, see page 960

Remote command:
`TRIGger<m>:PARallel:TYPE` on page 2323

#### 14.3.1.2 Edge

Using the edge trigger, you can also trigger on a single digital channel (a logic bit), and a logical combination of digital channels.

Depending on the selected trigger source, different trigger settings are available. The trigger level is already set - in MSO the logical threshold is used as trigger level.

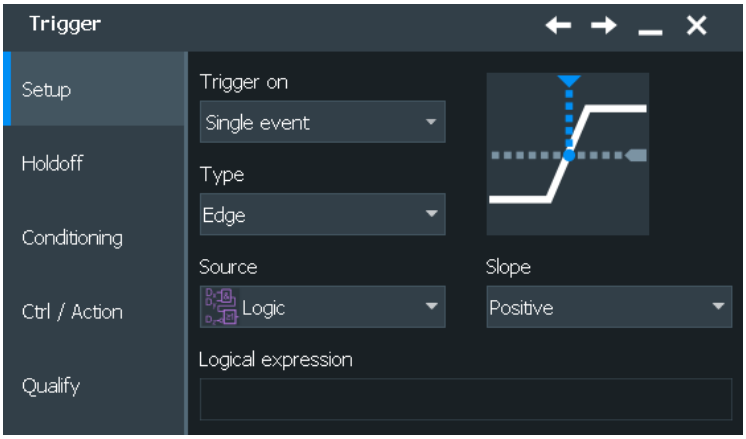



*Figure 14-1: Edge trigger settings for trigger source = logical combination of digital channels (Logic)*

**Slope**
Defines the edge - the state transition - of the signal.

"Positive" Means a 0 to 1 transition of the state.

"Negative" Means a 1 to 0 transition of the state.

"Either" Triggers on any activity on the selected trigger source.

Remote command:
`TRIGger<m>:PARallel:EDGE:SLOPe` on page 2324

**Logical expression**
Defines a logical combination of several digital channels as trigger condition if "Logic" is set for "Source".

If the "Slope" is positive, the trigger occurs when the logical expression becomes true. If the "Slope" is negative, the trigger occurs when the logical expression becomes false.

Remote command:
`TRIGger<m>:PARallel:EDGE:EXPRession[:DEFine]` on page 2323

#### 14.3.1.3 Width

The width trigger detects positive and/or negative pulses of a pulse width (duration) inside or outside of a defined time limit. It can trigger on a single digital channel or a logical combination of digital channels.

The instrument triggers at the end of the detected pulse.

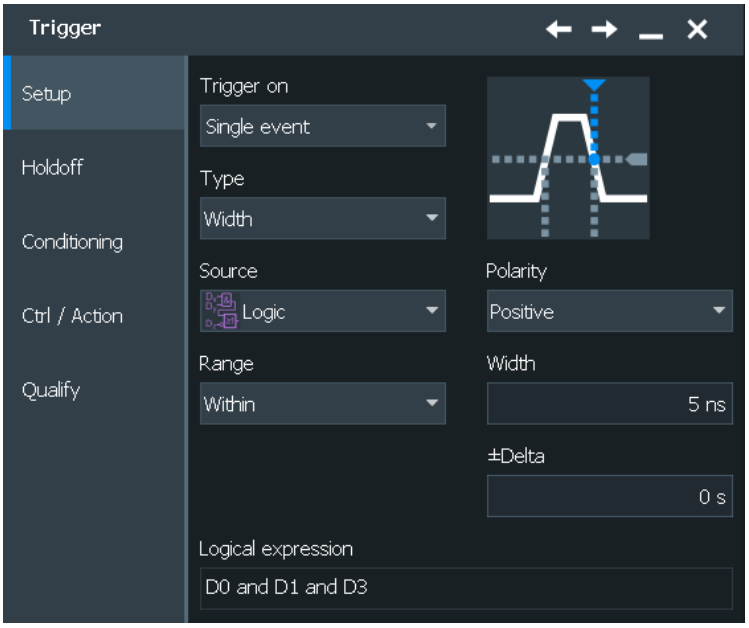



*Figure 14-2: Width trigger settings for trigger source = logical combination of digital channels*

**Polarity**
Sets the polarity of a pulse to "Positive", "Negative", or "Both".

When triggering on a positive pulse, the trigger event occurs on the high to low transition of the pulse if the timing condition is true. When triggering on a negative pulse, the trigger event occurs on the low to high transition of the pulse if the timing condition is true.

Remote command:
`TRIGger<m>:PARallel:WIDTh:POLarity` on page 2324

**Logical expression**
Defines a logical combination of several digital channels as trigger condition if "Logic" is set for "Source". As long as the digital signals match the logical expression (true), the pulse is positive. Otherwise, the pulse is negative.

Remote command:
`TRIGger<m>:PARallel:WIDTh:EXPRession[:DEFine]` on page 2323

**Range**
Selects how the range of a pulse width is defined:

"Within" Triggers on pulses inside a given time range. The time limit is defined by *Width ± Delta*.

"Outside" Triggers on pulses shorter or longer than a given time range. The time limit definition is the same as for "Within" range.

"Shorter" Triggers on pulses shorter than the given "Width".

"Longer" Triggers on pulses longer than the given "Width".

Remote command:
`TRIGger<m>:PARallel:WIDTh:RANGe` on page 2325

**Width**
For the ranges "Shorter" and "Longer", the width defines the maximum and minimum pulse width, respectively.

For the ranges "Within" and "Outside", the width defines the center of a range which is defined by the limits "±Delta".

Remote command:
`TRIGger<m>:PARallel:WIDTh:WIDTh` on page 2325

**±Delta**
Defines a range around the given width value.

The combination "Range" = Within and "±Delta" = 0 triggers on pulses with a pulse width that equals "Width".

The combination "Range" = Outside and "±Delta" = 0 means to trigger on pulse widths ≠ "Width".

Remote command:
`TRIGger<m>:PARallel:WIDTh:DELTa` on page 2325

### 14.3.1.4 Timeout

The timeout trigger event checks if the trigger source signal stays above or below the threshold voltage for a specified time lapse. In other words, the event occurs if the state condition remains unchanged for the specified time.

You can use the timeout trigger on a single digital channel, or a logical combination of digital channels.

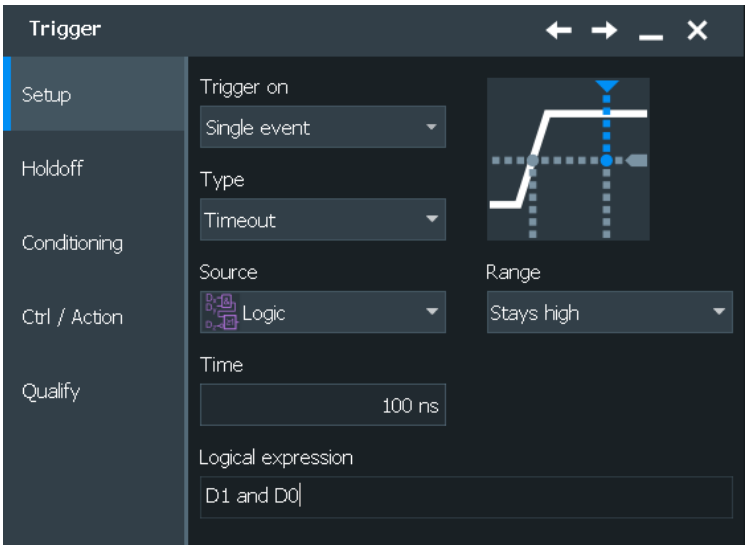



*Figure 14-3: Timeout trigger settings for trigger source = logical combination of digital channels*

**Range**
Sets the state condition:

"Stays high" The level of a digital channel stays above the threshold, or the logical expression for "Logic" trigger source is true.

"Stays low" The level of a digital channel stays below the threshold, or the logical expression for "Logic" trigger source is false.

"High or low" The signal state remains unchanged.

Remote command:
`TRIGger<m>:PARallel:TIMeout:RANGe` on page 2326

**Time**
Defines the time limit for the timeout at which the instrument triggers.

Remote command:
`TRIGger<m>:PARallel:TIMeout:TIME` on page 2326

**Logical expression**
Defines a logic combination of several digital channels as trigger condition if "Logic" is set for "Source". The "Qualification Editor" supports the entry of the expression.

Remote command:
`TRIGger<m>:PARallel:TIMeout:EXPRession[:DEFine]` on page 2323
`TRIGger<m>:PARallel:STATe:EXPRession[:DEFine]` on page 2323
`TRIGger<m>:PARallel:PATTern:EXPRession[:DEFine]` on page 2323
`TRIGger<m>:PARallel:SPATtern:EXPRession[:DEFine]` on page 2323

#### 14.3.1.5 Setup & Hold

The Setup & Hold trigger event occurs when the state of the trigger source signal changes inside a given time before the clock edge (setup time) or after the clock edge (hold time). The trigger event occurs at the clock edge for which the setup and/or hold time was violated.

In firmware versions < 5.00, the Setup & Hold was called Data2Clock trigger.

With Setup & Hold trigger, you can trigger on a single digital channel, or a parallel bus to check several digital channels simultaneously. The clock signal is connected to one of the digital channels.

If you configure this trigger type for a parallel bus, the bus configuration is adjusted by the instrument if necessary. The bus is defined as clocked bus, and the clock source of the trigger is set as clock source of the bus.

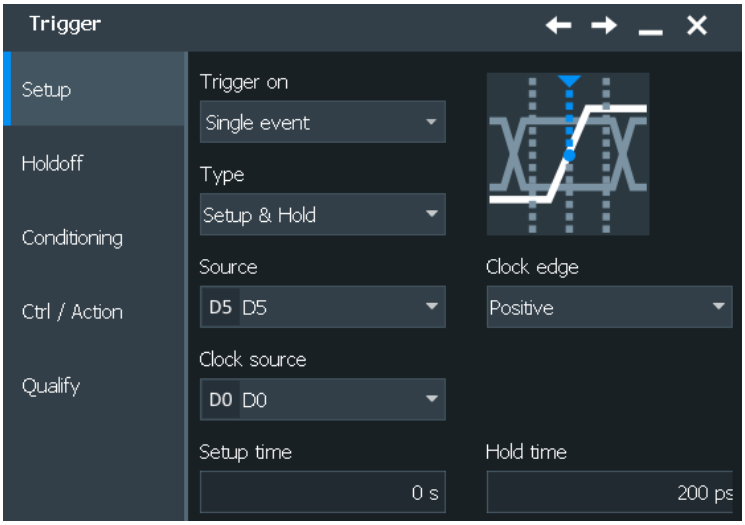



***Figure 14-4: Setup & Hold trigger settings***

**Clock source**
Selects the digital channel of the clock signal.

Remote command:
`TRIGger<m>:PARallel:SETHold:CSOurce[:VALue]` on page 2323
`TRIGger<m>:PARallel:STATe:CSOurce:VALue` on page 2323
`TRIGger<m>:PARallel:SPATtern:CSOurce[:VALue]` on page 2323

**Clock edge**
Sets the edge of the clock signal. The crossing of the clock edge and the logical threshold defines the time reference point for the setup and hold time measurement.

Remote command:
`TRIGger<m>:PARallel:SETHold:CSOurce:EDGE` on page 2327

**Setup time**
Sets the minimum time *before* the clock edge while data is stable and not change its state.

The setup time can be negative. In this case, the setup interval starts after the clock edge, and the hold time starts after the setup time has expired. Thus, the hold time is always positive. If you change the negative setup time, the hold time is adjusted by the instrument.

Remote command:
`TRIGger<m>:PARallel:SETHold:STIMe` on page 2327

**Hold time**
Sets the minimum time *after* the clock edge while data is stable and not change its state.

The hold time can be negative. In this case, the hold time ends before the clock edge, and the setup interval ends when the hold interval starts. Thus, the setup time is always positive. If you change the negative hold time, the setup time is adjusted by the instrument.

Remote command:
TRIGger<m>:PARallel:SETHold:HTIMe on page 2327

#### 14.3.1.6 State

The state trigger detects the logical state of several logically combined digital channels at a given clock edge. The trigger source is a logical combination of digital channels or a parallel bus. The trigger occurs at the clock edge at which the state condition is true.

*Figure 14-5: State trigger settings for trigger source = parallel bus*

**Clock source**
Selects the digital channel of the clock signal.

Remote command:
TRIGger<m>:PARallel:SETHold:CSOurce[:VALue] on page 2323
TRIGger<m>:PARallel:STATe:CSOurce:VALue on page 2323
TRIGger<m>:PARallel:SPATtern:CSOurce[:VALue] on page 2323

**Clock edge**
Sets the edge of the clock signal. The crossing of the clock edge and the logical threshold defines the time at which the logical states and the bus value are analyzed.

Remote command:
TRIGger<m>:PARallel:STATe:CSOurce:EDGE on page 2328

**Channel states**
Available for "Trigger Source"= "Par. Bus1/2/3/4"

For each digital channel that is used in the bus, set the required state: 1, 0, or X (don't care).

Remote command:
TRIGger<m>:PARallel:STATe:BIT<0..15> on page 2328

**Logical expression**
Defines a logic combination of several digital channels as trigger condition if "Logic" is set for "Source". The "Qualification Editor" supports the entry of the expression.

Remote command:
TRIGger<m>:PARallel:TIMeout:EXPRession[:DEFine] on page 2323
TRIGger<m>:PARallel:STATe:EXPRession[:DEFine] on page 2323
TRIGger<m>:PARallel:PATTern:EXPRession[:DEFine] on page 2323
TRIGger<m>:PARallel:SPATtern:EXPRession[:DEFine] on page 2323

### 14.3.1.7 Pattern

The pattern trigger identifies a logical state of several logically combined digital channels (pattern) and a time limitation (holdoff). The pattern definition is defined by the logical expression, if "Logic" is used for trigger source. For a parallel bus trigger source, the pattern is defined by setting the state of each digital channel.

The timing starts when the pattern comes true. The decision level is the logical threshold.

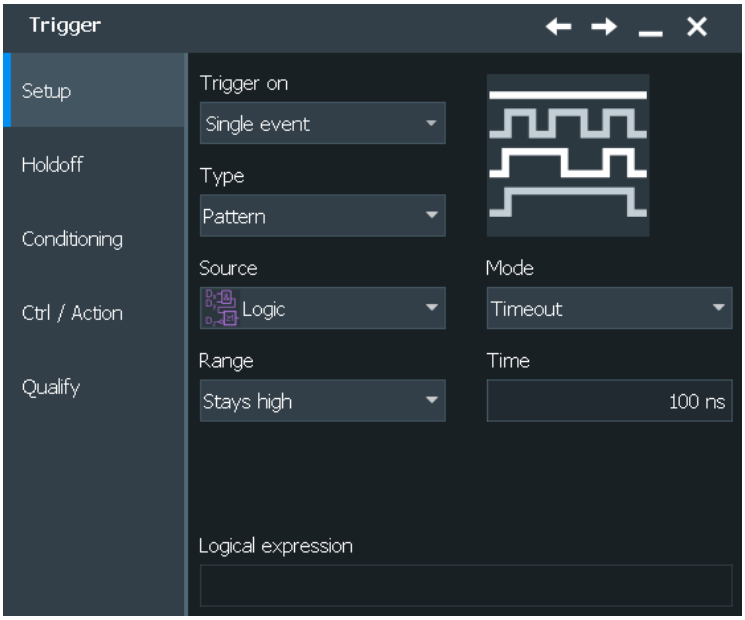



*Figure 14-6: Pattern trigger settings for trigger source = parallel bus and timeout*

**Channel states**
Available for "Trigger Source"= "Par. Bus1/2/3/4"

For each digital channel that is used in the bus, set the required state: 1, 0, or X (do not care).

Remote command:
TRIGger<m>:PARallel:PATTern:BIT<0..15> on page 2328

**Logical expression**
Defines a logic combination of several digital channels as trigger condition if "Logic" is set for "Source". The "Qualification Editor" supports the entry of the expression.

Remote command:
`TRIGger<m>:PARallel:TIMeout:EXPRession[:DEFine]` on page 2323
`TRIGger<m>:PARallel:STATe:EXPRession[:DEFine]` on page 2323
`TRIGger<m>:PARallel:PATTern:EXPRession[:DEFine]` on page 2323
`TRIGger<m>:PARallel:SPATtern:EXPRession[:DEFine]` on page 2323

**Mode: Off, Timeout, Width**
Sets the mode of the timing condition.

"Off" No timing condition, only the logical pattern condition is relevant.

"Timeout" Defines a minimum time qualification to avoid triggering on unstable or transitional conditions. Even in best-designed systems, there are slight delays between the signal when digital signals change states. This means that there are always transitional state conditions when signals are switching.
See "Timeout settings" on page 959 for a description of the settings.
The trigger event occurs when the pattern stays unchanged for the specified time.

"Width" Sets a pulse width as timing condition, see "Width settings" on page 960. The pulse starts when the pattern comes true, and the trigger event occurs when the pattern comes false during the specified time limit.
Using this mode, you can, for example, trigger exclusively on unstable conditions - if the pattern is present for less than a specified time.

Remote command:
`TRIGger<m>:PARallel:PATTern:MODE` on page 2329

**Timeout settings**
The timeout settings "Range" and "Time" appear if the timing mode is set to "Timeout".

**Range ← Timeout settings**
Sets the state condition:

"Stays high" The pattern stays true for the specified time.

"Stays low" The pattern stays false for the specified time.

"High or low" The pattern remains unchanged for the specified time.

Remote command:
`TRIGger<m>:PARallel:PATTern:TIMeout:MODE` on page 2329

**Time ← Timeout settings**
Defines the time limit for the timeout at which the instrument triggers.

Remote command:
`TRIGger<m>:PARallel:PATTern:TIMeout[:TIME]` on page 2329

**Width settings**
The width settings "Range", "Width" and "±Delta" appear if the timing mode is set to "Width".

**Range ← Width settings**
Selects how the range of a pulse width is defined:

"Within" Triggers when the pattern comes false inside a given time range. The time limit is defined by *Width ± Delta*.

"Outside" Triggers when the pattern comes false before or after the given time range. The time limit definition is the same as for "Within" range.

"Shorter" Triggers when the pattern comes false before the given "Width" has expired.

"Longer" Triggers when the pattern comes false after the given "Width" has expired.

Remote command:
`TRIGger<m>:PARallel:PATTern:WIDTh:RANGe` on page 2330

**Width ← Width settings**
For the ranges "Shorter" and "Longer", the width defines the maximum and minimum time limit, respectively.

For the ranges "Within" and "Outside", the width defines the center of a range which is defined by the limits "±Delta".

Remote command:
`TRIGger<m>:PARallel:PATTern:WIDTh[:WIDTh]` on page 2330

**±Delta ← Width settings**
Defines a range around the given width value.

The combination "Range" = Within and "±Delta" = 0 triggers on pulses with a pulse width that equals "Width".

The combination "Range" = Outside and "±Delta" = 0 means to trigger on pulse widths ≠ "Width".

Remote command:
`TRIGger<m>:PARallel:PATTern:WIDTh:DELTa` on page 2331

#### 14.3.1.8 Serial pattern

The serial pattern trigger identifies a serial bit string trigger on a single digital channel, or for a logical combination of digital channels. The trigger requires a clocked bus; the bits are read at the specified clock edge. The trigger event occurs at the last clock edge of the serial bit string.

This trigger type allows you to trigger on specific address or data transmissions in serial input and output signals.

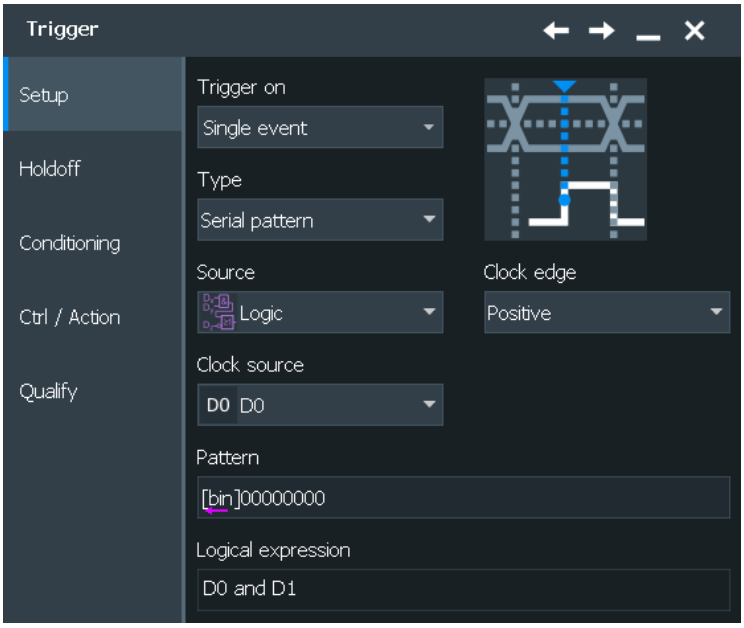



*Figure 14-7: Serial pattern trigger settings for trigger source = logical combination of digital channels*

**Clock edge**
Sets the edge of the clock signal. The bit value is determined at the crossing of the clock edge and the logical threshold.

Remote command:
`TRIGger<m>:PARallel:SPATtern:CSOurce:EDGE` on page 2331

**Clock source**
Selects the digital channel of the clock signal.

Remote command:
`TRIGger<m>:PARallel:SETHold:CSOurce[:VALue]` on page 2323
`TRIGger<m>:PARallel:STATe:CSOurce:VALue` on page 2323
`TRIGger<m>:PARallel:SPATtern:CSOurce[:VALue]` on page 2323

**Logical expression**
Defines a logic combination of several digital channels as trigger condition if "Logic" is set for "Source". The "Qualification Editor" supports the entry of the expression.

Remote command:
`TRIGger<m>:PARallel:TIMeout:EXPRession[:DEFine]` on page 2323
`TRIGger<m>:PARallel:STATe:EXPRession[:DEFine]` on page 2323
`TRIGger<m>:PARallel:PATTern:EXPRession[:DEFine]` on page 2323
`TRIGger<m>:PARallel:SPATtern:EXPRession[:DEFine]` on page 2323

**Pattern**
Defines the serial bit string on which to trigger. Touch and hold the "Pattern" field to open the "Bit Pattern Editor" where you can enter the pattern in various formats. The pattern has to be defined exactly, X (do not care) is not supported in binary format.

See also: Chapter 13.1.7, "Bit pattern editor", on page 524

Remote command:
TRIGger<m>:PARallel:SPATtern:PATTern on page 2331

### 14.3.2 Triggering on digital signals and parallel buses

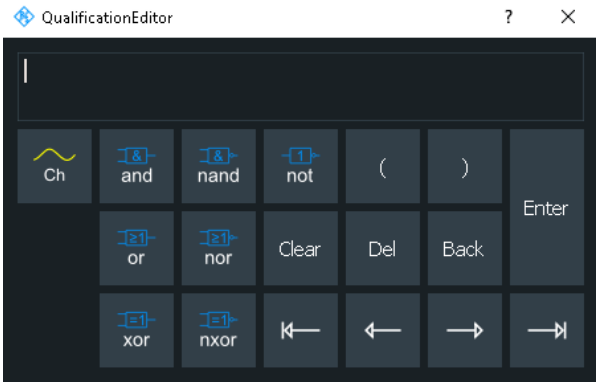
For a detailed description of the settings, see Chapter 14.3.1, "Trigger settings for digital signals and parallel buses", on page 950.

1. Open "Menu">"Trigger" > "Setup" tab
2. Select the trigger "Source":
   - One of the digital channels "D0" ... "D15"

     
   - A logic combination of digital channels: "Logic"

     
   - One of the parallel buses "Par. bus1" ... "Par. bus4"

     
3. Select the trigger "Type".
4. Under "Trigger type dependent settings", configure the trigger.
5. For trigger source "Logic", enter the logical expression of the digital channel combination. Tap and hold the "Logical expression" field until the "Qualification Editor" opens. It provides all logic operators that can be used in the expression.




## 14.4 Measurements on digital channels

For measurements on digital channels, the number of measurement categories and types is reduced to applicable measurement.

These are:

- Period
- Frequency
- Positive and negative pulse
- Pulse count
- Delay
- Phase
- Positive and negative duty cycle
- Burst width
- Edge count

Except for delay measurement, the measurements have the same settings and results for analog and digital sources.

Delay measurement on digital channels is reduced to measure the time between two subsequent rising or two subsequent falling edges.

See also Chapter 8.2, "Automatic measurements", on page 319.

## 14.5 Data export

The data of digital channels and parallel buses can be saved to file in the same way as analog waveform data. One digital channel or bus per file can be saved.

The data format of the stored values is defined with Menu > "Logic" > "Bus" tab > "Data format".

If the data is written to XML or CSV files, the selected format is used. If the target file format is BIN, you can save signed and unsigned binary data. The data format "Signed" writes signed data; all other formats are saved as unsigned binary data.

The export of raw data is not available.

### Export of a digital channel

If the data of digital channels is stored in BIN format, 1 bit is written for each sample. 8 data samples are written in 1 byte (data word). Thus, the file size is

*File size = Number of samples / 8*

For example, 100 MSa are written into a 12.5 MByte BIN file. After reading the file, you have to extract the samples from the data words.

If saved to BIN file, the digital channel can be imported as reference waveform.

### Export of a parallel bus

A parallel bus can be exported to file if "Enable bus" and "Show bus" are both activated, and the bus representation is "Comb".

All data formats can be saved to XML, CSV, and BIN files. If you save binary format to XML or CSV, you can see the values of each line for each sample.

In BIN files, 4 Bytes are written for each sample.

See also:

- Chapter 12.2.7, "Saving and loading waveform data", on page 505
- Chapter 12.2.1, "Waveform export files", on page 483

Remote commands for export to file:

- `EXPort:WAVeform:SOURce` on page 1614
- `EXPort:WAVeform:NAME` on page 1616
- `EXPort:WAVeform:SAVE` on page 1617

Remote commands for remote data transfer:

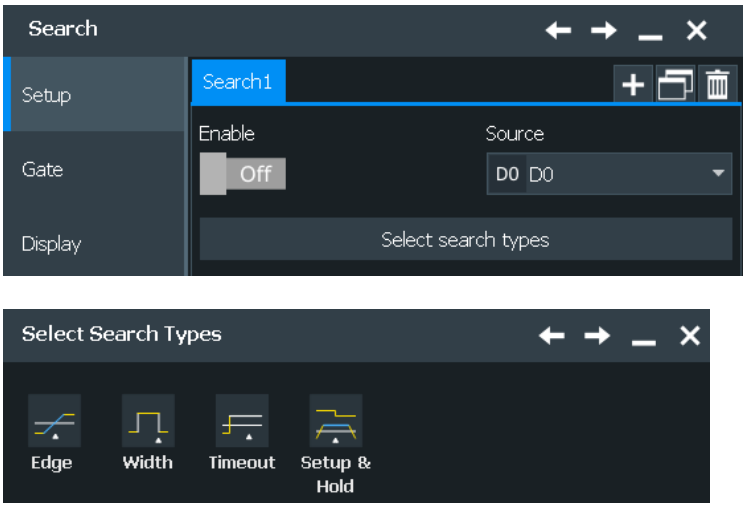
- `BUS<m>:PARallel:DATA:FORMat` on page 2317
- `BUS<m>:PARallel:DATA:HEADer?` on page 2333
- `BUS<m>:PARallel:DATA[:VALues]?` on page 2333
- `DIGital<m>:DATA:HEADer?` on page 2332
- `DIGital<m>:DATA[:VALues]?` on page 2332

## 14.6 Mathematics

A parallel bus that is displayed as quasi-analog waveform can be analyzed with FFT. To configure the FFT, use the "Advanced" mode and the formula editor.

## 14.7 Search

Access: "Menu" > "Apps" > "Analysis" tab > "Search".



It is also possible to search on digital channels for specified events. Search conditions use the same parameters as the trigger event definition, see Chapter 14.3.1, "Trigger settings for digital signals and parallel buses", on page 950.

You can search for edge, width, timeout, and Setup & Hold conditions.

# 15 Waveform generator (option R&S RTP-B6)

The R&S RTP includes an integrated waveform generator which can generate input signals and patterns during testing. You can setup and output two waveform generators and a pattern generator.

With each of the waveform generators, one can output simple functions, modulated sine waveform, arbitrary waveforms and sweep waveforms. It is possible to couple and synchronize the settings of the waveform generators.

The instrument preset does not affect the generator settings. Each generator has its own preset.

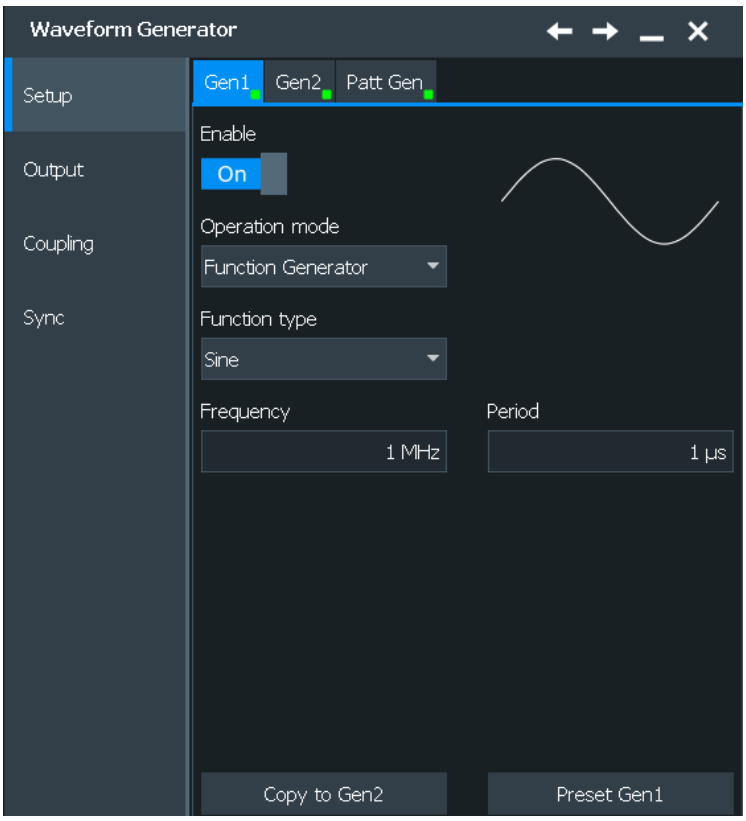
Always use double-shielded cables to prevent electromagentic interference (EMI) from impairing measurement results.

## 15.1 Setup of the waveform generator

Access: "Menu" > "Apps" > "Generators" > "Gen1" > "Setup" tab.

The "Setup" tab is divided into several sections:

- General settings like enabling and presetting the waveform generator
- Signal settings, depending on the selected Operation mode refer to one of the following:
  - Chapter 15.1.2, "Function generator", on page 968
  - Chapter 15.1.3, "Modulation", on page 973
  - Chapter 15.1.4, "Sweep", on page 979
  - Chapter 15.1.5, "Arbitrary", on page 980




Make sure that the tab of the correct waveform generator is selected on the left side.

The settings of the waveform generators are not affected by an instrument preset. Press "Preset Gen1/2" to preset the settings of the corresponding waveform generator.

### 15.1.1 General settings

The general waveform generator settings are for enabling and presetting the generator and selecting the "Operation Mode".

**Enable**
Enables the waveform generator/ pattern generator and outputs the signal to the connectors.

Remote command:
`WGENerator<m>[:ENABle]` on page 2334
`PGENerator:ENABle` on page 2349

**Operation mode**
Selects the operation mode for the waveform generator. The "Signal" settings depend on the selected mode.

For the settings of the different operation modes, refer to:

- Chapter 15.1.2, "Function generator", on page 968
- Chapter 15.1.3, "Modulation", on page 973
- Chapter 15.1.4, "Sweep", on page 979
- Chapter 15.1.5, "Arbitrary", on page 980

Remote command:
`WGENerator<m>:SOURce` on page 2334

**Copy settings from Gen1/Gen2**
Copies all settings from Gen1/Gen2 and applies them to Gen2/Gen1.

Remote command:
`WGENerator<m>:ACOPy` on page 2334

**Preset Gen1/Gen2/Patt Gen**
Sets the parameters of the generator to their default values. The settings of the generators are not affected by an instrument preset.They are also not stored in the user-defined preset.

Remote command:
`WGENerator<m>:PRESet` on page 2335
`PGENerator:PRESet` on page 2350

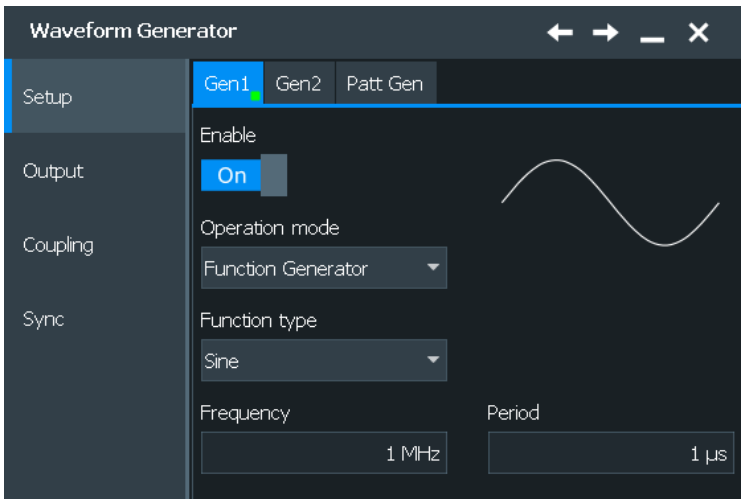
**Message**
Displays relevant messages concerning the coupling and sync settings.

## 15.1.2 Function generator

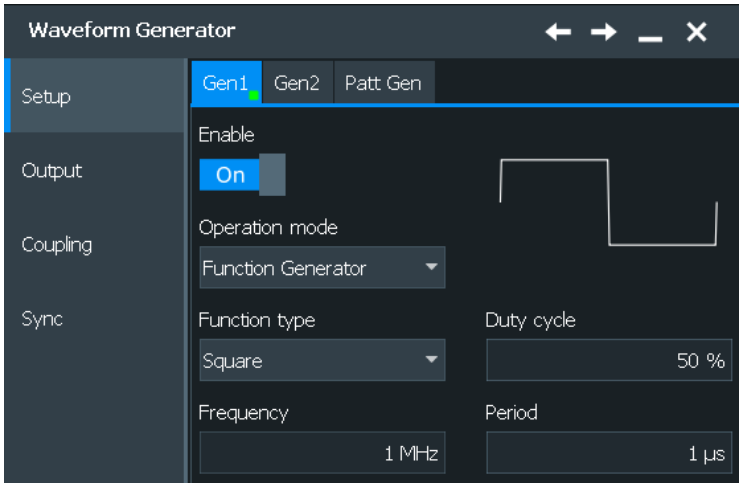
The R&S RTP function generator can generate input signals during testing. These signals can be used, for example, when testing circuits.

**Function type**
Selects the type of waveform to be generated for the function generator.

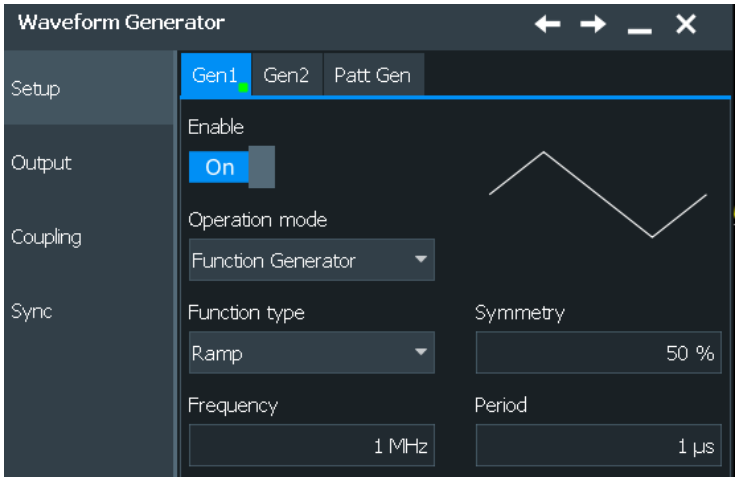
"Sine" Generates a sine wave. You can set its Frequency and Period.



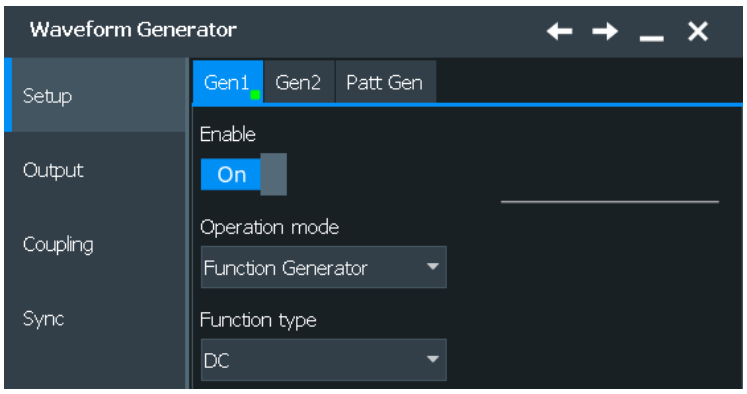
"Square" Generates a square wave. You can set its Frequency, Period and Duty cycle.



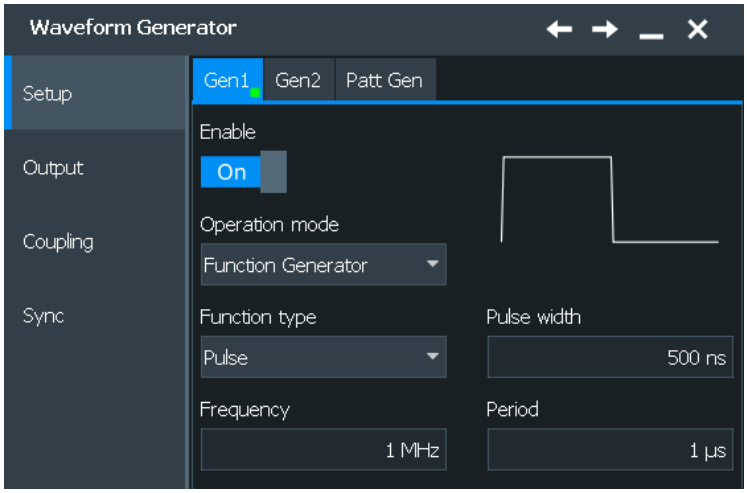

"Ramp" Generates a ramp signal. You can set its Frequency, Period and Symmetry.



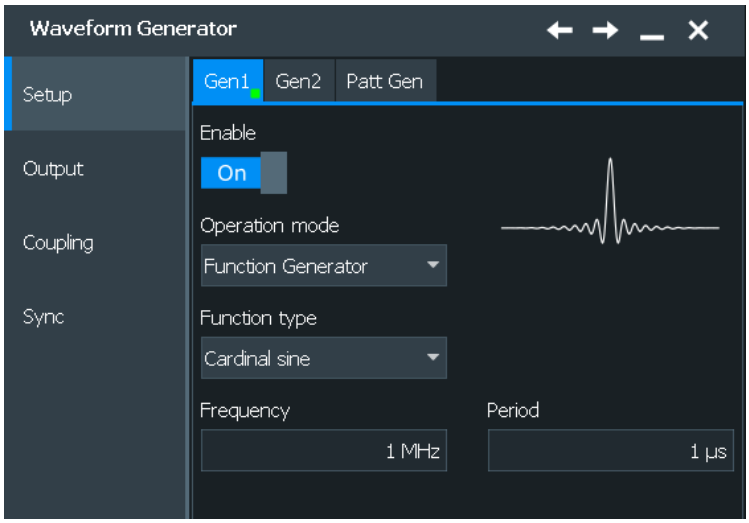

"DC" Generates a direct current (DC) signal. You can set the "Absolute level" on page 990.



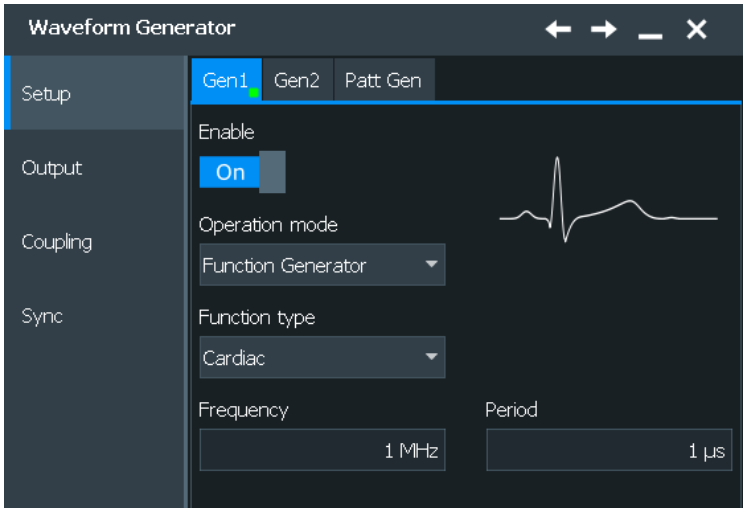
"Pulse" Generates a pulse signal. You can set the Frequency, Period and Pulse width.



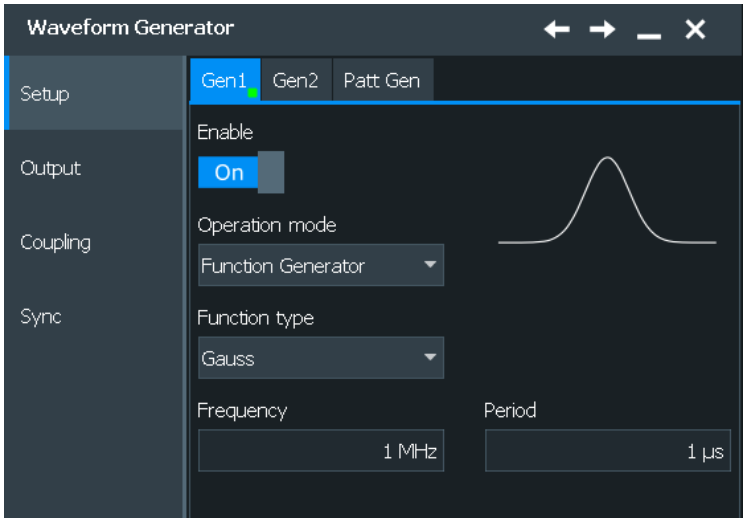

"Cardinal sine" Generates a cardinal sine wave. You can set the Frequency and Period.



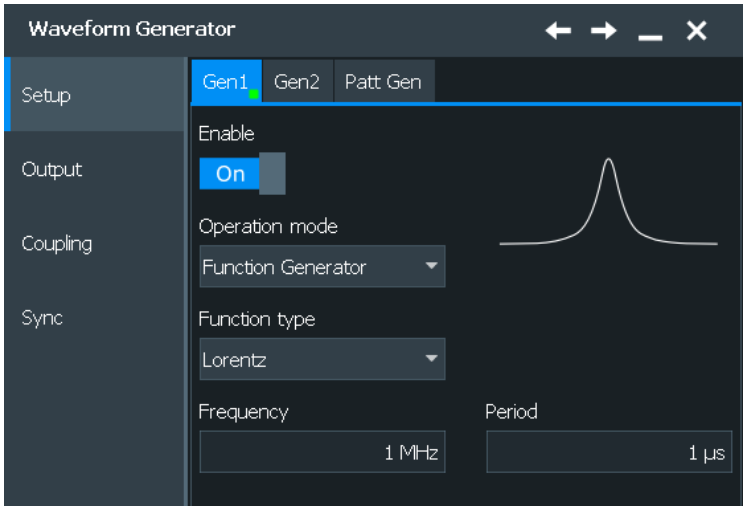
"Cardiac" Generates a cardiac signal. You can set the Frequency and Period.



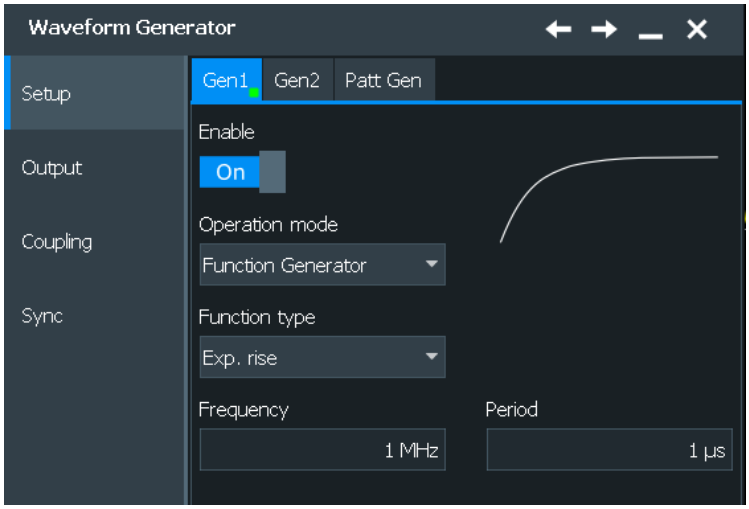

"Gauss" Generates a gauss signal . You can set the Frequency and Period.



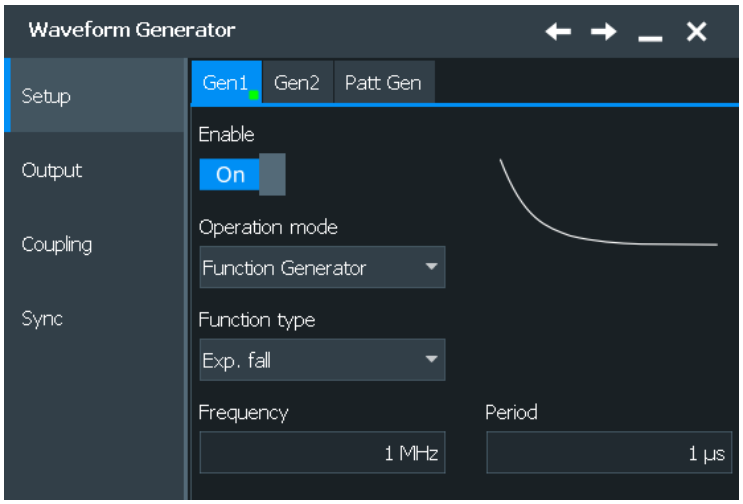

"Lorentz" Generates a Lorentz signal . You can set the Frequency and Period.



"Exp. rise" Generates an exponential rise signal. You can set the Frequency and Period.




"Exp. fall" Generates an exponential fall signal. You can set the Frequency and Period.




Remote command:
`WGENerator<m>:FUNCtion[:SELect]` on page 2335

**Frequency**
Sets the frequency of the waveform. The available frequency range depends on the selected "Function Type", see Frequency range of the function generator waveforms.

***Table 15-1: Frequency range of the function generator waveforms***

| "Function type" | Min frequency | Max frequency |
|---|---|---|
| "Sine" | 0.001 Hz | 100 MHz |
| "Square" | 0.001 Hz | 30 MHz |
| "Ramp" | 0.001 Hz | 1 MHz |
| "DC" | - | - |

| "Function type" | Min frequency | Max frequency |
|---|---|---|
| "Pulse" | 0.001 Hz | 30 MHz |
| "Cardinal sine" | 0.001 Hz | 5 MHz |
| "Cardiac" | 0.001 Hz | 1 MHz |
| "Gauss" | 0.001 Hz | 25 MHz |
| "Lorentz" | 0.001 Hz | 10 MHz |
| "Exp.rise" | 0.001 Hz | 1 MHz |
| "Exp. fall" | 0.001 Hz | 1 MHz |

The values of the "Frequency" and "Period" depend on each other, as:

*Period = 1/ Frequency*

Remote command:
`WGENerator<m>:FREQuency` on page 2335

**Period**
Sets the period of the waveform. The available period range depends on the selected "Function Type".

Remote command:
`WGENerator<m>:PERiod` on page 2336

**Duty cycle**
Sets the duty cycle for a square waveform. The duty cycle expresses for what percentage of the period, the signal state is high.

Remote command:
`WGENerator<m>:FUNCtion:SQUare:DCYCle` on page 2337

**Symmetry**
Sets the symmetry of a ramp waveform, the percentage of time the waveform is rising. By changing the symmetry of the ramp, you can create, for example, triangular waveforms.

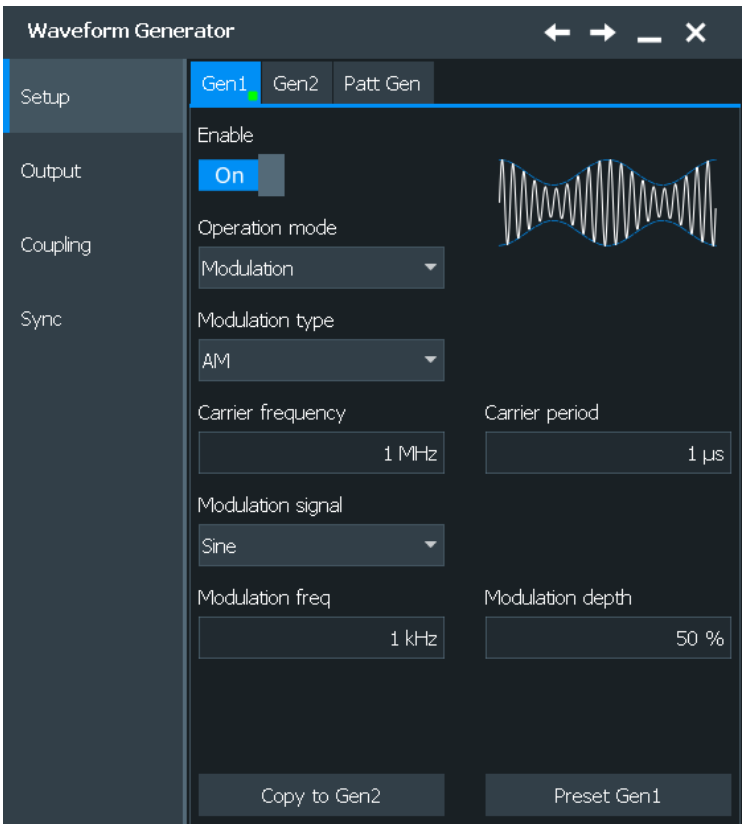
Remote command:
`WGENerator<m>:FUNCtion:RAMP[:SYMMetry]` on page 2336

**Pulse width**
Sets the pulse width, the pulse duration of the generated pulse waveform.

Remote command:
`WGENerator<m>:FUNCtion:PULSe[:WIDTh]` on page 2336

### 15.1.3 Modulation

Modulation is when properties of an original periodic waveform, the carrier signal, are varied according to a second modulating signal. The type of modulation used determines which properties are changed.




### 15.1.3.1 General settings

Consists of settings for selecting the modulation type.

**Modulation type**
Selects the modulation type, which defines how the carrier signal is modified.

"AM" Amplitude modulation.
See: Chapter 15.1.3.2, "AM modulation", on page 975.

"FM" Frequency modulation.
See: Chapter 15.1.3.4, "FM modulation", on page 977.

"PWM" Pulse width modulation
See: Chapter 15.1.3.3, "PWM modulation", on page 976

"FSK" Frequency shift keying (FSK) modulation.
See:Chapter 15.1.3.5, "FSK modulation", on page 979.

Remote command:
`WGENerator<m>:MODulation:TYPE` on page 2337

**Carrier frequency**
Sets the frequency of the carrier signal.

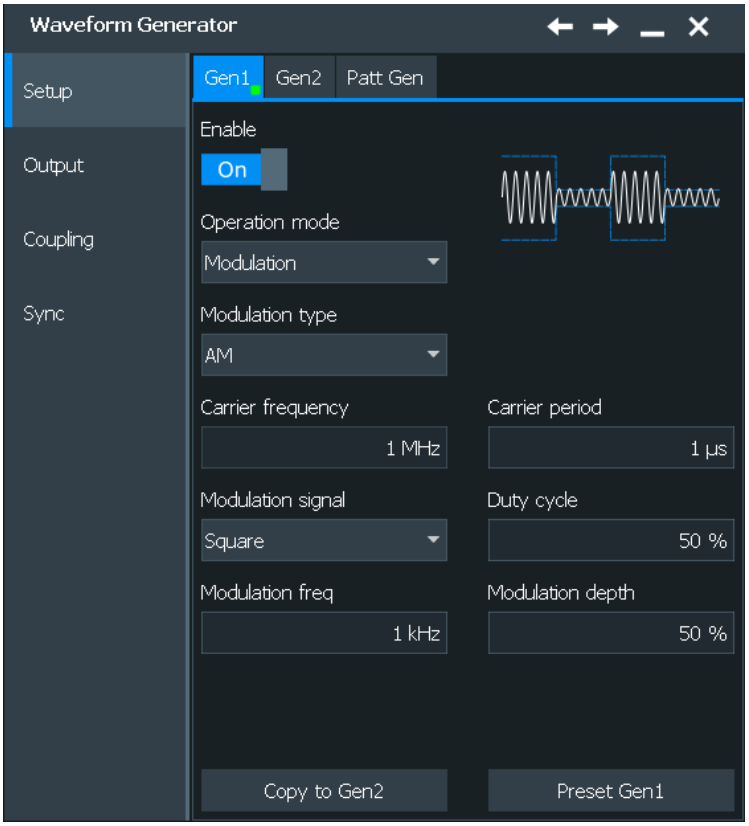
Remote command:
`WGENerator<m>:MODulation:CARRier:FREQuency` on page 2339

**Carrier period**
Sets the period of the carrier signal.

Remote command:
WGENerator<m>:MODulation:CARRier:PERiod on page 2339

### 15.1.3.2 AM modulation

For amplitude modulation (AM), the amplitude of the carrier signal is varied according to the modulation signal.




**Modulation signal**
Selects the type of the modulation signal for the AM modulation types.

Remote command:
WGENerator<m>:MODulation:AM[:FUNCtion] on page 2339

**Modulation freq**
Sets the frequency of the modulation waveform.

Remote command:
WGENerator<m>:MODulation:AM:FREQuency on page 2338

**Modulation depth**
Sets the modulation depth, the percentage of the amplitude range that is used for the modulation.

Remote command:
WGENerator<m>:MODulation:AM:DEPTh on page 2338

**Symmetry**
Sets the symmetry for the ramp modulation waveform, the percentage of time that the waveform is rising.

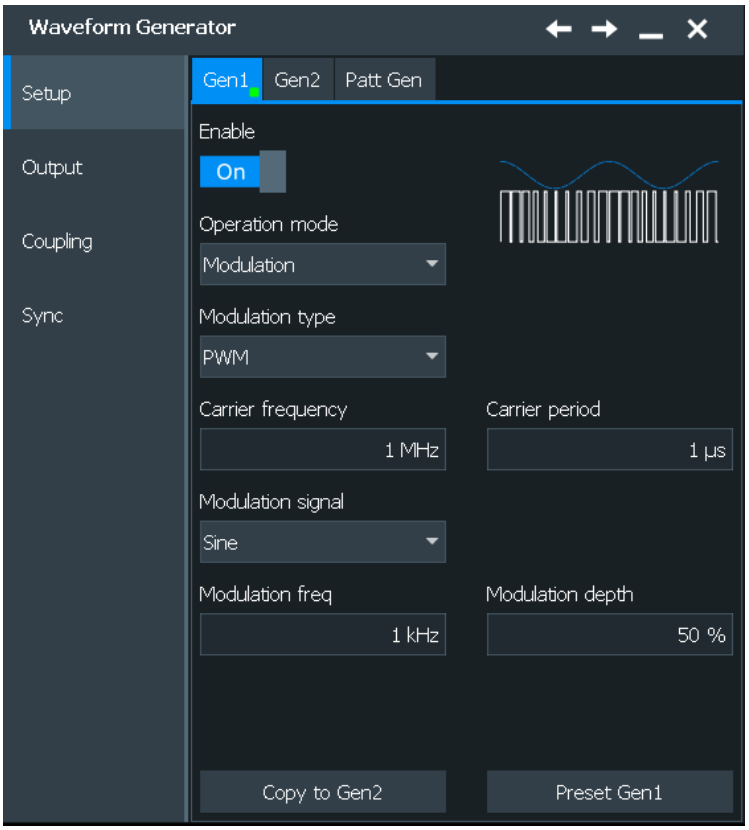
Remote command:
WGENerator<m>:MODulation:AM:SYMMetry on page 2339

**Duty cycle**
Sets the duty cycle for a square waveform. The duty cycle expresses for what percentage fraction of the period, the waveform is active, i.e. the signal state is high.

Remote command:
WGENerator<m>:MODulation:AM:DCYCle on page 2338

### 15.1.3.3 PWM modulation

For pulse width modulation (PWM), the time for which the signal is in a high state is varied according to the modulation signal.




**Modulation signal**
Selects the type of the modulation signal for the PWM modulation types.

Remote command:
WGENerator<m>:MODulation:PWM[:FUNCtion] on page 2343

**Modulation freq**
Sets the frequency of the modulation waveform.

Remote command:
WGENerator<m>:MODulation:PWM:FREQuency on page 2343

**Modulation depth**
Sets the modulation depth, the percentage of the pulse width range that is used for the modulation.

Remote command:
WGENerator<m>:MODulation:PWM:DEPTh on page 2342

**Symmetry**
Sets the symmetry for the ramp modulation waveform, the percentage of time that the waveform is rising.

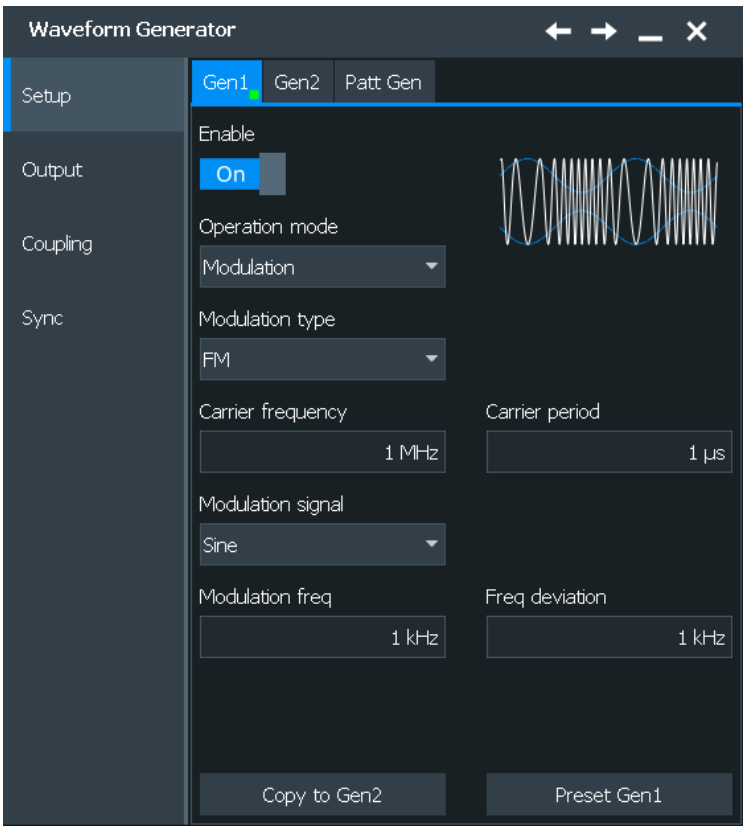
Remote command:
WGENerator<m>:MODulation:PWM:SYMMetry on page 2343

**Duty cycle**
Sets the duty cycle for a square waveform. The duty cycle expresses for what percentage fraction of the period, the waveform is active, i.e. the signal state is high.

Remote command:
WGENerator<m>:MODulation:PWM:DCYCle on page 2342

#### 15.1.3.4 FM modulation

For frequency modulation (FM), the frequency of the carrier signal is varied according to the modulation signal.




**Modulation signal**
Selects the type of the modulation signal for the FM modulation types.

Remote command:
`WGENerator<m>:MODulation:FM[:FUNCtion]` on page 2341

**Modulation freq**
Sets the frequency of the modulation waveform.

Remote command:
`WGENerator<m>:MODulation:FM:FREQuency` on page 2340

**Symmetry**
Sets the symmetry for the ramp modulation waveform, the percentage of time that the waveform is rising.

Remote command:
`WGENerator<m>:MODulation:FM:SYMMetry` on page 2341

**Freq deviation**
Sets the frequency deviation, the maximum difference between and FM modulated signal and the carrier signal.

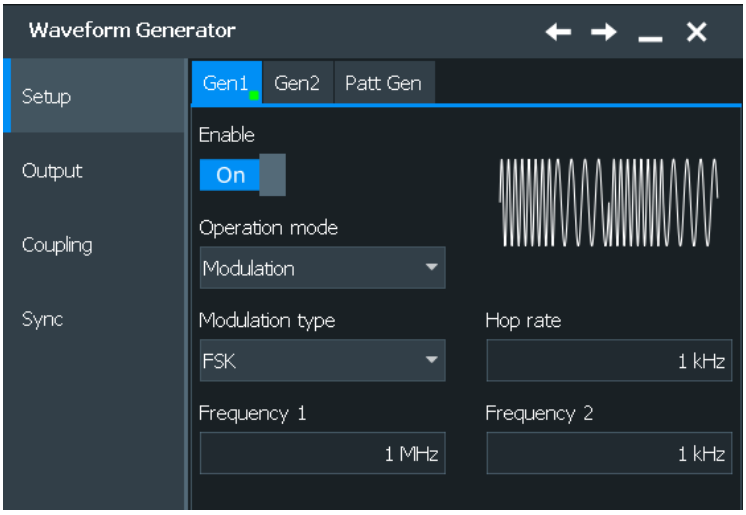
Remote command:
`WGENerator<m>:MODulation:FM:DEViation` on page 2340

**Duty cycle**
Sets the duty cycle for a square waveform. The duty cycle expresses the percentage of the period during which the waveform is active, i.e. the signal state is high.

Remote command:
`WGENerator<m>:MODulation:FM:DCYCle` on page 2340

#### 15.1.3.5 FSK modulation

For frequency shift keying (FSK) modulation, the signal switches between Frequency 1 and Frequency 2 at a Hop rate.




**Frequency 1**
Sets the frequency of the first signal in FSK modulated signal.

Remote command:
`WGENerator<m>:MODulation:FSK:FONE` on page 2341

**Frequency 2**
Sets the frequency of the second signal in FSK modulated signal.

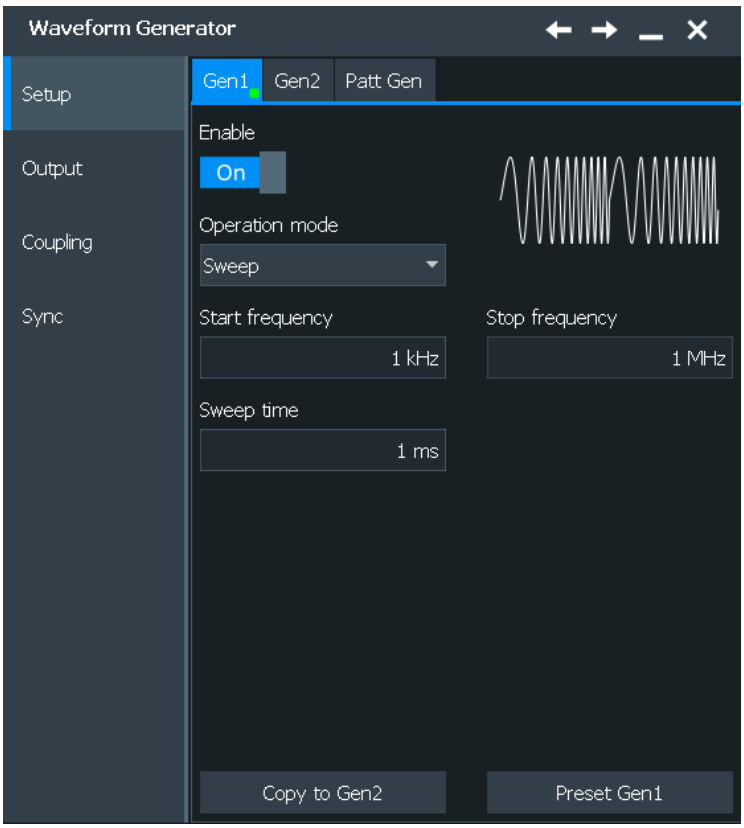
Remote command:
`WGENerator<m>:MODulation:FSK:FTWO` on page 2341

**Hop rate**
Sets the hop rate, the frequency at which signal switches between Frequency 1 and Frequency 2.

Remote command:
`WGENerator<m>:MODulation:FSK[:RATE]` on page 2342

### 15.1.4 Sweep

In the sweep mode, the R&S RTP generates a signal whose frequency gradually changes from the Start frequency to the Stop freuqency for a certain Sweep time.




**Start frequency**
Sets the start frequency of the sweep signal.

Remote command:
`WGENerator<m>:SWEep:FSTart` on page 2344

**Stop freuqency**
Sets the stop frequency of the sweep signal.

Remote command:
`WGENerator<m>:SWEep[:FEND]` on page 2344

**Sweep time**
Sets the duration of the sweep.

Remote command:
`WGENerator<m>:SWEep:TIME` on page 2344

### 15.1.5 Arbitrary

The arbitrary waveform generator allows you to output a user-defined waveform for testing your devices. You can output a waveform from a file or from the current R&S RTP reference curve format. Files in `*.csv` and `*.bin` formats are supported. This files must follow a defined structure. You can load `*.csv` files in an R&S Wave

Gen format (see "Content and format of the Rohde & Schwarz arbitrary waveform CSV files" on page 981), Tektronix AFG format or Keysight WaveGen format.

**Content and format of the Rohde & Schwarz arbitrary waveform CSV files**

The waveform generator format can contain the following values:

- Rate
- Time value
- Voltage value

If all the values are defined, the file format is as follows:

```
Rate = 5000000           //Sample rate of the arbitrary waveform.
0.000000E+000,-5.995    //Time value 1, Voltage value 1
1.237011E-005,-6.0      //Time value 2, Voltage value 2
.......
```

In this case, the rate is reflected in the "Sample Rate" field of the user interface. The total number of Time/Voltage values is reflected in the "Samples" of the user interface. Anything written after `//` is ignored as a comment.

You can define only some of the values. According to what you define, the file format looks different and is handled differently:

- With specified *Rate*:
  Time values are ignored. You can specify just rate and voltage values as below:

```
Rate = 5000000              //Sample rate of the arbitrary waveform
-5.995                      //Voltage value 1
-6.0                        //Voltage value 2
.......
```

- Without specified *Rate* and without specified *Time* values:
  The last user-defined sample rate is used to calculate the waveform. You can change the "Sample rate" in the user interface. The voltage values are then played with this sample rate.

```
-5.995               //Voltage value 1
-6.0                  //Voltage value 2
.......
```

- Without specified *Rate* and with specified *Time*:
  The timing information of the first 2 time values is used to calculate the sample rate.

**Example:**

Consider the following file:

```
0.000000E+000,-5.995    //Time value 1, Voltage value 1
1.237011E-005,-6.0      //Time value 2, Voltage value 2
```

The sample rate is:

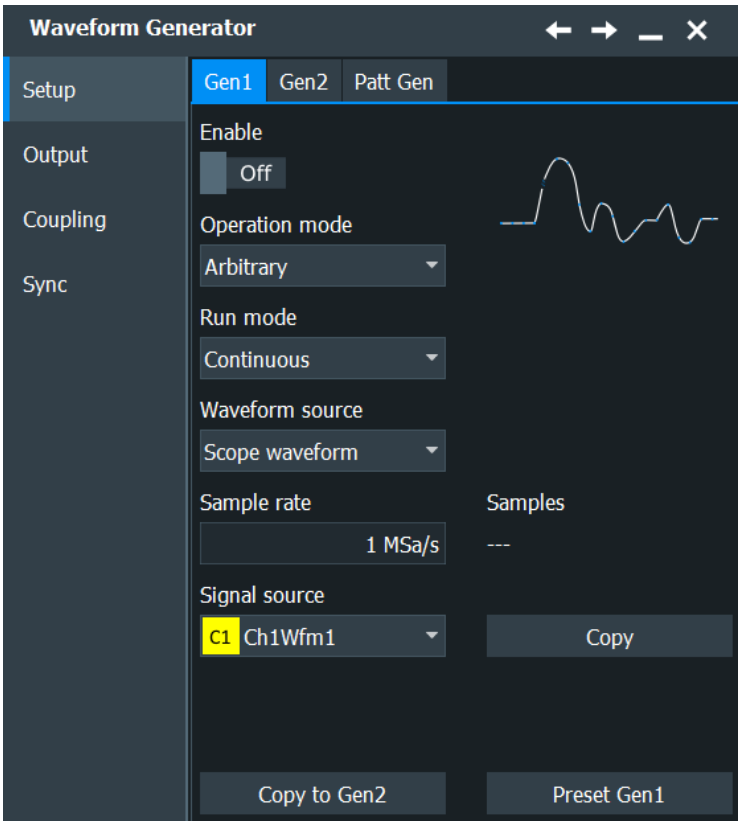
*Sample rate = 1/ Time between first two samples* =1 / 1.237011E-005 = 80.840KSampe/sec

**Content and format of the Rohde & Schwarz arbitrary generator BIN files**

The file stream should contain the following information in the given order:

- Sample rate [double format]
- Number of samples [unsigned integer format]
- Samples [double format] * number of samples

### 15.1.5.1 General settings




**Waveform source**
Selects the arbitrary waveform source. You can load an existing file or load the current oscilloscope waveform.

Remote command:
`WGENerator<m>:ARBGen[:SOURce]` on page 2348

**Run mode**
Selects the duration for which the signal of the arbitrary generator will be output after the trigger event. You can choose between a "Continuous" and "Single period" duration.

Remote command:
`WGENerator<m>:ARBGen:RUNMode` on page 2347

**Sample rate**
Sets the sample rate for the arbitrary waveform.

Remote command:
WGENerator<m>:ARBGen:SRATe on page 2348

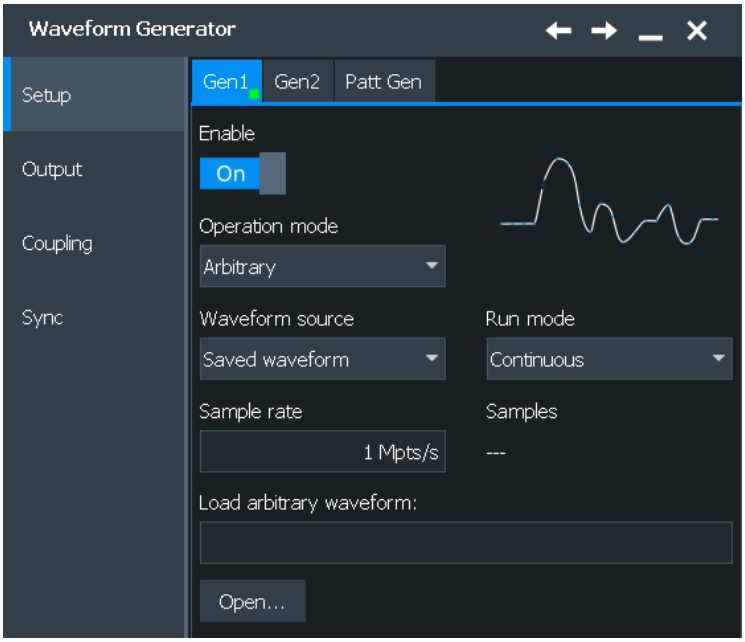
**Samples**
Displays the number of samples in the loaded waveform.

Remote command:
WGENerator<m>:ARBGen:SAMPles? on page 2347

**Run single**
Generates a single period of the arbitrary waveform if "Run mode" is set to "Single period".

Remote command:
WGENerator<m>:ARBGen:RUNSingle on page 2348

### 15.1.5.2 Saved waveform



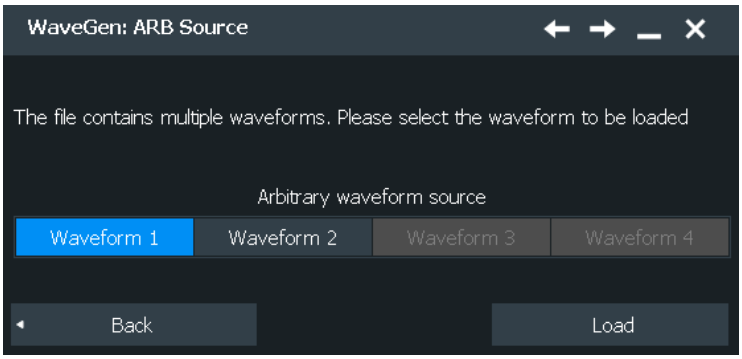

For a saved waveform, the following settings are available:

**Load arbitrary waveform**
Opens a file selection dialog box and loads the selected file. Supported are `.bin` and `.csv` extension files.

Remote command:
WGENerator<m>:ARBGen:NAME on page 2346
WGENerator<m>:ARBGen:OPEN on page 2346

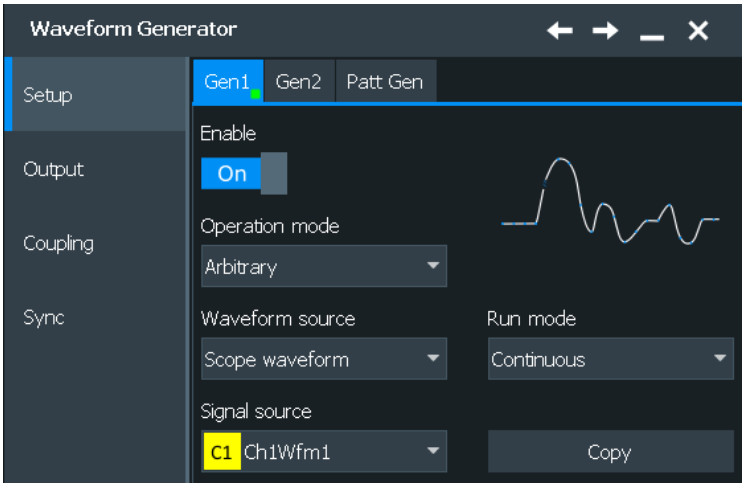
**Select waveform**
When a multichannel file is loaded into the arbitrary waveform generator, a dialog appears to select which waveform from the file is loaded.

Select the waveform and press "Load" to load it into the arbitrary waveform generator.




Remote command:
`WGENerator<m>:ARBGen:MULTichannel:NAME` on page 2345
`WGENerator<m>:ARBGen:MULTichannel:IMPort` on page 2345
`WGENerator<m>:ARBGen:MULTichannel:OPEN` on page 2346

### 15.1.5.3 Scope waveform




For a scope waveform, the following settings are available:

**Signal source**
Selects the oscilloscope source, from which the arbitrary signal is loaded.

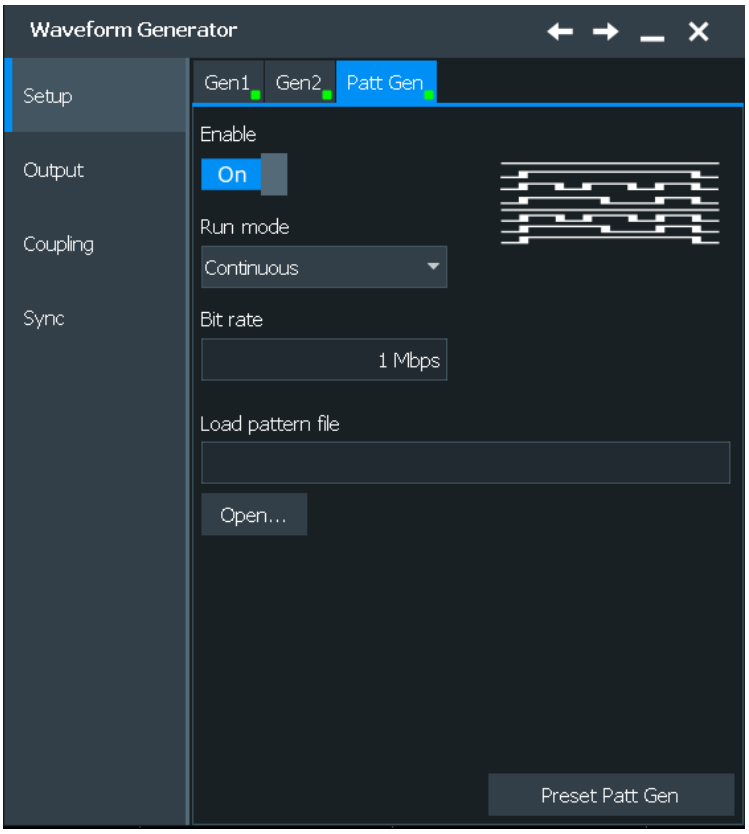
Remote command:
`WGENerator<m>:ARBGen:SELect` on page 2347

**Load**
Loads the waveform from the selected "Signal source".

Remote command:
`WGENerator<m>:ARBGen:COPY` on page 2345

## 15.2 Setup of the pattern generator

The pattern generator outputs parallel patterns.




**NOTICE**

**Using pattern generator accessories**

The pattern generator with connected Patt Gen Cable and Patt Gen Board (1329.7054.02) is considered as a test probe, EN 61326-2-1, clause 5.2.4.101, note 1. Therefore normal operation may display increased emissions above the limits as specified in EN 55011 and/or reduced interference resistance as required in EN 61326-1, table 1, basic requirements.

If the cable and the board are connected, other surrounding electronic devices may be disturbed. Furthermore, signals at the analog generator outputs Gen1 and Gen2 may be distorted by surrounding devices.

The settings of the pattern generator are not affected by an instrument preset. Press "Preset Patt Gen" to preset the settings of the pattern generator.

### 15.2.1 Content and format of the pattern generator files

The pattern generator supports `.bin` or `.csv` file formats.

#### Content and format of the R&S*.csv files

The file's header have the following structure:

- Bit rate [double]: the number of transmitted bits per second. The value is reflected in the user interface.
- High level [double]: the value is reflected in the user interface.
- Mask [string of 8 characters made up of "X"/"1"/"0"] : defines how the output of the pattern generator looks like. The following values are defined:
  - *1'* means that the pin output is always at high level
  - *0'* means that the pin output is always at low level (close to 0V)
  - *X* means that the pin output varies according to the given pattern
- Data sample format [HEX, BIN, OCT, DEC]: indicates how the samples are going to be interpreted. Each sample is represented as 8bit value (corresponding to the 8bit pattern generator) considering the selected format.

#### Example: Sample format HEX

```
Format= HEX        // Defines the format of the pattern values [HEX, DEC, BIN, OCT]
0F -> Data Sample 1
21 -> Data Sample 2
........
```

The samples are mapped on the 8 pins of the pattern generator as follows:

```
=> Pattern Samples are:
D7   D6   D5   D4   D3   D2   D1   D0
0    0    0    0    1    1    1    1
0    0    1    0    0    0    0    1
```

#### Example: .csv pattern generator file

```
R&S Pattern Generator File
Rate= 1000000       // Bit Rate [double]
HLevel= 1.5         // High Voltage Level [double]
Mask= X111000X      // Masks the Pins to be used in the Pattern Generator
                    // [0 => always LOW, 1 => always HIGH, X/x => used in the Pattern]
Format= DEC         // Defines the format of the pattern values [HEX, DEC, BIN, OCT]0
1
2
3
....
200
```

#### Content and format of the R&S pattern generator *.bin files

For the content of the fields, refer to "Content and format of the R&S*.csv files" on page 986.

The file stream should contain the following information in the given order:

- Bit rate [double]
- High level [double]
- Mask [string of 8 characters made up of "X"/"1"/"0"]
- Number of samples [UINT32]
- Data samples [UINT8] * number of samples

### 15.2.2 Pattern generator settings

**Enable**
Enables the waveform generator/ pattern generator and outputs the signal to the connectors.

Remote command:
`WGENerator<m>[:ENABle]` on page 2334
`PGENerator:ENABle` on page 2349

**Run mode**
Selects the duration for which the signal of the generator will be output after the trigger event. You can choose between a "Continuous" and "Single period" duration.

Remote command:
`PGENerator:RUNMode` on page 2350

**Bit rate**
Sets the number of transmitted bits per second for the pattern generator.

Remote command:
`PGENerator:BITRate` on page 2349

**Run single**
Generates a single period of the pattern waveform if "Run mode" is set to "Single period".

Remote command:
`PGENerator:RUNSingle` on page 2350

**Load pattern file**
Opens a dialog for selecting an existing pattern file. It is possible to load `.bin` or `.csv` files, see Chapter 15.2.1, "Content and format of the pattern generator files", on page 986.

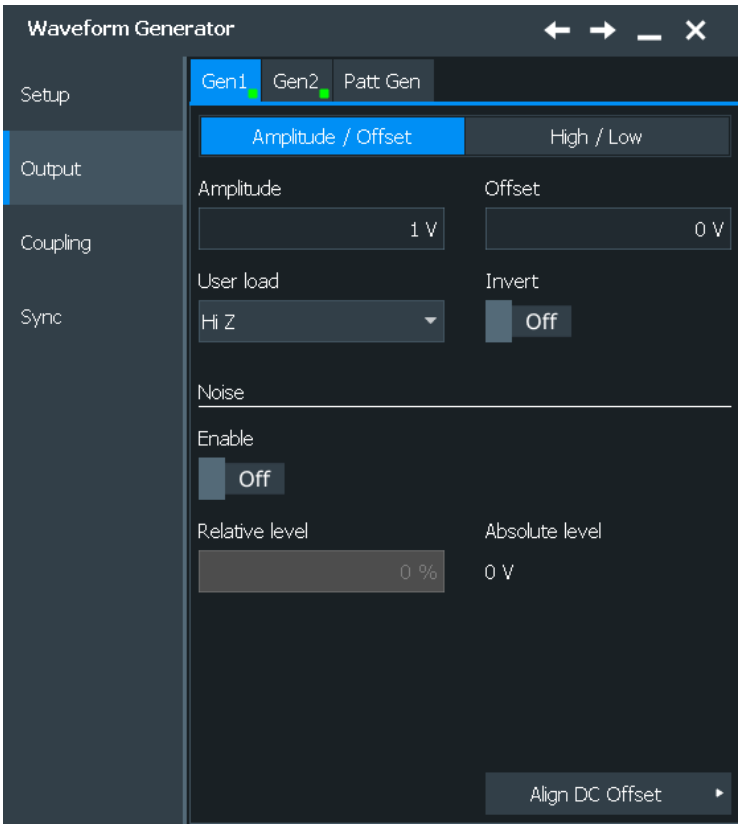
Remote command:
`PGENerator:FILE:OPEN` on page 2349
`PGENerator:FILE[:NAME]` on page 2349

**Preset Gen1/Gen2/Patt Gen**
Sets the parameters of the generator to their default values. The settings of the generators are not affected by an instrument preset.They are also not stored in the user-defined preset.

Remote command:
`WGENerator<m>:PRESet` on page 2335
`PGENerator:PRESet` on page 2350

## 15.3 Output

Access: "Menu" > "Apps" > "Generators" > "Gen1" > "Setup" tab.




In this tab you can define the generator output and add noise to the waveform.

### 15.3.1 Output

**Amplitude/Offset - High/Low**
Selects how the output is defined: with a defined "Amplitude"/ "Offset" or with "High"/"Low" values.

**Amplitude**
Sets the amplitude, peak to peak voltage, of the output waveform. It is defined as the voltage difference between the maximum ("High") and the minimum ("Low") voltage levels.

The "Amplitude" value is set for the currently selected "User Load". If the "User Load" is changed, the value of "Amplitude" is adapted to this new setting.

Remote command:
`WGENerator<m>:VOLTage[:VPP]` on page 2351

**Offset**
Sets a voltage offset.

Remote command:
`WGENerator<m>:VOLTage:OFFSet` on page 2352

**High**
Sets the high signal level of the output waveform.

Remote command:
`WGENerator<m>:VOLTage:HIGH` on page 2351

**Low**
Sets the low signal level of the output waveform.

Remote command:
`WGENerator<m>:VOLTage:LOW` on page 2352

**Invert**
Inverts the waveform at the offset level.

Remote command:
`WGENerator<m>:VOLTage:INVersion` on page 2351

**User Load**
Select the user load, the load of the DUT at its connection. You can select either a "50Ω" or a "HiZ" (high input impedance) load.

Remote command:
`WGENerator<m>:OUTPut[:LOAD]` on page 2350

**DC Level**
Sets the voltage DC level for the generated DC signal, for "Operation mode" >"Function Gen." and "Function type"> "DC".

Remote command:
`WGENerator<m>:VOLTage:DCLevel` on page 2351

### 15.3.2 Noise

**Enable**
Enables the adding of noise to the waveform.

Remote command:
WGENerator<m>:MODulation:NOISe on page 2353

**Relative level**
Sets the level of the noise in percentage of the set "Amplitude" output of the signal.

Remote command:
WGENerator<m>:MODulation:NLPCent on page 2353

**Absolute level**
Displays the level of the noise in volts.

Remote command:
WGENerator<m>:MODulation:NLABsolute? on page 2353

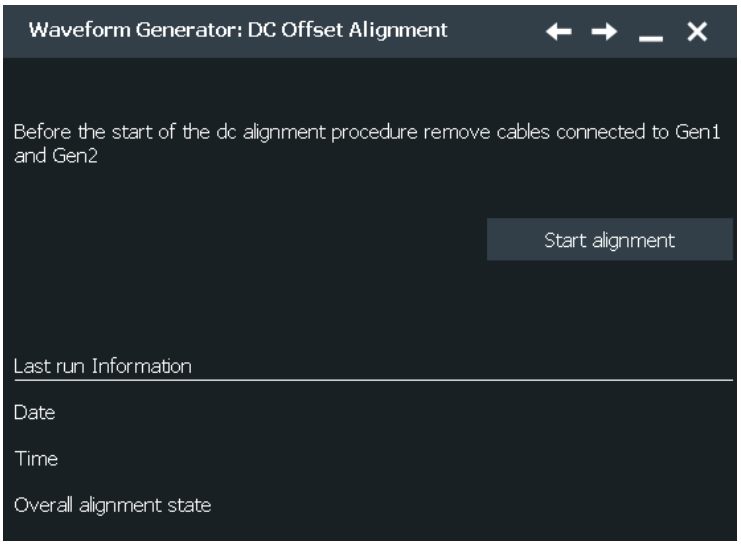
**Absolute level**
For "Function type">"DC" only.

Sets the level for the DC signal.

Remote command:
WGENerator<m>:MODulation:NDCLevel on page 2352

### 15.3.3 DC offset alignment

Access: [Gen] > "Output" > "Align DC Offset"




**Start Alignement**
Starts the alignment of the DC offset.

Remote command:
GENerator:ALIGnment:DC[:STARt] on page 2354

**Date**
Displays the date of the last performed DC offset alignment.

Remote command:
GENerator:ALIGnment:DC:RESult:DATE? on page 2354

**Time**
Displays the time of the last performed DC offset alignment.

Remote command:
GENerator:ALIGnment:DC:RESult:TIME? on page 2354

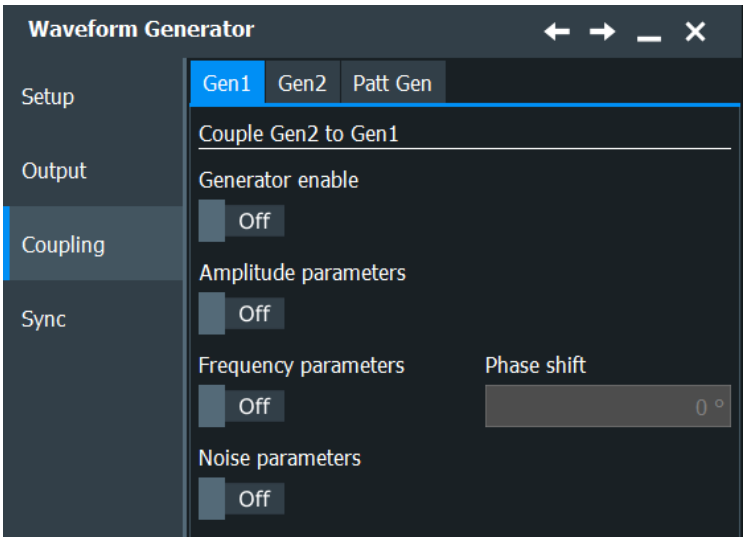
**Overall alignment state**
Displays the result of the DC offset alignment.

Remote command:
GENerator:ALIGnment:DC:RESult[:STATe]? on page 2354

## 15.4 Coupling settings

Access: "Wave Gen" menu > "Coupling & Sync" tab.

In the R&S RTP, you can couple certain settings of the available waveform generators. If one of the available coupling options is enabled for one of the generators, then you cannot change the coupled parameters at the "Setup" tab of the other generator. The values for all coupled parameters are automatically taken from the currently selected generator.




**Generator enable**
Enables the coupling of the generators, with the selected set of parameters: amplitude, frequency and noise.

Remote command:
WGENerator<m>:COUPling:STATe on page 2355

**Amplitude parameters**
Couples all amplitude parameters of "Gen1" to "Gen2" or "Gen2" to "Gen1" including the load. The values are taken from the currently selected generator.

- For all "Operation Mode": the output settings except of "Load" and "Inversion", "Amplitude", "High", "Offset" and "Low".
- For "Operation Mode" > "Function Gen.": "DC Level"
- For "Operation Mode" > "Modulation": "Modulation depth"

Remote command:
`WGENerator<m>:COUPling[:FREQuency]` on page 2356

**Frequency parameters**
Couples all frequency parameters of "Gen1" to "Gen2" or "Gen2" to "Gen1". The values are taken from the currently selected generator.

- For "Operation Mode" > "Function Gen.": "Frequency" and "Period"
- For "Operation Mode" > "Modulation": "Carrier frequency", "Carrier period", "Modulation freq", "Freq deviation", "Frequency 1", "Frequency 2" and "Hop rate"
- For "Operation Mode" > "Sweep": "Start frequency", "Stop frequency" and "Sweep time".
- For "Operation Mode" > "Arbitrary": "Arb wfm source" and "Signal source".

You can still change the other settings of the generators independently.

Remote command:
`WGENerator<m>:COUPling:AMPLitude` on page 2355

**Phase shift**
Sets the phase shift between the waveform of "Gen1" and "Gen2" when the frequency parameters of the two waveforms are coupled.

Remote command:
`WGENerator<m>:COUPling:PHASeshift` on page 2355

**Noise parameters**
Couples all noise parameters of "Gen1" to "Gen2" or "Gen2" to "Gen1". The values are taken from the currently selected generator.

Remote command:
`WGENerator<m>:COUPling:NOISe` on page 2356

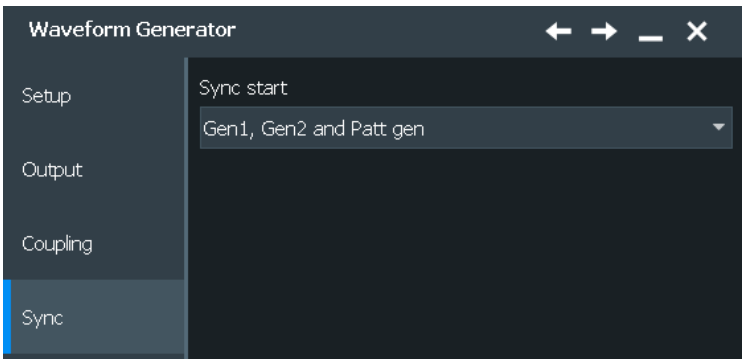
**Arbitrary waveform**
Available for General settings = "Arbitrary".

Couples the arbitrary waveform "Gen1" to "Gen2" or "Gen2" to "Gen1". The values are taken from the currently selected generator.

Remote command:
`WGENerator<m>:COUPling:NOISe` on page 2356

## 15.5 Sync settings

Access: [Gen] > "Sync"




**Sync start**
Selects, which signals generated from the waveform generator are synchronized.

Selecting one of the sync options indicates that the first samples of those signals are generated at the same time, irrespective of if the generators are on or off. Selecting one of the coupling options automatically syncs the signals generated by the two waveform generators.

Remote command:
`GENerator:SYNC[:COMBination]` on page 2356

## 15.6 Configuring the waveform generator

This chapter explains step-by-step how to configure the waveform generator.

- Configuring a function waveform....993
- Configuring a modulation waveform....994
- Configuring a sweep waveform....995
- Configuring an arbitrary waveform....995
- Configuring a pattern generator waveform....996

### 15.6.1 Configuring a function waveform

1. Open "Menu" > "Apps" > "Generators" > "Gen1".
2. Select the "Setup" tab.
3. Under "Operation mode", enable the "Function Gen." button.
4. Select the "Function type" that you want to generate, e.g. "Sine".
5. Depending on the selected "Function type", configure the settings of the waveform like "Frequency" and "Period".
6. If necessary, change the "Output" settings or add "Noise" to the waveform.

7. Press the "Enable" button, to output the waveform at the output connector of the waveform generator.

### 15.6.2 Configuring a modulation waveform

**Generating an AM modulated waveform**

1. Open "Menu" > "Apps" > "Generators" > "Gen1".
2. Select the "Setup" tab.
3. Under "Operation mode", enable the "Modulation" button.
4. Tap "Modulation type" and select "AM".
5. Set the "Carrier frequency" and the "Carrier period".
6. Tap "Modulation signal" and select the required waveform.
7. Depending on the selected "Modulation signal", configure the settings of the waveform like "Modulation freq" and "Modulation depth".
8. If necessary, change the "Output" settings or add "Noise" to the waveform.
9. Press the "Enable" button, to output the waveform at the output connector of the waveform generator.

**Generating an PWM modulated waveform**

1. Open "Menu" > "Apps" > "Generators" > "Gen1".
2. Select the "Setup" tab.
3. Under "Operation mode", enable the "Modulation" button.
4. Tap "Modulation type" and select "PWM".
5. Set the "Carrier frequency" and the "Carrier period".
6. Tap "Modulation signal" and select the required waveform.
7. Depending on the selected "Modulation signal", configure the settings of the waveform like "Modulation freq" and "Modulation depth".
8. If necessary, change the "Output" settings or add "Noise" to the waveform.
9. Press the "Enable" button, to output the waveform at the output connector of the waveform generator.

**Generating an FM modulated waveform**

1. Open "Menu" > "Apps" > "Generators" > "Gen1".
2. Select the "Setup" tab.
3. Under "Operation mode", enable the "Modulation" button.

4. Tap "Modulation type" and select "FM".
5. Set the "Carrier frequency" and the "Carrier period".
6. Tap "Modulation signal" and select the required waveform.
7. Depending on the selected "Modulation signal", configure the settings of the waveform like "Modulation freq" and "Freq deviation".
8. If necessary, change the "Output" settings or add "Noise" to the waveform.
9. Press the "Enable" button, to output the waveform at the output connector of the waveform generator.

**Generating an FSK modulated waveform**

1. Open "Menu" > "Apps" > "Generators" > "Gen1".
2. Select the "Setup" tab.
3. Under "Operation mode", enable the "Modulation" button.
4. Tap "Modulation type" and select "FSK".
5. Set the "Frequency 1", "Frequency 2" and the "Hop rate".
6. If necessary, change the "Output" settings or add "Noise" to the waveform.
7. Press the "Enable" button, to output the waveform at the output connector of the waveform generator.

### 15.6.3 Configuring a sweep waveform

1. Open "Menu" > "Apps" > "Generators" > "Gen1".
2. Select the "Setup" tab.
3. Under "Operation mode", enable the "Sweep" button.
4. Set the "Start frequency", the "Stop frequency" and the "Sweep time".
5. If necessary, change the "Output" settings or add "Noise" to the waveform.
6. Press the "Enable" button, to output the waveform at the output connector of the waveform generator.

### 15.6.4 Configuring an arbitrary waveform

**Generating an arbitrary waveform from a saved file**

1. Open "Menu" > "Apps" > "Generators" > "Gen1".
2. Select the "Setup" tab.

3. Under "Operation mode", enable the "Arbitrary" button.
4. Set the "Arb wfm source" to "Saved waveform".
5. Select the "Run mode".
6. Press "Open" and set the path to your saved arbitrary waveform.
7. Set the "Sample rate".
8. If necessary, change the "Output" settings or add "Noise" to the waveform.
9. Press the "Enable" button, to output the waveform at the output connector of the waveform generator.

**Generating an arbitrary waveform from the scope waveform**

1. Open "Menu" > "Apps" > "Generators" > "Gen1".
2. Select the "Setup" tab.
3. Under "Operation mode", enable the "Arbitrary" button.
4. Set the "Arb wfm source" to "Scope waveform".
5. Select the "Run mode".
6. Press the "Signal source" button and select the channel source for the waveform.
7. Set the "Sample rate".
8. If necessary, change the "Output" settings or add "Noise" to the waveform.
9. Press the "Enable" button, to output the waveform at the output connector of the waveform generator.

### 15.6.5 Configuring a pattern generator waveform

1. Open "Menu" > "Apps" > "Generators" > "Gen1".
2. Select the "Setup" tab.
3. Select the "Patt Gen" tab.
4. Select the "Run mode".
5. Press "Open" and set the path to your saved pattern file.
6. Set the "Bit rate" and "High level".
7. Press "Open" and set the path to your saved arbitrary waveform.
8. Press the "Enable" button, to output the pattern at the output connector of the pattern generator.

# 16 Pulse source and TDR/TDT analysis (options R&S RTP-B7/K130)

## 16.1 Pulse source (option R&S RTP-B7)

The pulse source option R&S RTP-B7 provides a symmetrical differential pulse signal with steep rise time and configurable parameters. The hardware module of the option features four 2.92 mm connectors (K type), two for signal output (Out, $\overline{\text{Out}}$), and two for reference signal output (Ref, $\overline{\text{Ref}}$).

Typical applications of the pulse source are:

- Stimulus for devices under test, for example, as a precise clock
- Accurate source for deskewing cables and probes for differential measurements
- Analysis of time-transmission characteristics of a differential signal path
- TDR/TDT analysis with option R&S RTP-K130

The module is installed at the front panel of the instrument. To ensure correct installation and calibration, send the instrument to a Rohde & Schwarz service center.

**Preventing electrostatic discharge (ESD)**

Electrostatic discharge is most likely to occur when you connect or disconnect a DUT.

► **NOTICE!** Electrostatic discharge can damage the electronic components of the product and the device under test (DUT).

Ground yourself to prevent electrostatic discharge damage:

a) Use a wrist strap and cord to connect yourself to ground.
b) Use a conductive floor mat and heel strap combination.

**Usage of cables**

Beside the pulse source module, the option includes several accessories: SMA cables, 50 Ω terminations, adapters, and more. The accessory case provides also space for optional accessories. All included accessories are listed in the data sheet.

**NOTICE**

When fastening the cable, observe the maximum torque of 0.9 Nm. Therefore, using a 0.9 Nm torque wrench is recommended (optional accessory 1328.8534.35).

To ensure accurate and reliable measurements, connect only cables with a maximum length of 3 m.

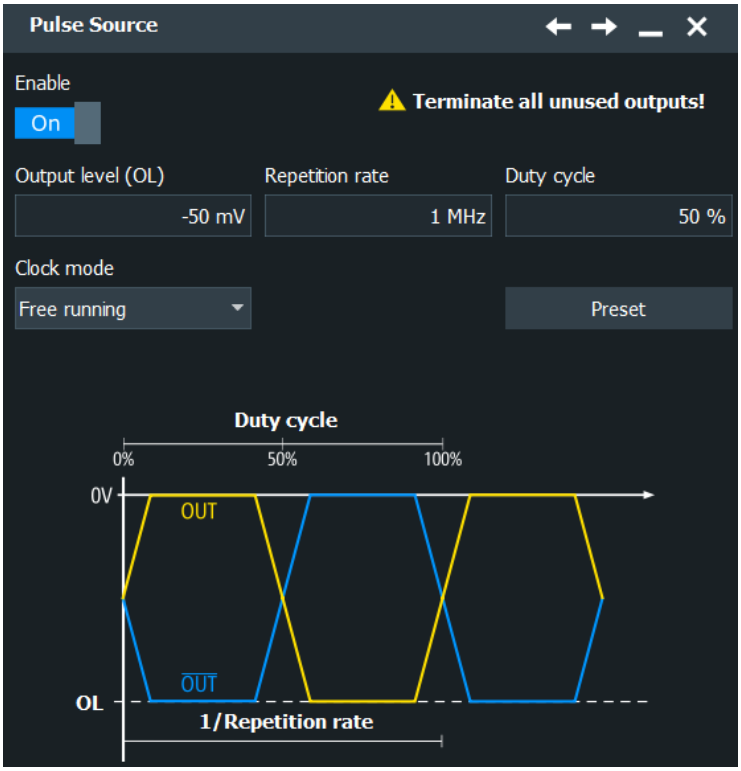
Always use double-shielded cables to prevent electromagentic interference (EMI) from impairing measurement results.

Always use cables to connect the DUT or test fixture to the pulse source outputs. Never connect the DUT or test fixture directly to the outputs.

### 16.1.1 Settings for the pulse source

Access: [Apps] > "Generators" > "PSC"

Terminate all unused outputs of the pulse source with 50 Ω terminating resistors.




**Enable**
Switches all four pulse source outputs on and off.

Remote command:
`PSRC[:STATe]` on page 2357

**Output level**
Sets the low level of the output pulse. The high level is 0 V and cannot be changed.

Remote command:
`PSRC:OUTPutlow` on page 2358

**Repetition rate**
Sets the pulse frequency, the repetition rate of the pulse. Available values depend on the selected clock mode.

Remote command:
`PSRC:REPRate` on page 2358

**Duty cycle**
Sets the duty cycle of the pulse in %, which is the ratio of the positive pulse width to the period of the signal.

For repetition rates > 5 MHz, the value is fixed at 50 %.

Remote command:
`PSRC:DUTYcycle` on page 2358

**Clock mode**
Sets the dependency of the pulse clock on the instrument's reference clock. The appropriate clock mode depends on the measurement application.

"Free running" The clock of the pulse source is independent. Deviations of the system do not affect the pulse clock, and deterministic conditions are avoided.
If the waveform aquisition is set to "Average", the "free running" clock mode can improve the measurement result. Internal spurious signals of the instrument are reduced by averaging, because they are not coherent to the pulse source signal.
See also: "Arithmetic" on page 133.

"Locked" The pulse source is locked to the reference clock of the instrument.

Remote command:
`PSRC:CLOCkmode` on page 2358

**Preset**
Resets the pulse source to the default state. Pulse source settings are not are not affected by an instrument preset. They are also not stored in the user-defined preset.

Remote command:
`PSRC:RST` on page 2359

## 16.2 TDR/TDT analysis (option R&S RTP-K130)

The option R&S RTP-K130 provides analysis of time domain reflectometry (TDR) and time domain transmission (TDT) to verify high-speed digital designs. Using the results of the TDR/TDT calibration, the instrument removes the effects of the cables (deembedding), and measures the DUT.

Typical use cases are:

- Locate faults in a circuit
- Measure impedance
- Measure step response of a DUT

You can use the TDR/TDT option for single-ended and differential measurements.

### 16.2.1 About TDR/TDT measurements

Using TDR/TDT measurements, you can analyze the reflections and transmission characteristics of a DUT. With TDR, it is possible to determine the type, magnitude and location of a fault. TDT delivers information on stability, losses and bandwidth of the DUT.

The signal source is the pulse source, and the oscilloscope measures the reflected and transmitted signals.

#### Requirements

TDR/TDT analysis requires the pulse source option R&S RTP-B7.

Cabling requirements:

- Use only high-quality double-shielded RF cables. Two suitable cables are delivered with the pulse source option R&S RTP-B7.
- Use low loss cables with frequency range that fits the oscilloscope bandwidth.
- Consider the recommended cable length: 0.5 m < length < 3 m.
- For differential TDR/TDT, use a matched pair cable, e.g. R&S RT-ZA17.

#### Calibration standards

During calibration, you measure the connecting cables of the measurement setup against standards. Calibration standards assumed to be ideal.

- Short with RF quality: a standard short is delivered with the pulse source option R&S RTP-B7. You can use other shorts with electrical length 0 ns … 2 ns. Default value is the R&S RTP-B7 short.
  For differential TDR, the short does not require ground connection, the calibration is fully differential.
- Open (DC)
- Match (DC): a standard 50 Ω match is delivered with the pulse source option R&S RTP-B7.
  You can use another match with 50 Ω±10% for single-ended measurements, or 100 Ω±10% for differential measurements.
- Through (RF): the length is assumed to be zero.

#### Calibration results

After successful calibration, the frequency response of the short and through measurements are displayed. Steep dips in the graph indicate high attenuation within your desired bandwidth, which impairs the performance. To improve the measurement,

reduce the bandwidth of the system. To check the impact of the frequency response calibration on the time domain waveform, you can disable the usage ("Use frequency response calibration"). Without frequency response correction, only the rescaling to the TDR/TDT measurements is performed.

The calibration results are used for correction of subsequent measurements and can be saved in a file for later usage.

**Correction**

Calibration results are used for correction of the signal in two ways:

- DC scaling using results of open, short and match calibration.
- RF correction and deembedding of transmission loss based on short calibration.

The R&S RTP corrects for the transmission loss of the system, but not for multiple reflections between DUT and oscilloscope.

## 16.2.2 Using TDR/TDT

### 16.2.2.1 Calibrating the setup

Before you can start TDR/TDT measurements, calibrate the used cables.

You can:

- Calibrate the setup using the calibration wizard of the R&S RTP.
- Load a file with results of a previous calibration of the same setup.
- Use default calibration and correction values.

**To calibrate the cables**

The calibration procedure is described for single-ended TDR and TDT analysis. If you need only TDR or only TDT, the procedure is the same but with fewer calibration steps and fewer measurements.

1. Open [Apps]> "Analysis" > "TDR / TDT"
2. To start a new calibration, select "Start wizard".

   If a calibration was loaded or done before, the calibration steps are skipped, and the measurement dialog box is shown.
3. Select the analysis type: "TDR", "TDT", or "TDR / TDT" (both TDR and TDT measurements).
4. Select the connection: "single-ended" or "differential".
5. Select "Calibrate setup".
6. Connect the cables, and select the input channels:
   a) Connect the first cable to the "Ref" connector of the pulse source and to a channel input. This cable is for TDR measurements.

b) Select the connected channel in the dialog box.
c) Connect the second cable to the "Out" connector. Do not connect the DUT.
d) Connect a third cable to a channel input. This cable is for TDT measurements.
e) Select the connected channel in the dialog box.
f) Terminate all unused connectors of the pulse source.
g) Set up the input channels, in particular, the trigger level and vertical settings. These settings must remain unchanged during and after calibration.
h) Select "Next".

7. Calibrate with a short:
   a) Connect a short to the cable that is connected to the "Out" connector.
   b) To change the short, select the edit icon.
   c) If you use another short than the delivered standard short, select "User.defined", and enter the electrical length of the used short.
   d) Select "Calibrate".
8. Calibrate with open cable:
   a) Remove the short.
   b) Select "Calibrate".
9. Calibrate with a match:
   a) Connect a match to the cable that is connected to the "Out" connector.
   b) Select "Calibrate".
10. Calibrate with a through:
    a) Connect a through to the cable that is connected to the "Out" connector.
    b) Connect the through to the third cable.
    c) Select "Calibrate".
11. Select "Next".
12. Check the graphical calibration result. To see the details, you can display the full-size graphic, zoom in and out, and use the cursor. To see the original plot again, select "Reset".

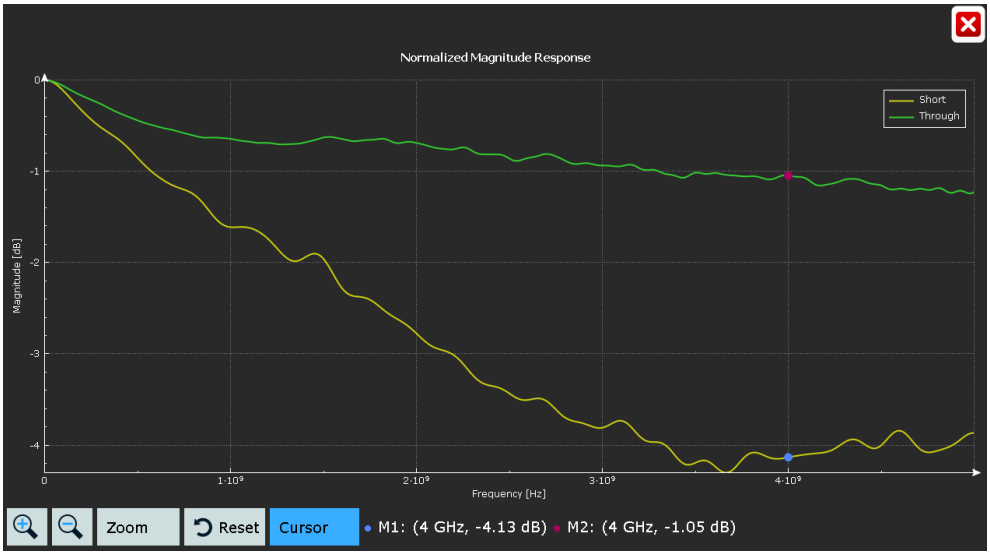



*Figure 16-1: Full-size display of calibration results, with cursor*

13. Save the calibration results for later measurements.
14. Select "Next".

    Now you can analyze the DUT.

### 16.2.2.2 Measuring the DUT

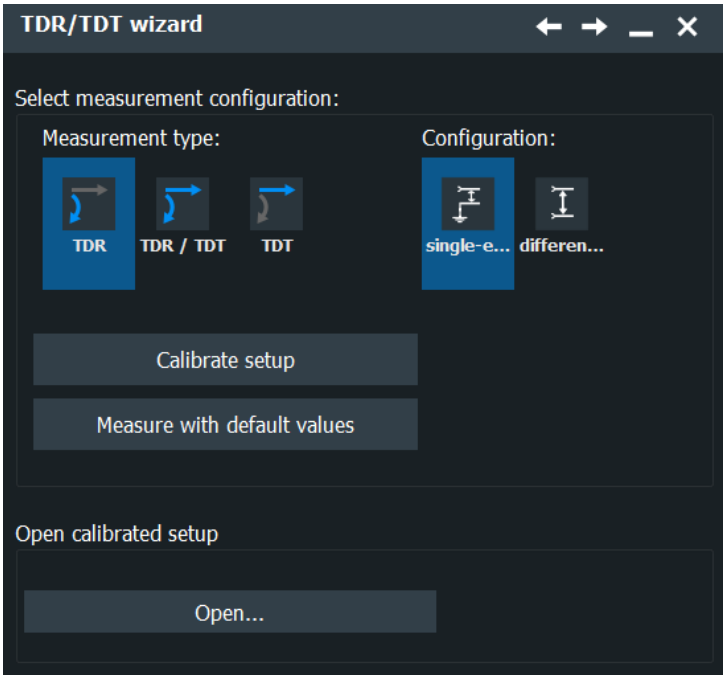
1. Open [Apps]> "Analysis" > "TDR / TDT"
2. Load an existing calibration file, or calibrate the setup as described in Chapter 16.2.2.1, "Calibrating the setup", on page 1001.
3. Discharge the DUT before connecting.
4. Connect the DUT instead of calibration standards as shown in the dialog.
5. Under "Signals", select the characteristics that you want to analyze.
6. Enter the "Bandwidth" of the DUT.
7. Select whether you want to see the time or the distance on the x-axis: "Domain".
8. If the distance is selected, enter the permittivity of the DUT: "ε_r".
9. To check the impact of the frequency response calibration on the time domain waveform, disable "Use frequency response calibration".
10. Close the dialog box.
11. Set the horizontal time scale so that the complete DUT is captured and visible on the screen.

12. Analyze the selected signals. See also: Chapter 16.2.6, "Measurement results", on page 1012

## 16.2.3 Calibration settings

### 16.2.3.1 Setup: TDR/TDT configuration

Access: [Apps]> "Analysis" > "TDR / TDT"




**Measurement type**
Selects the analysis type: "TDR" (time domain reflectometry), "TDT" (time domain transmission), or "TDR / TDT" (both TDR and TDT).

Remote command:
`TDRT:TYPE` on page 2360

**Configuration**
Selects the connection: "single-ended" or "differential" measurement.

Remote command:
`TDRT:CONFig` on page 2360

**Calibrate setup**
Starts the wizard for manual calibration. In the next steps, you connect several calibration standards and measure the correction parameters.

**Measure with default values**
Starts the measurement with default correction values instead of calibrating the setup. Default values are rough values without time reference.

Remote command:
`TDRT:DEFault` on page 2360

**Open**
Opens a saved calibration file of a previous calibration and starts the measurement.

Remote command:
`TDRT:CALibration:OPEN:NAME` on page 2360
`TDRT:CALibration:OPEN[:EXECute]` on page 2360

### 16.2.3.2 Setup: channels

Access: [Apps]> "Analysis" > "TDR / TDT" > "Calibrate setup"

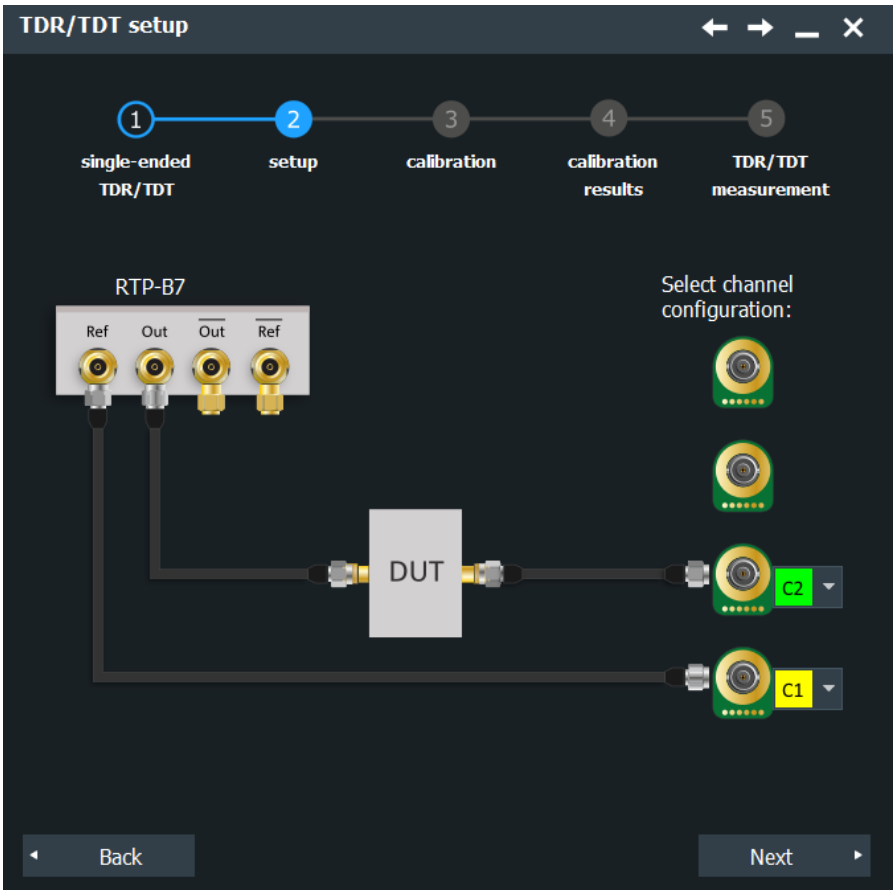



*Figure 16-2: Channel setup for single-ended TDR/TDT*

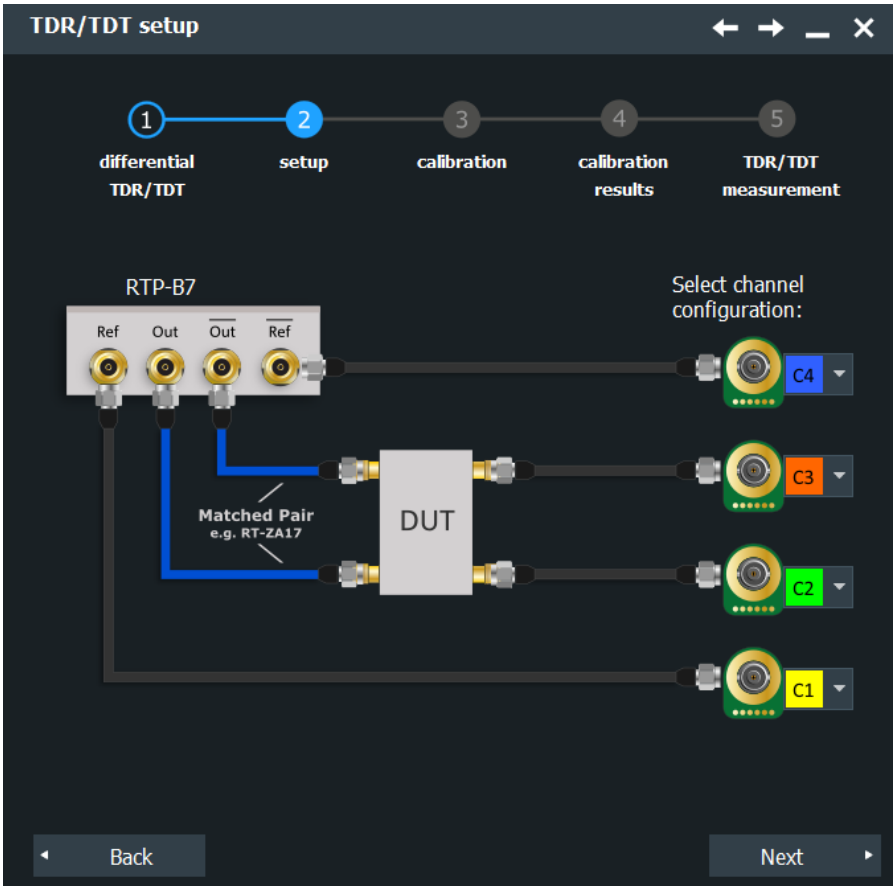



*Figure 16-3: Channel setup for differential TDR/TDT*

**Channel**
Select the input channels that are connected to the "Ref" connector and the "Out" connector of the pulse source.

For differential measurements, connect also $\overline{\text{Out}}$ and $\overline{\text{Ref}}$.

**Note:** The position of the DUT is shown for information only. Do not connect the DUT.

Remote command:
`TDRT:PULSe:OUT` on page 2361
`TDRT:PULSe:REFerence` on page 2361
`TDRT:PULSe:INVout` on page 2361
`TDRT:PULSe:INVReference` on page 2361

### 16.2.3.3 Calibration

Access: [Apps]> "Analysis" > "TDR / TDT" > "Calibrate setup" > "Next"

*Figure 16-4: Calibration of single-ended TDR/TDT*

*Figure 16-5: Calibration of differential TDR/TDT*

The used short type is shown in the wizard. To change the short, tap the "Edit" icon.

**Short type**
Selects whether the short is the Rohde & Schwarz short that is delivered with the pulse source R&S RTP-B7, or another one.

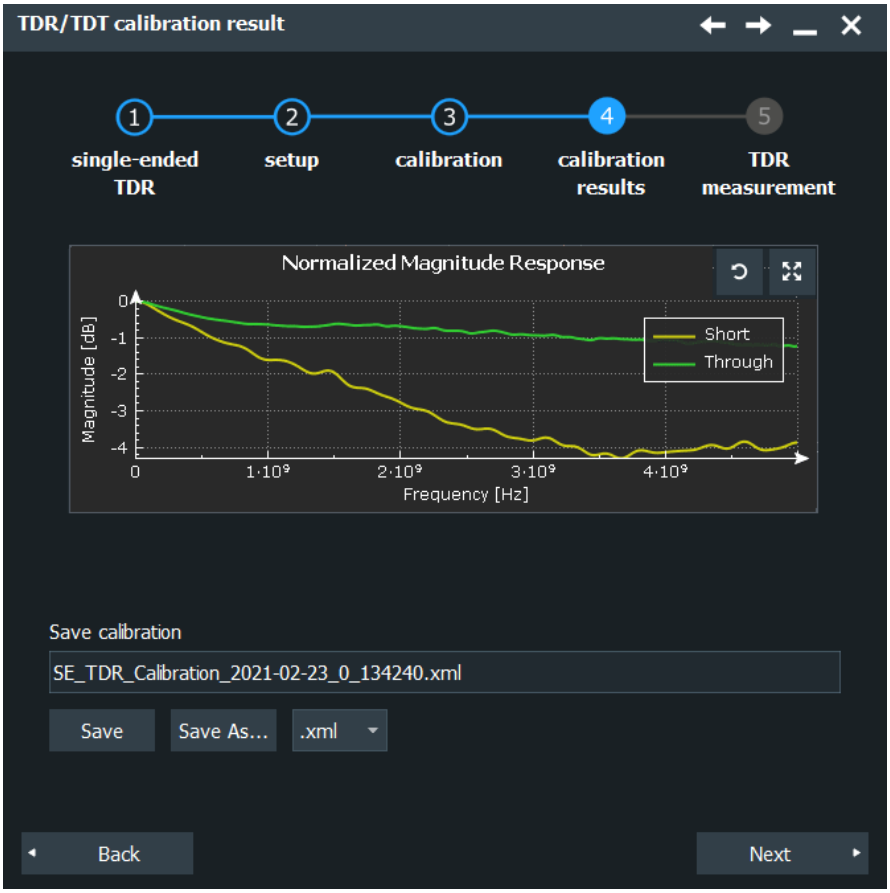
If you do not use the Rohde & Schwarz short, enter the "Electrical short length" of the used short.

Remote command:
`TDRT:SLENgth` on page 2362

**Calibrate**
Starts the calibration of the current calibration step (short, open, match through).

## 16.2.4 Calibration results

Access: [Apps]> "Analysis" > "TDR / TDT" > "Calibrate setup" > "Next"




**Magnitude response plot**
Shows the graph of magnitude over frequency for the short calibration (TDR) and through calibration (TDT).

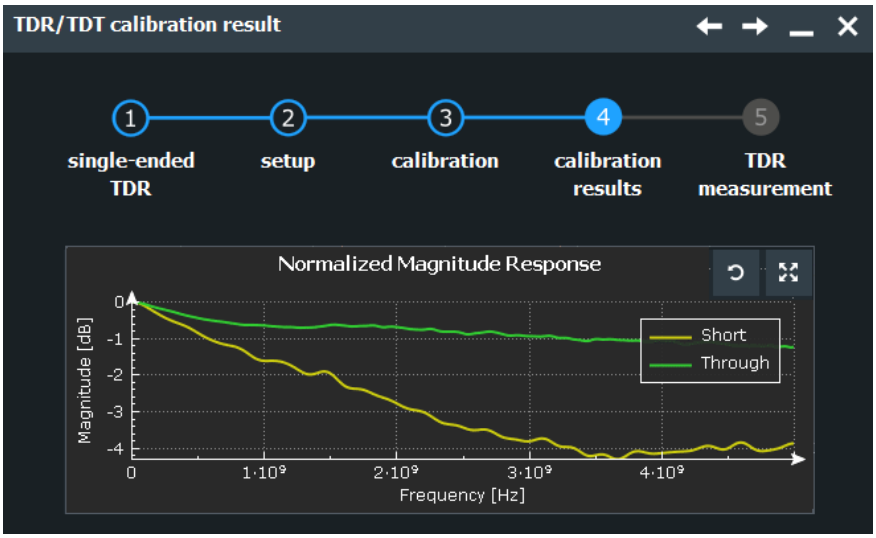



*Figure 16-6: Result of TDR and TDT calibration: smooth graph with moderate attenuation*

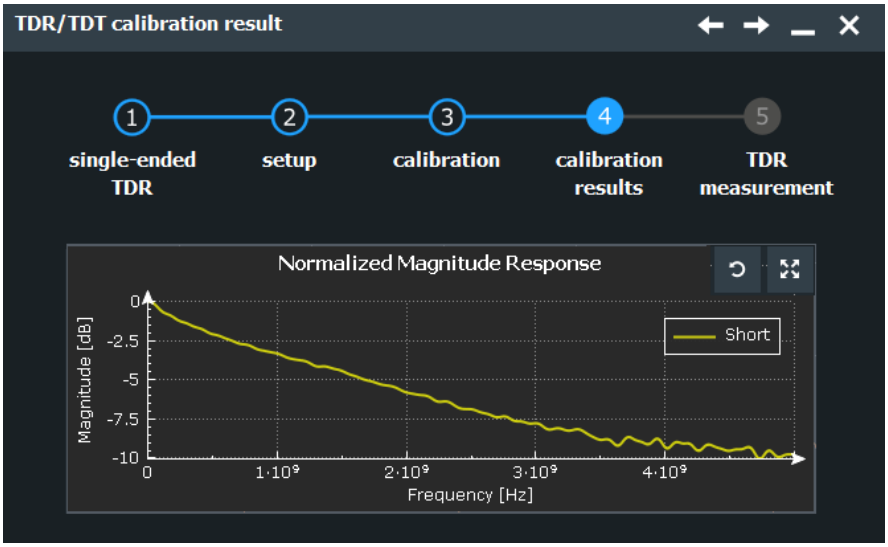


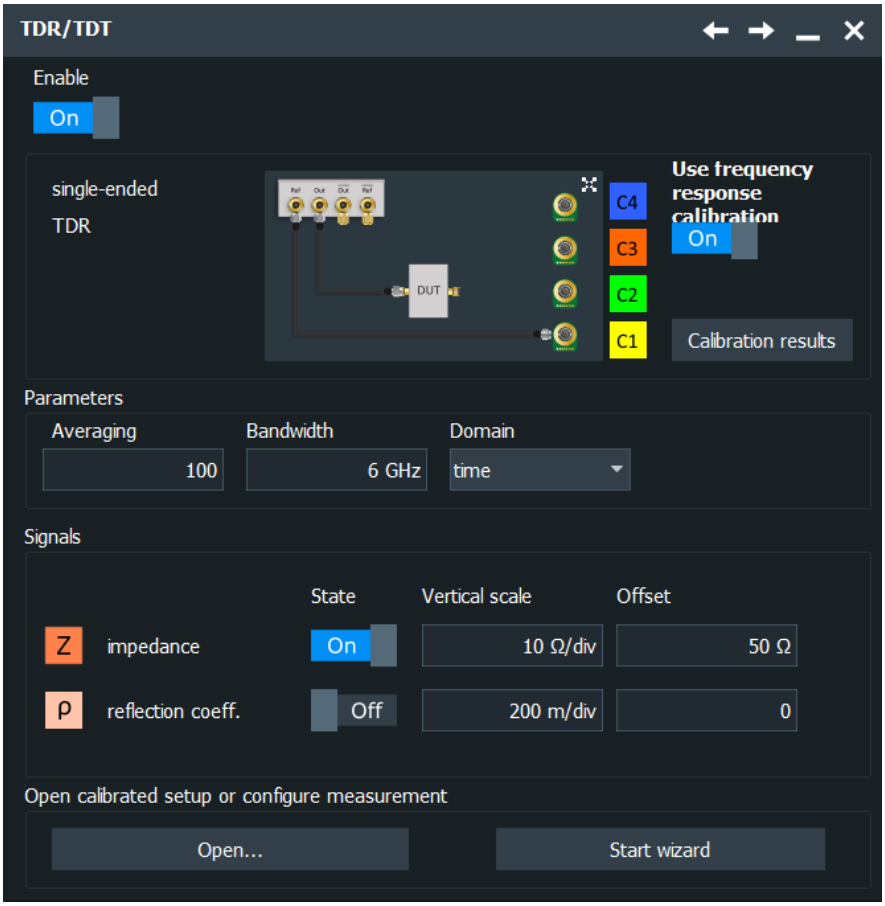

*Figure 16-7: Result of TDR calibration: high attenuation and noisy graph at higher frequencies*

**Save calibration**
Saves the current correction values to a file. Using the saved file, you can repeat the measurements at any time.

Remote command:
`TDRT:CALibration:SAVE:NAME` on page 2362
`TDRT:CALibration:SAVE[:EXECute]` on page 2362

## 16.2.5 Measurement settings

Access: [Apps]> "Analysis" > "TDR / TDT" > "Calibrate setup" > "Next"> "Next" > "Next"




**Enable**
Activates the TDR/TDT measurements.

Remote command:
`TDRT:STATe` on page 2363

**Use frequency response calibration**
Disables the frequency response calibration. Without calibration, the measured signals are only rescaled, and you can see the impact of the frequency response calibration on the time domain waveform.

The setting is not available when measuring with default values.

Remote command:
`TDRT:CALibration:USECaldata` on page 2363

**Averaging**
The "Averaging" value is the "Average count (N-single count)", which you can also set in the acquisition and trigger settings. See "Average count (N-single count)" on page 130.

Increase the value to reduce noise.

**Bandwidth**
Sets the bandwidth of the DUT.

Remote command:
TDRT:BW on page 2363

**Domain, ε_r**
Sets the domain of the x-axis: time or distance (in m or inch). If you select distance, enter also the permittivity "ε_r" of the used DUT.

Remote command:
TDRT:DOMain on page 2364
TDRT:PERMittivity on page 2364

**State**
Select the TDR/TDT signals to be displayed and analyzed.

TDR delivers the impedance Z and the reflection coefficient ρ.

TDT delivers the normalized and corrected step response.

Remote command:
TDRT:SIGNal<m>STATe on page 2364

**Vertical scale**
Set the vertical scale of the TDR/TDT signals.

Remote command:
TDRT:SIGNal<m>:SCALe on page 2364

**Offset**
Set the offset of the TDR/TDT signals.

To set the position, assign position to the vertical Position knob ("File" menu > "Front-panel Setup" > "Knobs"). Turn the vertical Position knob to change the position.

Remote command:
TDRT:SIGNal<m>:OFFSet on page 2365
TDRT:SIGNal<m>:POSition on page 2365

### 16.2.6 Measurement results

To get exact measurement results, set the horizontal time scale so that the complete DUT is captured and visible on the screen. Thus, interference between the reflected signals from the DUT is avoided. Interference can occur because the repetition rate of the pulse source is linked to the horizontal scale to maximize measurement speed.

The results of the TDR measurements are the impedance waveform Z and the waveform of the reflection coefficient ρ.

To investigate details of the waveforms, use the zoom function. For further analysis, you can use automatic and cursor measurements, histogram, mathematics and reference waveforms.

**Data export**

In the same way as channel waveforms, you can export the data of Z, ρ and H waveforms to file (one waveform per file). Raw data is not available.

See also: Chapter 12.2, "Save and recall waveform data and results", on page 482.

Remote commands:

- `EXPort:WAVeform:SOURce` on page 1614
- `TDRT:SIGNal<m>:DATA:HEADer?` on page 2365
- `TDRT:SIGNal<m>:DATA:VALues?` on page 2366

### 16.2.7 Troubleshooting

If the calibration or measurement does not show meaningful results, the following table helps you to find the cause.

| | Issue | Checks |
|---|---|---|
| Calibration | No waveform is visible. | Check the cabling.<br>Check the pulse source. |
| Calibration | Calibration with short failed | Check the cable length: 0.5 m < length < 3 m<br>Use cables with low loss.<br>Use a short that has a low impedance for high frequencies.<br>For differential TDR, use a matched pair cable, e.g. R&S RT-ZA17. |
| Calibration | Calibration with open failed | Use cables with low loss. |
| Calibration | Calibration with match failed | Use cables with low loss.<br>Use match with 50 Ω±10% for single-ended measurements.<br>Use 100 Ω±10% for differential measurements. |
| Calibration | Calibration with through failed | Use cables with low loss.<br>Check the cable length: 0.5 m < length < 3 m |
| Calibration | Failure after completion of all calibration steps | Use cables with low loss.<br>Check the pulse source. |
| Measurement | Waveform is noisy or unstable | Evaluate the calibration results for any resonances or high attenuation in frequency response.<br>Increase averaging.<br>Reduce bandwidth.<br>Disable calibration. If noise disappears, reduce attenuation or remove resonances from your setup and recalibrate. |

# 17 I/Q software interface (option R&S RTP-K11)

The option R&S RTP-K11 "I/Q software interface" acquires modulated signals and outputs I/Q data for further analysis using other software, for example, MATLAB. Rohde & Schwarz provides the R&S®VSE vector signal explorer for analysis of I/Q data on external computer.

In I/Q mode, the input signal is captured with a sample rate of 20 Gsample/s. On R&S RTP134B and R&S RTP164B instruments with more than 8 GHz bandwidth, a sample rate of 40 Gsample/s is supported.

For specifications, see the R&S RTP data sheet.


- I/Q mode ..... 1014
- I/Q sampling setup ..... 1014
- I/Q data output ..... 1020
- Deembedding in I/Q mode ..... 1021


## 17.1 I/Q mode

The I/Q mode is a special operating mode of R&S RTP. In this mode, the usual analyzing tools (cursor, zoom, measurements, FFT and so on) of the oscilloscope are deactivated. The instrument displays the magnitude of the I/Q vector to get an impression of the data before it is exported.

Triggering is performed on the input data before conversion to I/Q data, using the usual R&S RTP trigger functionality. In case of complex input signals, the instrument triggers on one component of the complex signal (I or Q) depending on the selected trigger source.

**To activate the IQ mode**

1. Press the [Apps ] key on the front panel.
2. In the "RF" tab, select "IQ".
3. In the "IQ Sampling" dialog, tap "Enable".

   The instrument switches to I/Q mode, and the setup dialog for I/Q data recording is shown.

## 17.2 I/Q sampling setup

Access: [Apps] menu > "RF" tab > "IQ" > "IQ Sampling Setup"

In the "IQ Sampling" dialog box, you configure the complete I/Q sampling for real and complex input signals.

If the values in the dialog box result in an unfavorable combination, a note is shown. The note indicates:

- "Undersampling" if the carrier frequency is too fast.
- "CF + BW/2 > DevBW", where CF is the carrier frequency, BW is the filter bandwidth, and DevBW is the maximum instrument bandwidth.
- Remote commands:
  `CHANnel<m>:IQ:WARNing[:STATe]?` on page 2371
  `CHANnel<m>:IQ:WARNing:MESSage?` on page 2372

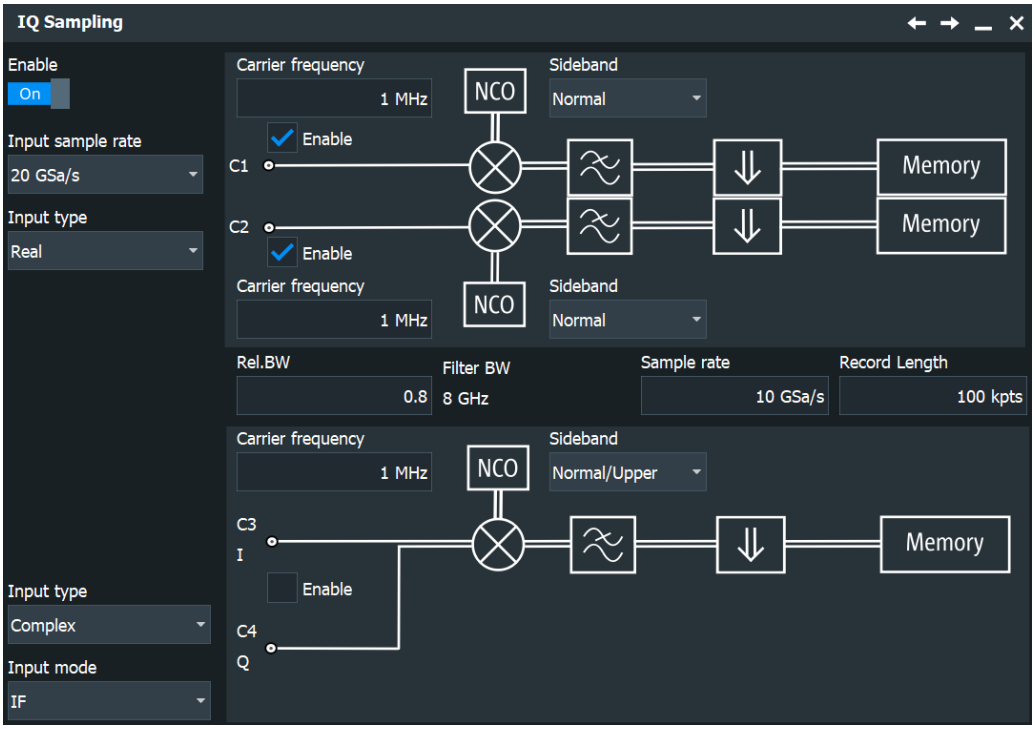



*Figure 17-1: IQ sampling settings at input sample rate of 20 GHz*

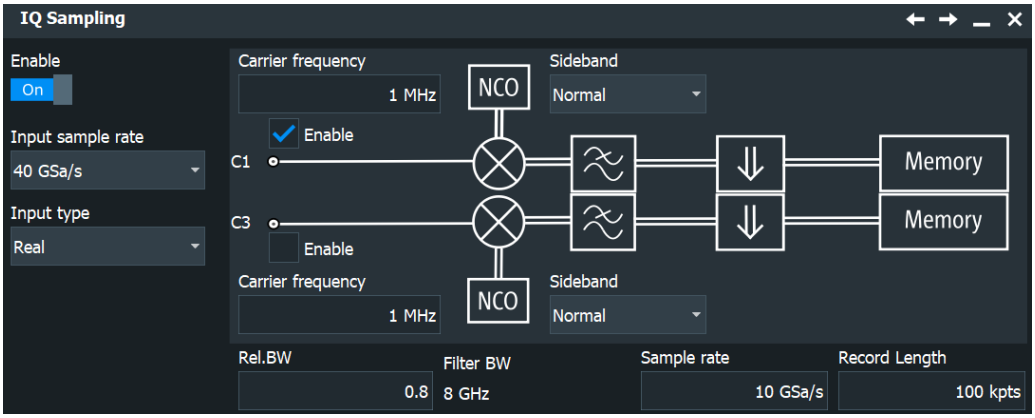



*Figure 17-2: IQ sampling settings at input sample rate of 40 GHz, available on R&S RTP134B and R&S RTP164B*

As usual, the signal icons of the enabled channels show the current settings. In I/Q mode, the I/Q settings are shown, see Figure 17-3. The left signal icon shows a complex input signal in IF range with carrier frequency 1 MHz and normal/upper sideband. C1 is the in-phase component, C2 the quadrature component. The right signal icon shows a real input signal with carrier frequency 1 MHz. C1 has inverse sideband, C2 has normal sideband.

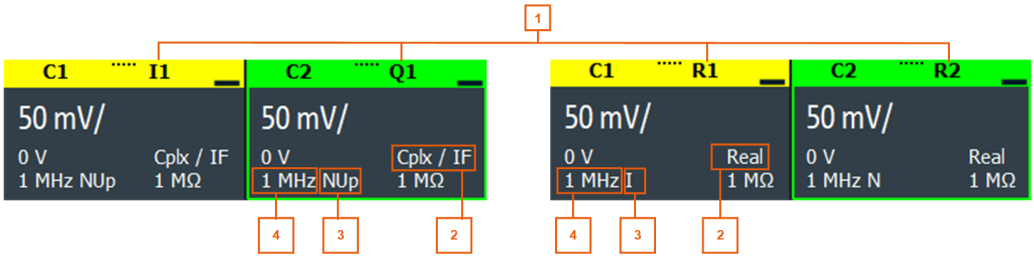



*Figure 17-3: Signal icons in I/Q mode*

1 = Signal component
2 = Input type/Input mode
3 = Sideband
4 = Carrier frequency

**Enable**
Enables the I/Q mode to record I/Q data.

The analyzing tools on the toolbar are deactivated, and the magnitude of the I/Q vector is displayed.

Remote command:
`IQ:STATe` on page 2367

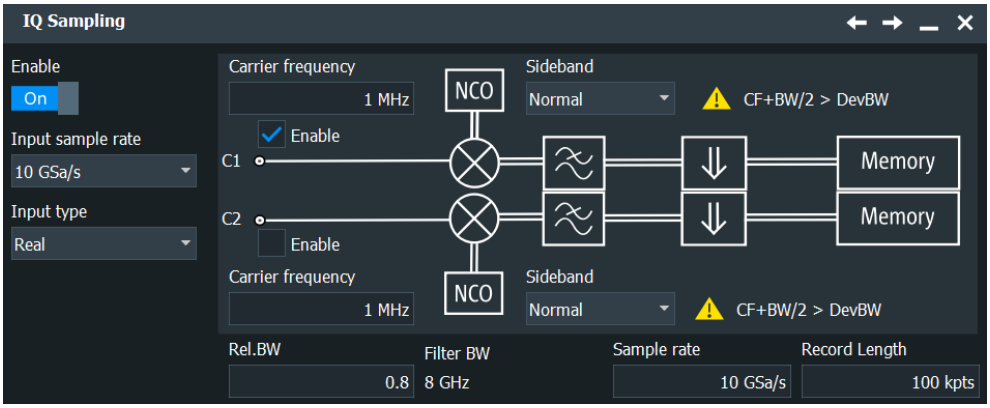
**Input sample rate**
Shows the sample rate of capturing the input data in I/Q mode: 20 Gsample/s.

If you use R&S RTP134B or R&S RTP164B instruments, you can change the sample rate to 40 Gsample/s. In this case, only 2 channels are available for real RF signals, or one complex channel. For input, Ch 1 and Ch 3 are used, Ch 2 and Ch 4 are not available. The higher sample rate also reduces the acquisition range.

Remote command:
`IQ:INSRate` on page 2368

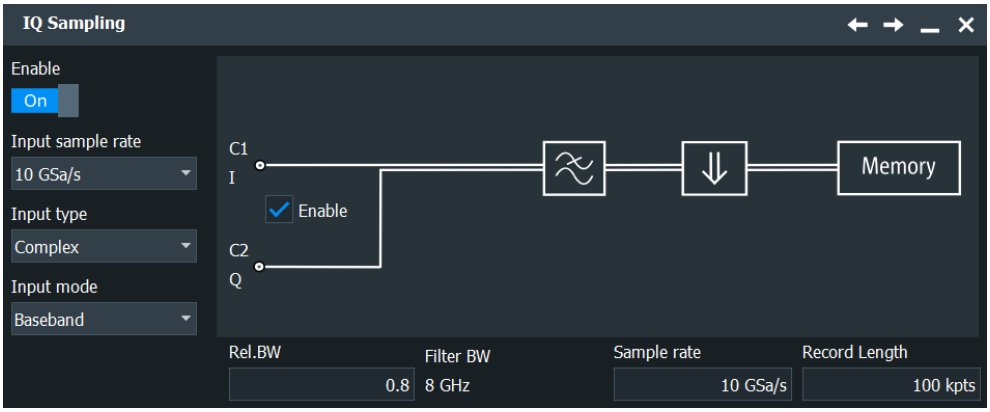
**Input type, Input mode**
"Input type" sets the format of the input signal. "Input mode" selects the frequency band of a complex input signal.

The R&S RTP I/Q software interface supports three formats of input signals.
- Real RF signals



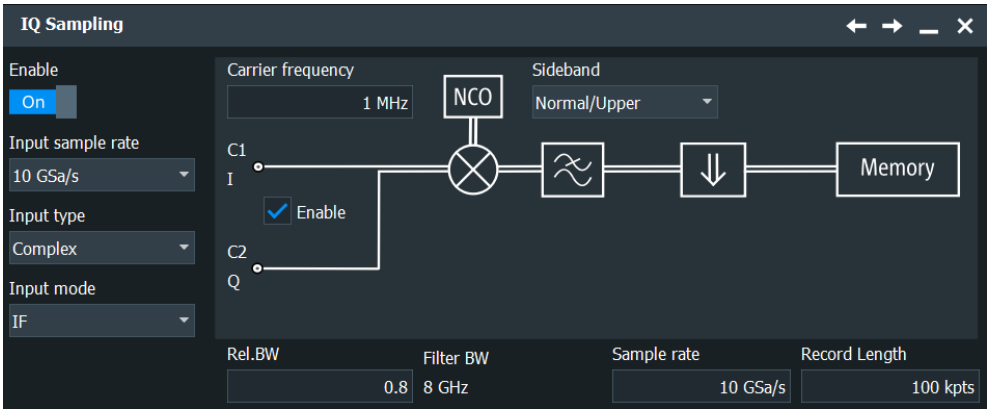

– The input signal is down-converted, filtered and resampled to the sample rate of the output I/Q signal.
– One channel is required for each input signal. Thus up to four real signals can be recorded in parallel.
– Sideband settings: see "Sideband (real input)" on page 1019.

- Complex I/Q baseband signals




– The input signal is filtered and resampled to the sample rate of the output I/Q signal.
– Two input channels are required for each input signal, one for the in-phase component, and one for the quadrature component. Thus up to two complex input signals can be recorded in parallel.

- Complex I/Q signals in low IF range



- The complex IF format requires an output sample rate of >250 Msample/s if input sample rate is 20 Gsample/s.
- The input signal is down-converted, filtered and resampled to the sample rate of the output I/Q signal.
- Two input channels are required for each input signal, one for the in-phase component, and one for the quadrature component. Thus up to two complex input signals can be recorded in parallel.
- Sideband settings: see "Sideband (complex IF input)" on page 1019.

Remote command:
`CHANnel<m>:IQ:INPType` on page 2368
`CHANnel<m>:IQ:INPMode` on page 2368

**Rel. BW, Filter BW**
"Rel. BW" sets the bandwidth factor to define the filter bandwidth:

*Filter BW = Relative BW * Sample rate*

Remote command:
`IQ:RBWidth` on page 2369
`IQ:BWIDth?` on page 2369

**Sample rate**
Sets the required sample rate of the output I/Q data.

The range depends on the input sample rate. For an input sample rate of 20 Gsample/s, the range of the output sample rate is 1 ksample/s to 20 Gsample/s. For an input sample rate of 40 Gsample/s, the range of the output sample rate is 400 Msample/s to 40 Gsample/s. See also: "Input sample rate" on page 1016.

Remote command:
`IQ:SRATe` on page 2369

**Record length**
Sets the required record length of the output I/Q data. The resulting acquisition time of the I/Q data is:

*Acquisition time = Record length / Sample rate*

R&S RTP044B, R&S RTP064B, R&S RTP084B:

The maximum record length of 40 Msample is available if the output sample rate is ≤250 Msample/s or >400 Msample/s. In the range between these values, the record length depends on the sample rate and increases linearly from 25 Msample to 40 Msample.

R&S RTP134B, R&S RTP164B:

The maximum record length of 80 Msample is available if the output sample rate is >800 Msample/s. At the minimum output sample rate of 400 Msample/s, the maximum record length is 40 MSa. In the range between 400 Msample/s and 800 Msample/s, the maximum record length is *Max record length = Sample rate / 10*.

Remote command:
`IQ:RLENgth` on page 2370

**Channel On/Off**
Switches the channel signal on or off. The signal icon appears on the signal bar. The waveform of the last acquisition is displayed in the diagram.

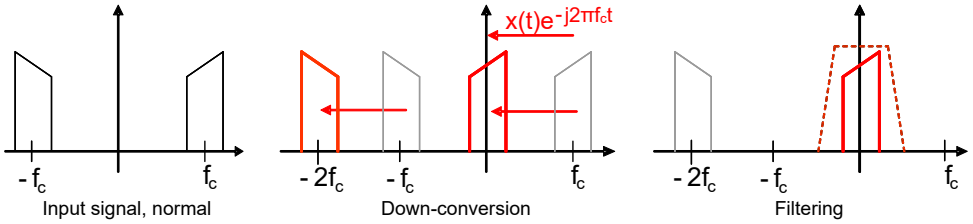
Remote command:
`CHANnel<m>:STATe` on page 1248

**Carrier frequency**
Sets the carrier frequency of the modulated RF signal or of the complex signal in IF range. The minimum carrier frequency is 10 mHz, the maximum frequency corresponds to the input sample rate.

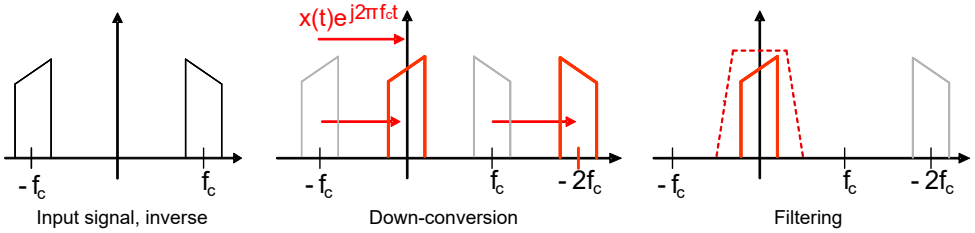
Remote command:
`CHANnel<m>:IQ:CFRequency` on page 2370

**Sideband (real input)**
Defines the frequency position of the RF spectrum in the input signal: normal or inverse. The position is important for correct down-conversion and filtering.

- "Normal" position of the RF spectrum



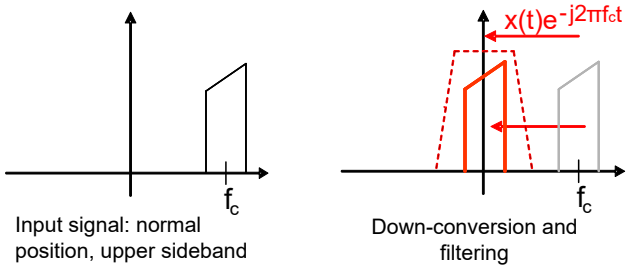

- "Inverse" position of the RF spectrum



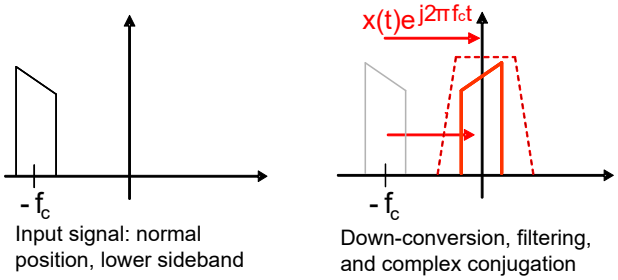

Remote command:
`CHANnel<m>:IQ:SBRF` on page 2370

**Sideband (complex IF input)**
Defines the sideband and the frequency position of complex modulated input signal in IF range.

- "Normal/Upper": Upper sideband in normal position



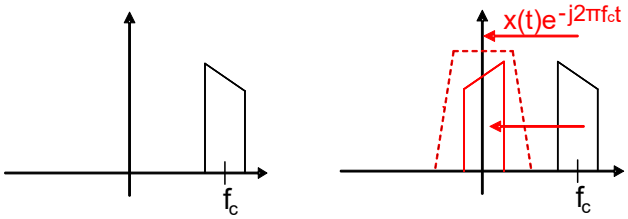

- "Normal/Lower": Lower sideband in normal position




Input signal: normal position, lower sideband

Down-conversion, filtering, and complex conjugation

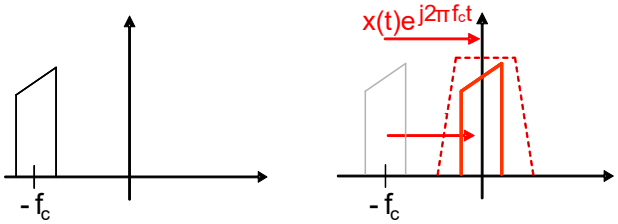
- "Inverse/Upper": Upper sideband in inverse position




Input signal: inverse position, upper sideband

Down-conversion, filtering, and complex conjugation

- "Inverse/Lower": Lower sideband in inverse position




Input signal: inverse position, lower sideband

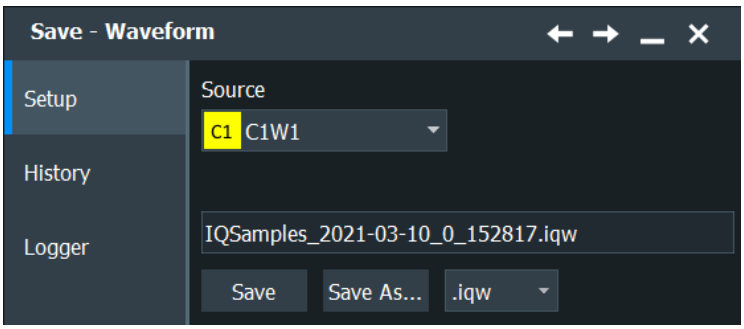
Down-conversion and filtering

Remote command:
`CHANnel<m>:IQ:SBIF` on page 2371

## 17.3 I/Q data output

The recorded I/Q data can be saved manually on hard disc or USB flash device for further analysis, or extracted using remote control from an external computer. If supported by the Rohde & Schwarz analysis software, a direct connection and data usage via LAN is also possible.

Access: "Menu" > "Save/Recall" > "Save" tab > "Waveform"



In I/Q mode, the "Save Waveform" dialog provides all functions to save the recorded I/Q data.

**Source**
Selects the channel for which the I/Q data is saved. For each input channel, a separate output file is written.

In case of a complex input signal that requires two input channels, the results of sources C1W1 and C2W1 are identical, as well as the results of C3W1 and C4W1.

At 40 Gsample/s input sample rate, Ch1Wfm1 and Ch3Wfm1 are identical, the other channels are not available.

**Save IQ data to file**
Enter the file name to save the I/Q data to. Double-tap the file name to open the file selection dialog box.

By default, the file name has the prefix "IQSamples_".

You can define a pattern for automatic naming in the "Autonaming" dialog, see Chapter 4.8.1, "Autonaming", on page 107.

| | |
|---|---|
| "Save" | Saves the I/Q data in the selected file. |
| "Save As..." | Opens the file selection dialog box and saves the I/Q data to the selected file. See also Chapter 12.5, "File selection dialog", on page 513. |
| ".iqw/.iq.tar/.bin /.xml/.csv" | Selects the file format.<br>IQW: specific format for analysis with R&S I/Q data analysis software. It contains the I and Q values in interleaved order.<br>IQ.TAR: iq-tar file format, packed format containing several files: an I/Q parameter XML file, an I/Q data binary file, and an optional I/Q preview XSLT file (style sheet). A detailed specification of the iq-tar file format is given in http://www.rohde-schwarz.com/en/manual/manual-r-s-iq-tar-file-format-specification-manuals-gb1_78701-37313.html.<br>CSV, XML, and BIN: usual formats, which are also used for common waveform export, see Chapter 12.2.1, "Waveform export files", on page 483. |

Remote command:
`CHANnel<m>:IQ:DATA[:VALues]?` on page 2372
`CHANnel<m>:IQ:DATA:HEADer?` on page 2373

## 17.4 Deembedding in I/Q mode

If realtime deembedding option R&S RTP-K122 is installed, you can use deembedding in I/Q mode. Deembedding can be enabled and configured for each channel separately, because the connection to the DUT can be different for each channel.

To use deembedding:

1. Open the deembedding setup: [Apps ] > "Analysis" > "Deembedding".

2. For each channel that needs deembedding:
   a) Enable "Real-time deembedding".
   b) Enable deembedding ("Enable" button).

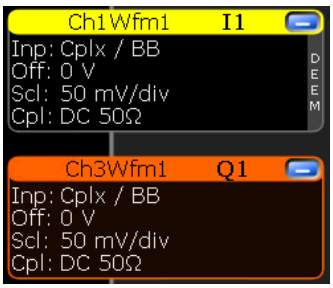



***Figure 17-4: Channel icon indicates deembedding. Here, Ch1 is deembedded.***

# 18 Jitter analysis and clock data recovery

Jitter describes the timing errors in a system. It is a significant and undesired factor in high-speed serial communication designs because it causes transmission errors.

R&S RTP provides different options for analyzing jitter:

**General jitter analysis (R&S RTP-K12)**

The jitter analysis option R&S RTP-K12 provides common analysis and visualization tools for signal integrity analysis and jitter characterization:

- Automated jitter measurements in time domain like period, frequency, cycle-to-cycle jitter and skew
- Track graph of jitter measurement results
- Jitter spectrum
- Eye mask definition and analysis
- Software-based clock data recovery

See: Chapter 18.1, "Jitter measurements (option R&S RTP-K12)", on page 1024.

**Advanced jitter analysis (R&S RTP-K133)**

The advanced jitter analysis option provides tools for detailed analysis including:

- All functionalities provided with option R&S RTP-K12
- All data for total jitter, random jitter, periodic jitter, data-dependant jitter and deterministic jitter.
- Allows separation into horizontal and vertical components.
- Provides options for extensive result presentation, like histogram, spectrum, track, jitter bathtub graphics and synthetic eye diagram.

See: Chapter 18.2, "Advanced jitter and noise (option R&S RTP-K133/K134)", on page 1033

**Advanced jitter and noise analysis (R&S RTP-K134)**

The advanced noise analysis provides tools for detailed jitter and noise analysis including:

- All functionalities provided with option R&S RTP-K133
- All data for total noise, random noise, periodic noise, data-dependant noise and deterministic noise.
- Allows separation into horizontal and vertical components.
- Provides options for extensive result presentation, like histogram, spectrum, track, noise bathtub graphics and synthetic eye diagram.

The different ways for analyzing jitter and noise are described in the following chapters.

In addition, option R&S RTP-K140/K141 provides hardware-based clock data recovery. The resulting clock edge stream can be used as trigger source for the high-speed serial pattern trigger.

See: Chapter 18.2, "Advanced jitter and noise (option R&S RTP-K133/K134)", on page 1033.


- Jitter measurements (option R&S RTP-K12) ..... 1024
- Advanced jitter and noise (option R&S RTP-K133/K134) ..... 1033
- Clock data recovery ..... 1067


# 18.1 Jitter measurements (option R&S RTP-K12)

This chapter describes the measurement setup in the "Jitter" category.


- Jitter measurements ..... 1024
- Jitter measurement settings ..... 1026
- Jitter statistics and histogram ..... 1032
- Track of jitter measurement results ..... 1032
- Jitter spectrum ..... 1033


## 18.1.1 Jitter measurements

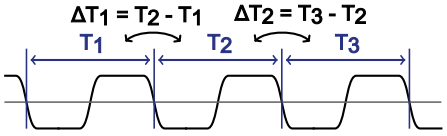
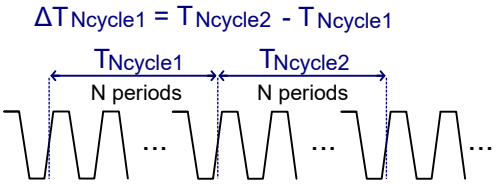
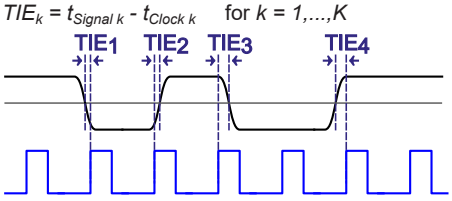
The measurement category "Jitter" gathers all measurements which are useful for jitter analysis. The category contains jitter-specific measurements. In addition, the following amplitude/time measurements are added to the jitter category, because they are useful for jitter analysis:

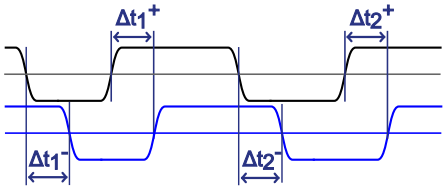
- Period
- Frequency
- Setup time
- Hold time
- Setup/Hold ratio

The amplitude/time measurements are described in "Time measurements" on page 343.

The specific jitter measurements are described in Table 18-1.

*Table 18-1: Jitter measurements*

| | Measurement | Description/Result |
|---|---|---|
| | Cycle-cycle jitter | Difference between the periods of two adjacent cycles. The measurement is based on the period measurement. You can select the slope and the reference level on which the period is measured.<br>$\Delta T_{Period\ k} = T_{Period\ k+1} - T_{Period\ k}$ for $k = 1,...,K-1$<br>  <br>See also: Chapter 18.1.2.2, "Settings for clock measurements", on page 1027 |
| | N-cycle jitter | Difference between the time of two adjacent groups of N cycles (periods) each. You can select the slope and the reference level on which the time is measured, and the number of periods in the groups.<br>$\Delta T_{Ncycle\ k} = T_{Ncycle\ k+1} - T_{Ncycle\ k}$ for $k$ = *cycle group index*<br>  |
| | Cycle-cycle width | Difference between the pulse width of two adjacent cycles. The measurement is based on the pulse width measurement. You can select the pulse polarity to be measured.<br>$\Delta T_{Pulse\ k} = T_{Pulse\ k+1} - T_{Pulse\ k}$ for $k = 1,...,K-1$ |
| | Cycle-cycle duty cycle | Difference between the duty cycle of two adjacent cycles. The measurement is based on the duty cycle measurement. You can select the pulse polarity for the duty cycle measurement.<br>$\Delta R_{Cyc\ k} = R_{Cyc\ k+1} - R_{Cyc\ k}$ for $k = 1,...,K-1$ |
| | Time interval error | Time difference between the slope of the input signal and the slope of a reference signal. The reference signal can be a captured clock waveform, or a clock generated by clock data recovery (CDR, software algorithm or hardware generation). You can select the slope and the reference level on which the TIE is measured.<br>$TIE_k = t_{Signal\ k} - t_{Clock\ k}$ for $k = 1,...,K$<br>  <br>See also: Chapter 18.1.2.3, "Data measurement settings", on page 1029 |
| | Unit interval | Period of the clock signal. If no clock signal is available, it is recovered by CDR. The period is calculated as the time difference between two consecutive clock edges of the same polarity.<br>$UI_k = t_{Clock\ k+1} - t_{Clock\ k}$ for $k = 1,...,K-1$ |

| | Measurement | Description/Result |
|---|---|---|
| | Data rate | Frequency of the clock signal. If no clock signal is available, it is recovered by CDR. The measurement is based on the unit interval measurement.<br>$R_{Clock\ k} = 1 / UI_k$ for $k = 1,...,K-1$ |
| | Skew delay | Delay between the edges of two interdependent waveforms. The measurement is a simplified variant of the "Delay" measurement assuming that both sources are similar except for the delay.<br>$Skew\ delay = \Delta t_k = t_{Source2} - t_{Source1}$ for $k = 1,...,K$<br>  <br>See also: Chapter 18.1.2.4, "Delay measurement settings", on page 1031 |
| | Skew phase | Phase difference between the edges of two waveforms.<br>$Skew\ phase = Skew\ delay / Period * 360° = \Delta t_k / \Delta T_{Period\ k} * 360°$ |

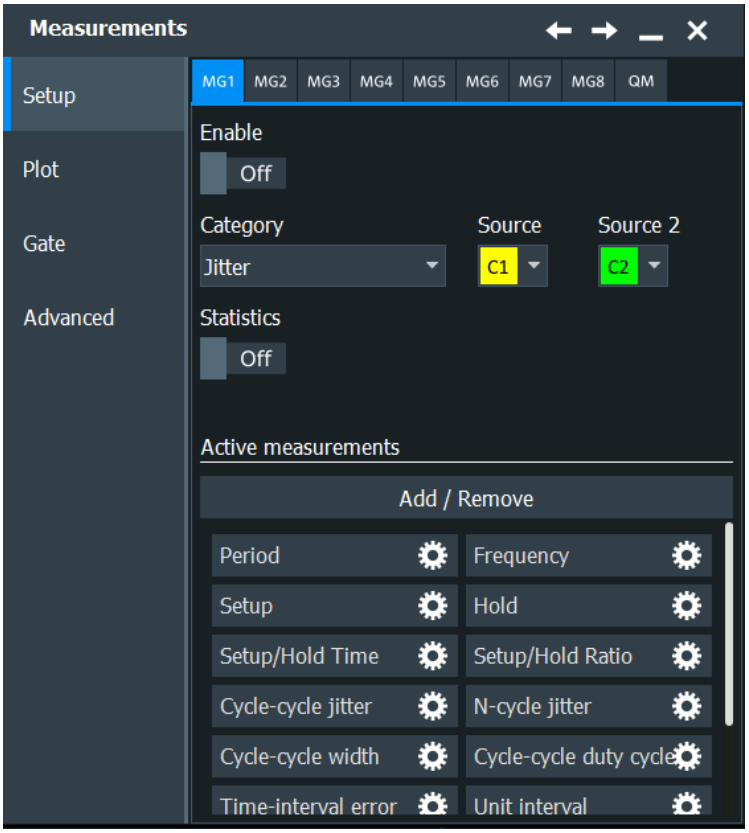
Limit and margin checks are also available for jitter measurements, see Chapter 8.2.13, "Limit and margin checks", on page 392. Limit and margin checks are based on the amplitude/time measurements.

## 18.1.2 Jitter measurement settings

### 18.1.2.1 Measurement selection

Access: [Measure] > "Meas Group" tab > "Jitter" category

Jitter measurements are only available for sources in the time domain.




**Active Measurements**
The "Active Measurements" list shows the measurements that are selected in the selected category. At least, one measurement must be selected. If further settings are available for a measurement, a settings icon is shown beside the measurement's name.

To change the selection, tap "Add / Remove".

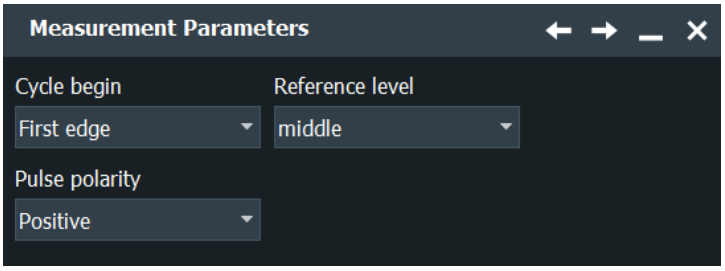
For details on the available measurements, see Chapter 18.1.1, "Jitter measurements", on page 1024.

Remote command:
`MEASurement<m>:MAIN` on page 1430
`MEASurement<m>:ADDitional` on page 1431

### 18.1.2.2 Settings for clock measurements

The cycle-cycle measurements are intended to analyze the quality of clock signals. They require a few additional settings for period and pulse width measurement. See also: Table 18-1

► To configure a measurement, tap it in the "Active Measurements" list.




**Cycle begin**
Selects the slope at which the periods and thus the jitter is measured.

The setting is available for the following measurements: cycle-cycle jitter, N-cycle jitter, and cycle-cycle duty cycle.

"First edge" Measures the period from the first edge that is found, no matter of its direction.

"Positive" Measures the period at positive going edges.

"Negative" Measures the period at negative going edges.

"Either" Measures the period at both positive and negative going edges. This option is useful, for example, to check the clock stability of a double data rate clock.

Remote command:
`MEASurement<m>:JITTer:CCSLope` on page 2374

**Data ref level / Reference level**
The setting defines the "Data ref level" for setup and hold measurements, and for time-interval error measurements (option R&S RTP-K12). It selects the reference level of the data signal on which the time is measured. The intersection of slope and reference level defines the time point for measurements.

For clock jitter measurements (option R&S RTP-K12), it sets the "Reference level" for the time measurement.

Remote command:
`MEASurement<m>:AMPTime:DATA<n>:LSELect` on page 1445

**Pulse polarity**
Sets the polarity of pulses for which the pulse width is measured to obtain the cycle-cycle width and the cycle-cycle duty cycle.

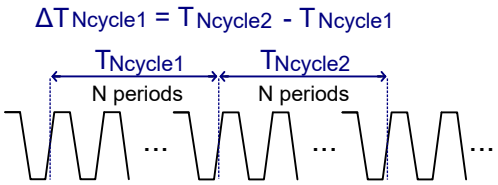
The setting is available for the following measurements: cycle-cycle width and cycle-cycle duty cycle.

"Positive" Pulse width of positive pulses is measured.

"Negative" Pulse width of negative pulses is measured.

Remote command:
`MEASurement<m>:JITTer:PULSe` on page 2375

**Number of cycles**
Sets the number of periods (cycles) that are accumulated to measure the N-cycle jitter.




Remote command:
`MEASurement<m>:JITTer:NCYCles` on page 2376

### 18.1.2.3 Data measurement settings

The following measurements are intended to analyze serial data. The clock can be a captured clock signal, or it can be recovered from the data signal using one of the clock recovery algorithms:

- Time interval error
- Unit interval
- Data rate

► To configure a measurement, tap it in the "Active Measurements" list.

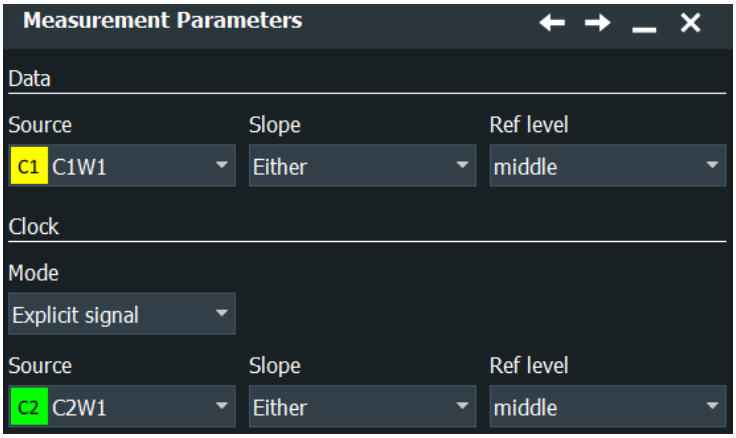



*Figure 18-1: Settings for time interval error measurement with real clock signal*

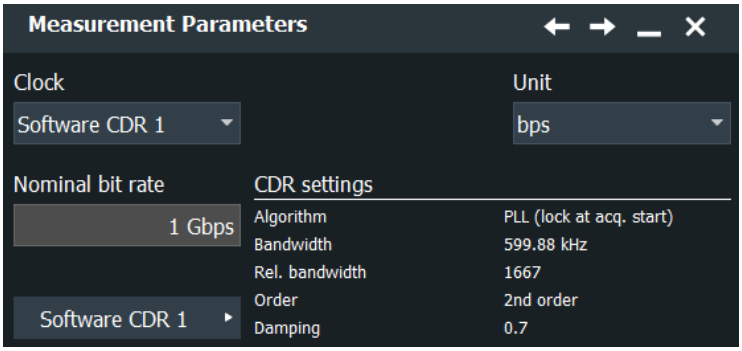



*Figure 18-2: Settings for data rate measurement with software clock recovery*

**Clock**
Defines the origin of the clock signal - either an existing clock signal or waveform, or a clock generated using one of the CDR methods.

The setting is available for the following measurements: time-interval error, unit interval and data rate.

"Explicit signal" The clock is an existing clock signal. Select the clock source, clock slope, and the reference level.

"Software CDR 1 / 2" The clock is generated by a software algorithm. The R&S RTP provides two setups for software CDR, which can be configured and used independently. To configure the CDR, tap the "Software CDR 1/2 setup" button.

Remote command:
`MEASurement<m>:JITTer:CDRMode` on page 2376

**Slope**
Sets the edges that are used for measurements.

For time-interval error measurement, you set the data slope, and the clock slope if the clock is a real signal. For unit interval and data rate measurements, you set only the clock slope if the clock is a real signal.

"Positive" The positive clock slope can be used, for example, for single data rate (SDR) signals with bit start at the positive clock edge.

"Negative" The negative clock slope can be used, for example, for SDR signals with bit start at the negative clock edge.

"Either" For clock edges, this option can be used for double data rate (DDR) signals.
For data edges, it is the most common setting.

Remote command:
`MEASurement<m>:JITTer:SOURce<n>:TIESlope` on page 2377

**Clock ref level**
Selects the reference level of the clock on which the time is measured. The intersection of slope and reference level defines the time point for measurements.

The setting is used for setup and hold measurements, and for jitter measurements (option R&S RTP-K12).

Remote command:
`MEASurement<m>:AMPTime:CLCK<n>:LSELect` on page 1444

**Data ref level / Reference level**
The setting defines the "Data ref level" for setup and hold measurements, and for time-interval error measurements (option R&S RTP-K12). It selects the reference level of the data signal on which the time is measured. The intersection of slope and reference level defines the time point for measurements.

For clock jitter measurements (option R&S RTP-K12), it sets the "Reference level" for the time measurement.

Remote command:
`MEASurement<m>:AMPTime:DATA<n>:LSELect` on page 1445

**Unit**
Sets the unit to bps (bit per second), or to Hz for analysis of radar signals. The setting is available for data rate measurements.

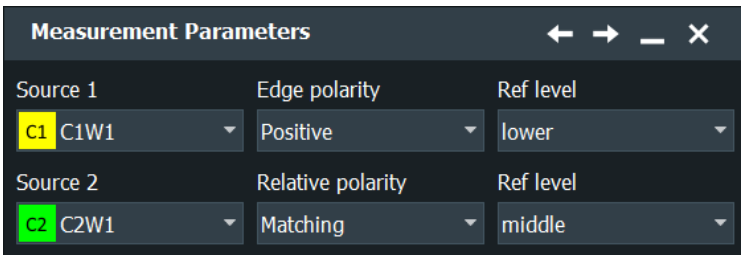
Remote command:
MEASurement<m>:JITTer:DRUNit on page 2377

### 18.1.2.4 Delay measurement settings

Skew delay and skew phase measurements are intended to measure the time difference between the edges of two waveforms. The measurements are simplified variants of the "Delay" and "Phase" measurements assuming that both sources are similar except for the delay.

See also: Table 18-1

► To configure a measurement, tap it in the "Active Measurements" list.




**Source 1, Source 2**
"Source 1" is the reference signal, and "Source 2" is the signal compared to the reference signal.

*Skew delay* = $\Delta t_k = t_{Source2} - t_{Source1}$ for $k = 1,...,K$

**Edge polarity**
Sets the edge of the first waveform from which the skew delay or phase is measured: positive, negative or both.

Remote command:
MEASurement<m>:JITTer:SKWSlope on page 2378

**Relative polarity**
Sets the edge of the second waveform relative to the first waveform.

"Matching" Measures from positive to positive edge or from negative to negative edge.

"Inverse" Measures from positive to negative edge or from negative to positive edge.

Remote command:
MEASurement<m>:JITTer:SKWRelation on page 2378

**Data ref level / Clock ref level**
See "Data ref level / Reference level" on page 351 and "Clock ref level" on page 350.

### 18.1.3 Jitter statistics and histogram

Since jitter is a random component of all signals, statistical measurement results are required to characterize the jitter.

**To get measurements statistics of a jitter measurement**

Prerequisite: the jitter measurements are selected and configured, and the measurement group is active.

1. Double tap the "Meas Results" box.

   The "Measurement" toolbar assist opens.

2. In the overlay menu, enable "Statistics state".
3. Tap "Advanced Setup".
4. On the "Measure" > "Advanced" tab, enable "Measure all events in each acquisition".

The histogram plots the density of data. It shows the frequency of occurrence of the measurement values. The maximum count of a measurement value is assigned to the full height of the histogram diagram (= 1000). All other count values are displayed relative to the maximum.

**To enable the histogram**

1. On the "Plot" tab, enable the "Histogram".
2. If the histogram is not displayed as expected, disable "Continuous auto scale" and adjust the "Meas scale".

### 18.1.4 Track of jitter measurement results

A track graph displays the results of the first jitter measurement from a single acquisition as a time-correlated waveform. To generate the track graph, multiple measurement points are required. Thus, enabling the track automatically activates "Measure all events in each acqu.".

You can measure amplitude and time on the track waveform: Configure a new measurement group, e.g. "MG2", that uses the track waveform as measurement source.

You can also zoom into the track waveform, perform cursor measurements on it, and export the track.

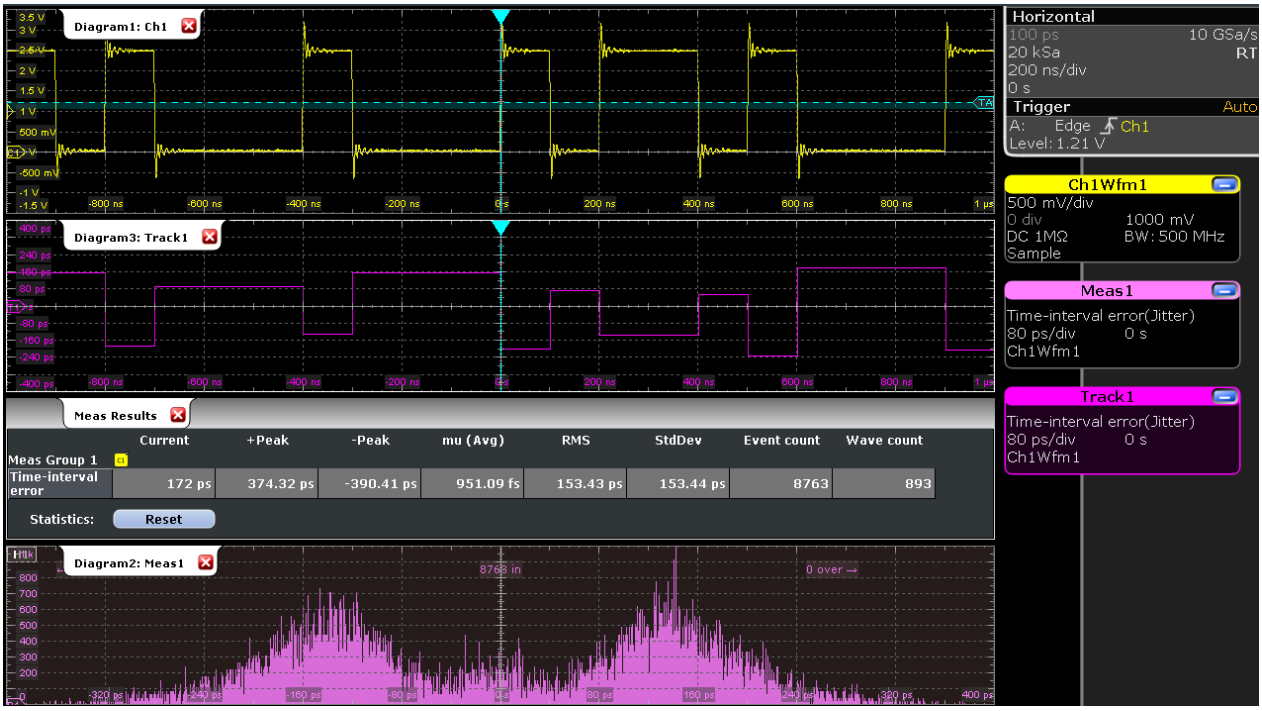



*Figure 18-3: TIE measurement with CDR trigger, 10 Mbps, with track and histogram*

### 18.1.5 Jitter spectrum

An extension of jitter measurements is the FFT analysis of track waveforms of jitter measurements. The results are displayed in the frequency domain as a jitter spectrum. The modulation frequency is displayed on the horizontal axis, and the amplitude of modulation on the vertical axis. Using FFT, periodic components can be detected.

## 18.2 Advanced jitter and noise (option R&S RTP-K133/K134)

This chapter describes the settings of the advanced jitter option R&S RTP-K133 and of the advanced noise option R&S RTP-K134.

### 18.2.1 About jitter

Jitter typically consists of various jitter components that are caused by different jitter sources, see Figure 18-4.

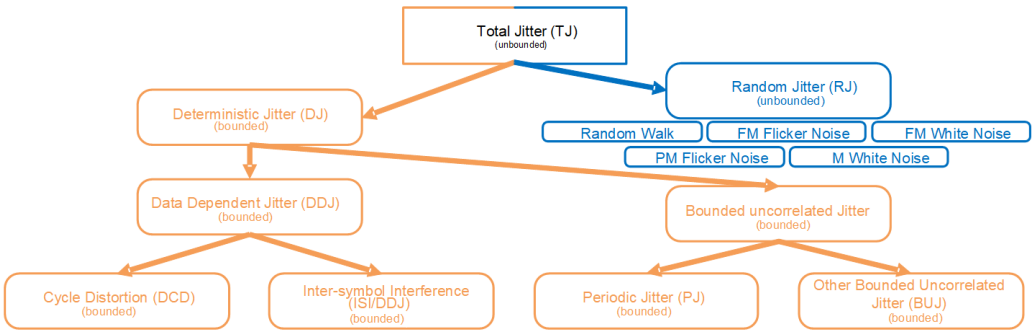



***Figure 18-4: Jitter decomposition tree***

With the advanced jitter option R&S RTP-K133, you can perform different measurements for calculating and displaying the different jitter components.

For the analysis of jitter, it is important to understand these sources and the contributions to the total jitter (TJ). For an analytical approach, a jitter model is frequently used, which splits jitter into the two major categories of random and deterministic jitter.

The jitter of the different components has different causes. Knowing what jitter component is influencing your system can help you identify what phenomenon causes the jitter.

### Deterministic jitter

Deterministic jitter (DJ), also called systematic jitter, is further broken down into periodic jitter (PJ) and data-dependent jitter (DDJ). Deterministic jitter is bounded and specified as peak-to-peak value.

### Periodic jitter

Periodic jitter (PJ) is caused by a periodic disturbance. Though this signal is not necessarily sinusoidal, it is frequently also named sinusoidal jitter. The amplitude of the periodic signal bounds the jitter.

A strong local RF oscillator, a switch-mode power supply, undesired crosstalk or an unstable, oscillating PLL cause period jitter due to an unintentional coupling into the signal.

### Data-Dependent jitter

Inter-symbol interference (ISI) causes data-dependent jitter (DDJ). When ISI is present, the signal is disturbed with an attenuated, time-shifted copy of itself or spectral parts of itself.

### Random jitter

Random jitter is unbounded and commonly specified by the standard deviation σ. Due to its irregular nature, random jitter (RJ) is uncorrelated to any other signal and unpredictable in timing behavior. Contributions to the random jitter are thermal noise, shot noise, 1/f noise and other physical effects.

### 18.2.2 About noise

Noise describes the amplitude deviation of a data symbol at the sampling time respective to a corresponding reference level. These deviations or level errors (LE) are caused by various sources.

The total noise is split into two main components: random noise (RN) and deterministic noise (DN). DN consists of periodic noise (PN) and non-periodic components.

The advanced noise algorithm enables various selectable result plots of the activated noise components in the results tab.

### 18.2.3 Configuring the advanced jitter

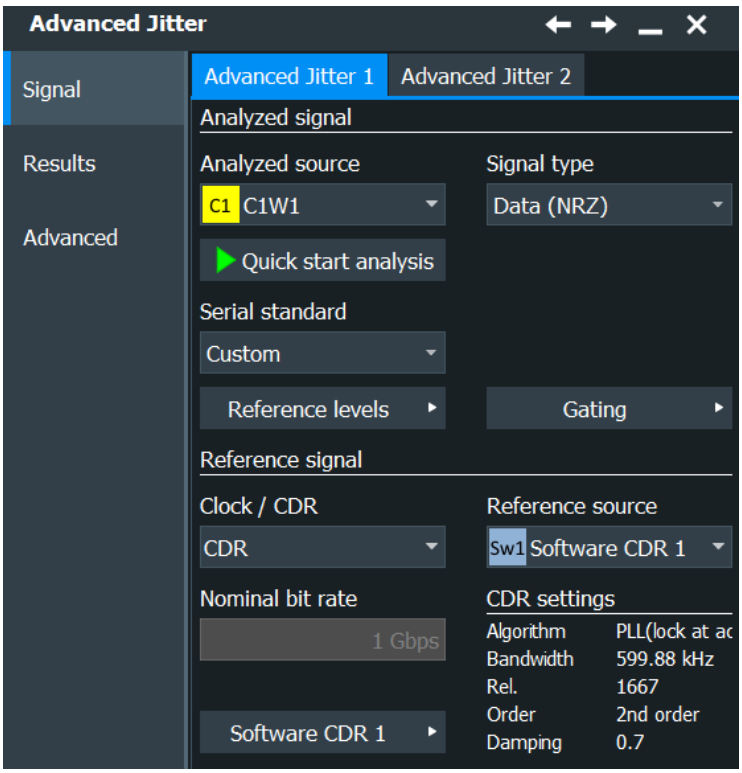
To perform an advanced jitter analysis:

1. Open the "Analysis" > "Advanced Jitter" menu.
2. In the "Signal" tab, set up the analyzed signal and reference signal.
3. In the "Advanced Settings" tab, configure the "Data dependent" settings.
4. If necessary, enable "Periodic"/"(Other) bounded uncorrelated".
5. In the "Result" tab, enable all general and component-specific results diagrams, that you want to display.
6. In the "Result" tab, "Add/ remove components". A dialog opens, where you can enable all required measurements.
7. Press "Enable" to view all selected measurement results and diagrams.

### 18.2.4 Signal settings

Access: Apps> "Analysis" > "Advanced Jitter / Noise" > "Signal" tab

In the "Signal" tab, you can set up the signal you want to analyze and define the reference clock signal.




**Analyzed source**
Selects the source for the advanced jitter analysis.

Remote command:
`ADVJitter<m>:SIGNal:SOURce` on page 2380

**Serial standard**
For some serial standards, the bit rate is known to the instrument. If you select the standard, the nominal bit rate for the advanced jitter analysis is set automatically.

The serial standards are available only if the instrument is able to support its maximum nominal bit rate, see "Maximum nominal bit rate" on page 242.

| Serial standard | Nom. bit rate | Bandwidth | Rel. bandwidth | Order | Damping |
|---|---|---|---|---|---|
| USB 2.0 | 480 Mbps | 1667 | 287.947 kHz | 1st | - |
| USB 3.1 Gen 1 | 5 Gbps | 500 | 10 MHz | 2nd | 0.7 |
| USB 3.1 Gen 2 | 10 Gbps | 667 | 14.99 MHz | 2nd | 0.7 |
| Displayport 1.1 | 1.296 Gbps | 1667 | 777.44 kHz | 1st | - |
| Displayport 1.2 | 4.32 Gbps | 1667 | 2.59 MHz | 1st | - |
| Fibre Channel 1x | 1 Gbps | 1667 | 599.88 kHz | 1st | - |

| Serial standard | Nom. bit rate | Bandwidth | Rel. bandwidth | Order | Damping |
|---|---|---|---|---|---|
| Fibre Channel 2x | 2 Gbps | 1667 | 1.2 MHz | 1st | - |
| Fibre Channel 4x | 4 Gbps | 1667 | 2.4 MHz | 1st | - |
| Fibre Channel 8x | 8 Gbps | 1667 | 4.8 MHz | 1st | - |
| Fibre Channel 16x | 14.025 Gbps | 1667 | 8.41 MHz | 1st | - |
| HDMI 1.2 | 1.32 Gbps | 1667 | 791.84 kHz | 1st | - |
| HDMI 1.4 | 3.4 Gbps | 1667 | 2.04 MHz | 1st | - |
| HDMI 2.0 | 4.8 Gbps | 1667 | 2.88 MHz | 1st | - |
| PCIe Gen 1 | 2.5 Gbps | 1667 | 1.5 MHz | 2nd | 0.7 |
| PCIe Gen 2 | 5 Gbps | 1000 | 5 MHz | 2nd | 0.7 |
| PCIe Gen 3 | 8 Gbps | 500 | 16 MHz | 2nd | 0.7 |
| PCIe Gen 4 | 16 Gbps | 500 | 32 MHz | 2nd | 0.7 |
| SATA Gen 1 (1/500) | 1.5 Gbps | 500 | 3 MHz | 2nd | 0.7 |
| SATA Gen 1 (1/1667) | 1.5 Gbps | 1667 | 899.82 kHz | 2nd | 0.7 |
| SATA Gen 1 | 1.5 Gbps | 1667 | 899.82 kHz | 2nd | 0.7 |
| SATA Gen 2 (1/500) | 3 Gbps | 500 | 6 MHz | 2nd | 0.7 |
| SATA Gen 2 (1/1667) | 3 Gbps | 1667 | 1.8 MHz | 2nd | 0.7 |
| SATA Gen 2 | 3 Gbps | 1667 | 1.8 MHz | 2nd | 0.7 |
| SATA Gen 3 (1/500) | 6 Gbps | 500 | 12 MHz | 2nd | 0.7 |
| SATA Gen 3 (1/1667) | 6 Gbps | 1667 | 3.6 MHz | 2nd | 0.7 |
| SATA Gen 3 | 6 Gbps | 1667 | 3.6 MHz | 2nd | 0.7 |
| SAS Gen 1 | 3 Gbps | 1667 | 1.8 MHz | 1st | - |
| SAS Gen 2 | 6 Gbps | 1667 | 3.6 MHz | 1st | - |
| SAS Gen 3 | 12 Gbps | 1667 | 7.20 MHz | 1st | - |

| Serial standard | Nom. bit rate | Bandwidth | Rel. bandwidth | Order | Damping |
|---|---|---|---|---|---|
| Thunder-bolt 10G | 10.3125 Gbps | 1667 | 6.19 MHz | 1st | - |
| XAUI | 3.75 Gbps | 1667 | 2.25 MHz | 1st | - |

Remote command:
`ADVJitter<m>:SIGNal:SERStandard` on page 2386

**Signal Type**
Select, if the analyzed signal is a data or clock signal.

Remote command:
`ADVJitter<m>:SIGNal:TYPE` on page 2380

**Quick start analysis**
Starts a quick advanced jitter analysis.

Before starting the quick start analysis, set the Analyzed source and the Signal Type.

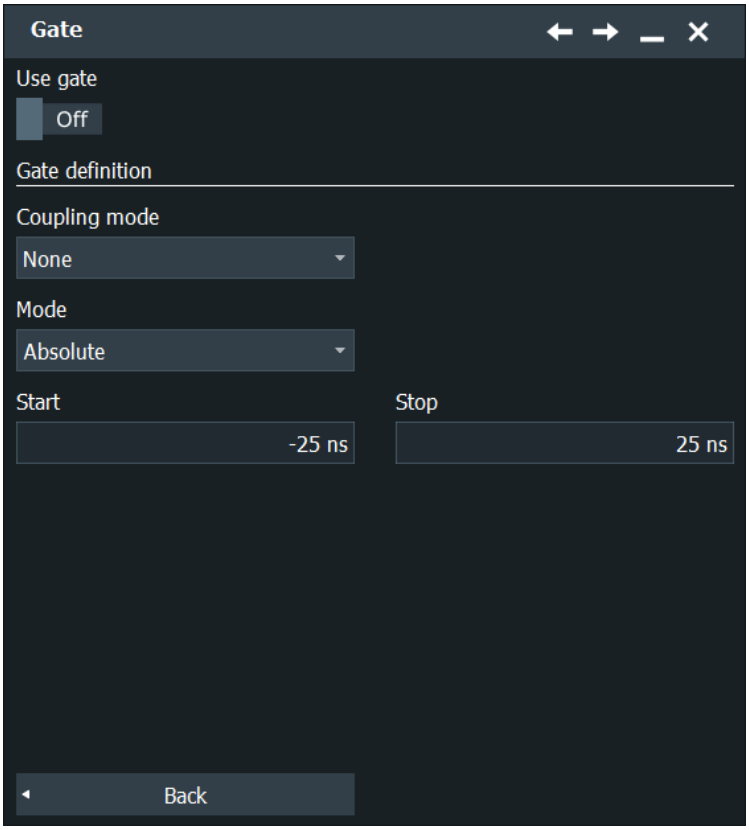
The quick start analysis performs the following:

- A bit rate estimation is performed on the applied signal.
- A software CDR of second order is used as reference signal for the jitter analysis.
- Enables basic measurement "Event Count" and advanced jitter components: TJ@BER, RJ, TJ, DJ and PJ.
- In the "Result" dialog it enables:
  - "Bathtub" for total jitter @ BER (TJ@BER).
  - "Step response" for the analyzed source signal.
  - "Histogram" for RJ, TJ, DJ, DDJ and PJ.
- Enables the advance jitter measurement analysis.

Remote command:
`ADVJitter<m>:SIGNal:QUICkmeas` on page 2380

**Reference Level**
Opens a dialog to set the reference levels. For details, see Chapter 8.2.4, "Reference levels", on page 332.

**Gating**
Opens a dialog to set up the gate of the waveform for the advanced jitter analysis. The gate limits the measurement to a user-defined range of the waveform.




**Use gate ← Gating**
Enables the gating setup.

Remote command:
ADVJitter<m>:SIGNal:GATE:STATe on page 2386

**Coupling mode ← Gating**
Selects if the gate is coupled to the zoom or cursor.

To use the coupling mode, you have to enable a cursor or zoom first.

Remote command:
ADVJitter<m>:SIGNal:GATE:COUPling on page 2383

**Cursor / Zoom ← Gating**
Selects the cursor/ zoom the gate is coupled to, if "Coupling mode" is set to "Cursor"/"Zoom".

Remote command:
ADVJitter<m>:SIGNal:GATE:CURSor on page 2384
ADVJitter<m>:SIGNal:GATE:ZDIagram on page 2385

**Mode ← Gating**
Selects if the gate area is defined in relative values (percent) or absolute.

Remote command:
ADVJitter<m>:SIGNal:GATE:RANGe:MODE on page 2385

**Start / Stop ← Gating**
Sets the start/stop time of the gate area.

Remote command:
ADVJitter<m>:SIGNal:GATE:RANGe:ABSolute:STARt on page 2384
ADVJitter<m>:SIGNal:GATE:RANGe:ABSolute:STOP on page 2384

**Relative start / stop ← Gating**
Sets the relative start/stop of the gate area in percent.

Remote command:
ADVJitter<m>:SIGNal:GATE:RANGe:RELative:STARt on page 2385
ADVJitter<m>:SIGNal:GATE:RANGe:RELative:STOP on page 2385

**Clock / CDR**
Selects if a clock or a CDR signal is used as a reference signal.

"CDR" Uses software clock data recovery.

"Clock" Uses an explicit clock signal. The clock must use an extra channel.

Remote command:
ADVJitter<m>:SIGNal:REFSource on page 2381

**Reference source**
Selects the reference source for the clock/CDR signal.

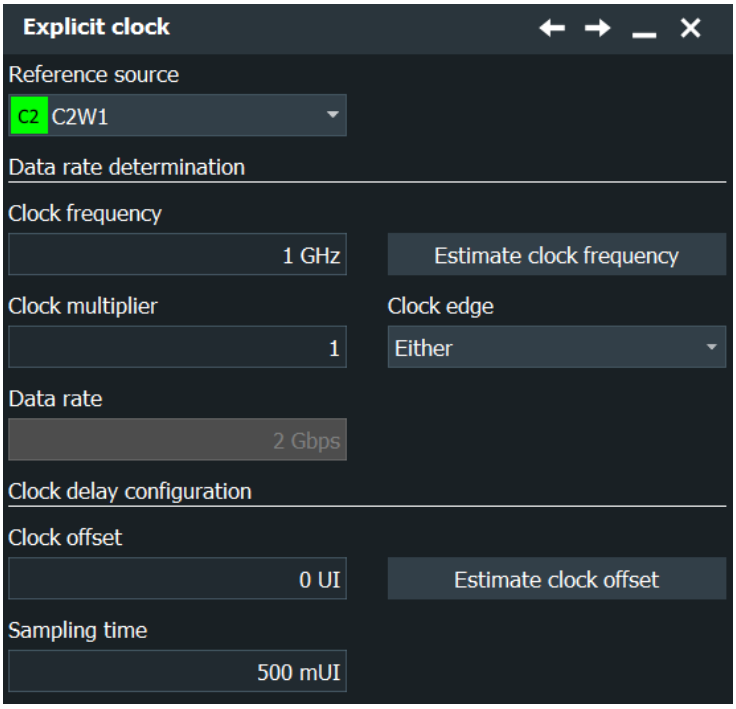
Remote command:
ADVJitter<m>:SIGNal:CLKSource on page 2381
ADVJitter<m>:SIGNal:CDRMode on page 2381

**Nominal bit rate**
Sets /displays the nominal bit rate. It corresponds to the data rate of the data stream from which the clock is to be recovered.

Remote command:
ADVJitter<m>:SIGNal:BITRate on page 2380

**Software CDR Setup**
Opens a dialog for configuring the CDR. For details, see Chapter 18.3, "Clock data recovery", on page 1067.

**Explicit clock**
Opens a dialog to define the setting of the explicit clock.




**Clock frequency ← Explicit clock**
Sets the clock frequency of the explicit clock signal. If this value is unknown, you can use "Estimate clock frequency".

Remote command:
`ADVJitter<m>:SIGNal:CLFRequency` on page 2382

**Estimate clock frequency ← Explicit clock**
Calculates an estimate value for the clock frequency.

Remote command:
`ADVJitter<m>:SIGNal:CLEStimation` on page 2382

**Clock multiplier ← Explicit clock**
Sets a value for the clock multiplier, the ratio of an internal clock rate to the externally supplied clock.

Remote command:
`ADVJitter<m>:SIGNal:CLFFactor` on page 2382

**Clock edge ← Explicit clock**
The clock edge sets the point in time when the state of the data signal is checked:

"Positive" Rising edge, a positive voltage change.

"Negative" Falling edge, a negative voltage change.

"Either" Both the rising and the falling edge.

Remote command:
`ADVJitter<m>:SIGNal:EDGE` on page 2381

**Data rate ← Explicit clock**
Displays the calculated data rate. It is calculated as follows:

*"Data rate" = "Clock frequency" * "Clock multiplier"*X*

Where *X* is:

- 1 for "Clock edge"= "Positive" or "Negative"
- 2 for "Clock edge"= "Either"

Remote command:
`ADVJitter<m>:SIGNal:DATRate?` on page 2383

**Clock offset ← Explicit clock**
Sets a clock offset. If this value is unknown, you can use "Estimate clock offset".

Remote command:
`ADVJitter<m>:SIGNal:CLOFfset` on page 2383

**Estimate clock offset ← Explicit clock**
Calculates an estimate value for the clock offset.

Remote command:
`ADVJitter<m>:SIGNal:CFEStimation` on page 2382

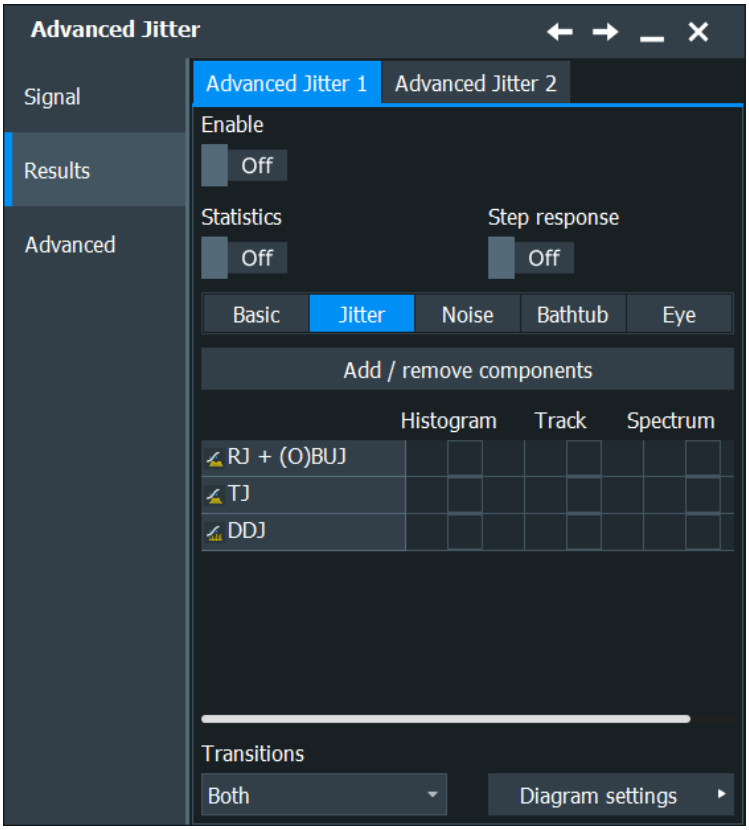
**Sampling time ← Explicit clock**
Sets a sampling time for the explicit clock signal, an offset for the clock edge in relation to the bit start. The clock edge sets the beginning of the unit interval. The sampling time is a number between 0 and 1. Value 0 sets the clock edge to the beginning of the bit period; value 0.5 sets the clock edge to the middle of the bit period.

Remote command:
`ADVJitter<m>:SIGNal:SAMTime` on page 2383

### 18.2.5 Results

Access: "Analysis" menu > "Advanced Jitter" > "Result" tab.

In this tab, you can select the jitter components measurements that you want to perform and enable the results.




The advanced jitter option provides different ways to display your measurement results. In the "Result" tab, you can select which measurement results you want to display.

Figure 18-5 and Figure 18-6 show an example of the displayed advanced jitter and noise analysis results.

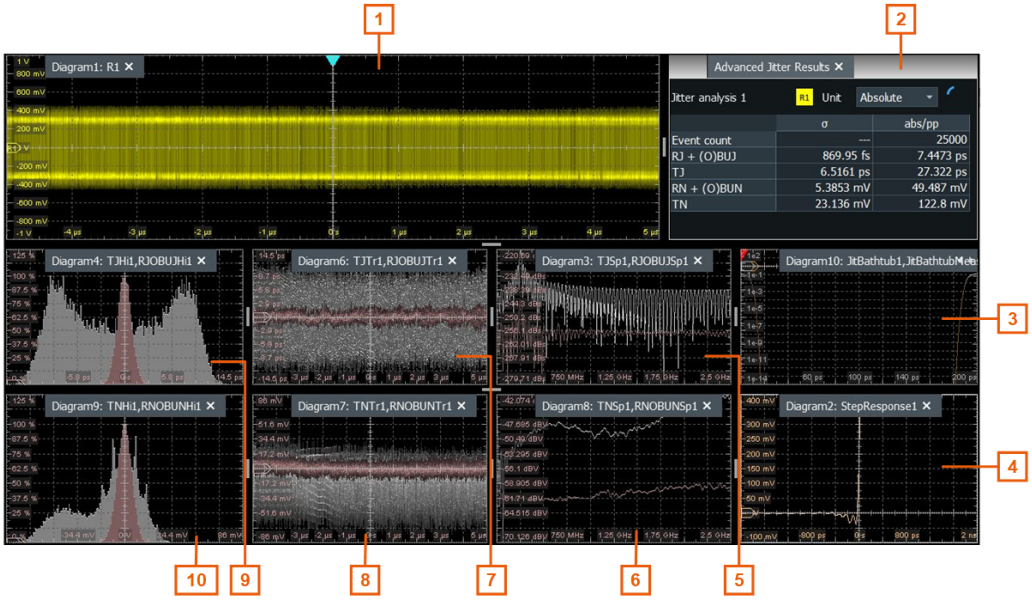



*Figure 18-5: Advanced jitter and noise analysis results display*

1 = Analyzed signal spectrum
2 = Advanced jitter results
3 = Jitter bathtub
4 = Noise bathtub
5 = Jitter spectrum
6 = Noise spectrum
7 = Jitter track
8 = Noise track
9 = Jitter histogram
10 = Noise histogram

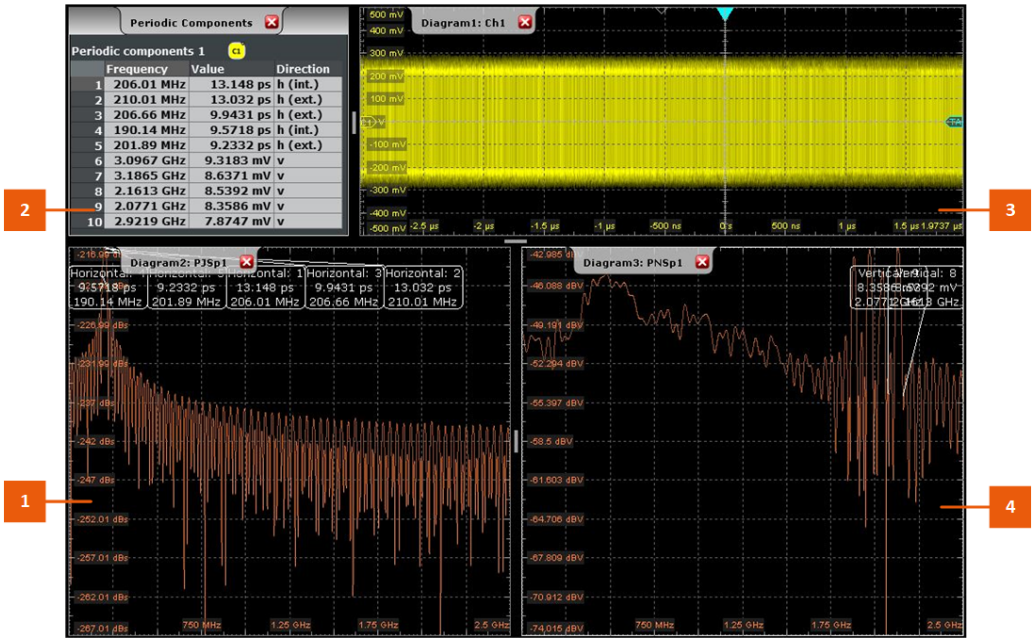



*Figure 18-6: Advanced jitter and noise periodic results display*

1 = Periodic jitter spectrum
2 = Periodic components results
3 = Analyzed signal spectrum
4 = Periodic noise spectrum

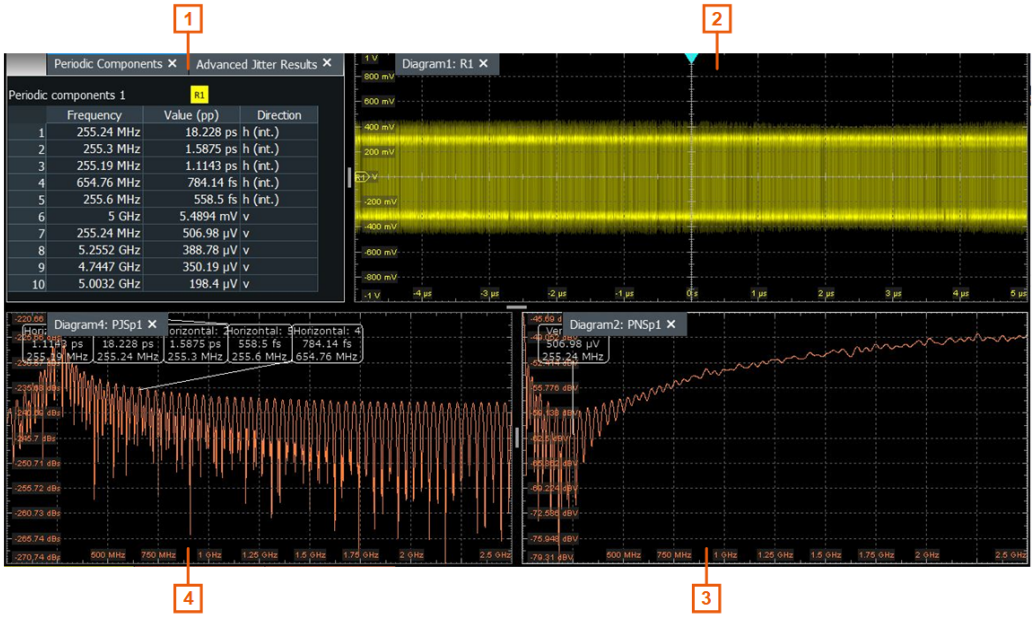


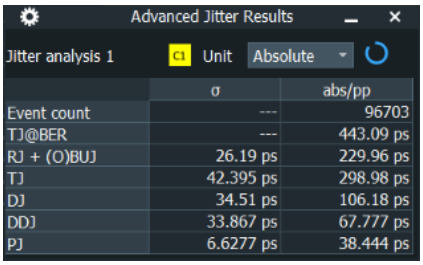

*Figure 18-7: Advanced jitter and noise periodic results display*

1 = Periodic components results
2 = Analyzed signal spectrum
4 = Periodic jitter spectrum
3 = Periodic noise spectrum

### 18.2.5.1 General settings

In the "General Settings" section of the "Results" dialog, you can enable the advanced jitter analysis, and select which results you want to display.

**Enable**
Enables the advanced jitter analysis. It displays all enabled diagrams and displays a table with the results for the component measurements. It also displays the measurement results for all enabled Basic components, Jitter Components.

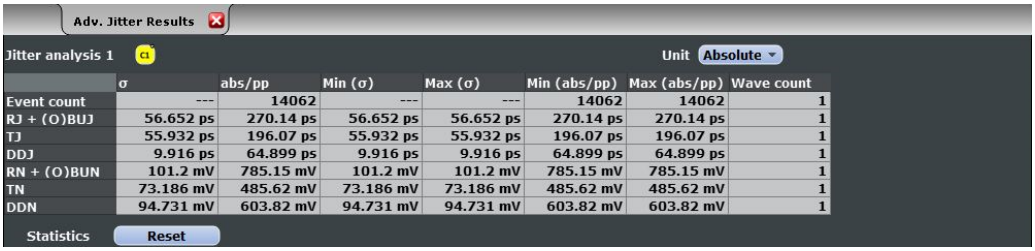



Remote command:
`ADVJitter<m>:RESult:ENABle` on page 2389

**Statistics**
Enables the calculation and display of statistical results. They include:

- "Min (abs/pp)": absolute minimum value of the component.
- "Max (abs /pp)": absolute maximum value of the component.
- "Min (σ)": minimum value of the standard deviation.
- "Max (σ)": maximum value of the standard deviation.
- "Wave count": Number of waveforms (acquisitions) the measurement is based on.

In the "Advanced Jitter Results" table, you can reset the statistics.



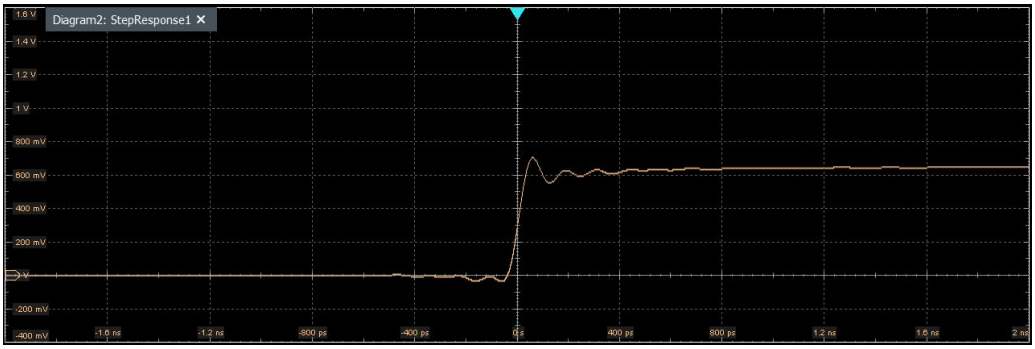

| | σ | abs/pp | Min (σ) | Max (σ) | Min (abs/pp) | Max (abs/pp) | Wave count |
|---|---|---|---|---|---|---|---|
| Event count | --- | 14062 | --- | --- | 14062 | 14062 | 1 |
| RJ + (O)BUJ | 56.652 ps | 270.14 ps | 56.652 ps | 56.652 ps | 270.14 ps | 270.14 ps | 1 |
| TJ | 55.932 ps | 196.07 ps | 55.932 ps | 55.932 ps | 196.07 ps | 196.07 ps | 1 |
| DDJ | 9.916 ps | 64.899 ps | 9.916 ps | 9.916 ps | 64.899 ps | 64.899 ps | 1 |
| RN + (O)BUN | 101.2 mV | 785.15 mV | 101.2 mV | 101.2 mV | 785.15 mV | 785.15 mV | 1 |
| TN | 73.186 mV | 485.62 mV | 73.186 mV | 73.186 mV | 485.62 mV | 485.62 mV | 1 |
| DDN | 94.731 mV | 603.82 mV | 94.731 mV | 94.731 mV | 603.82 mV | 603.82 mV | 1 |

Remote command:
`ADVJitter<m>:RESult:STATistics` on page 2390

**Step response**
Enables a step response diagram. The diagram displays the time behavior of the system when the inputs change from zero to one or from one to zero within a short time period.

You can set the Length and the Step position for the step response diagram in the "Advanced Settings" dialog.

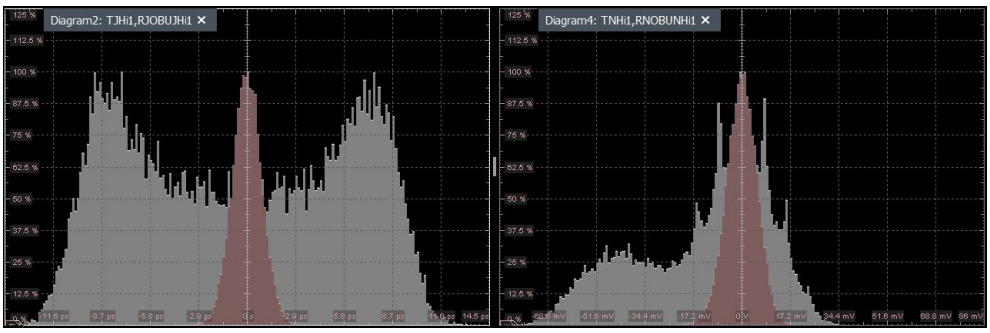
For further analysis of the step response, you can use it as a source for "Amp/Time" measurements. See Chapter 8.2.5, "Amplitude/time measurements", on page 341.



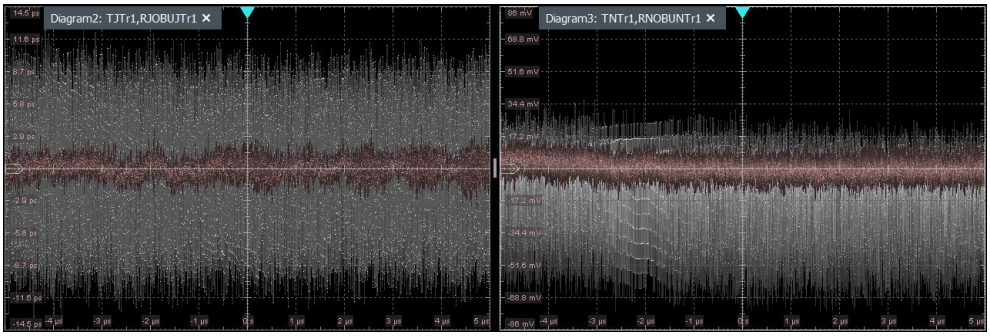

Remote command:
`ADVJitter<m>:RESult:STEPresponse` on page 2390

**Component specific results**
For the selected jitter component, selected you can display:

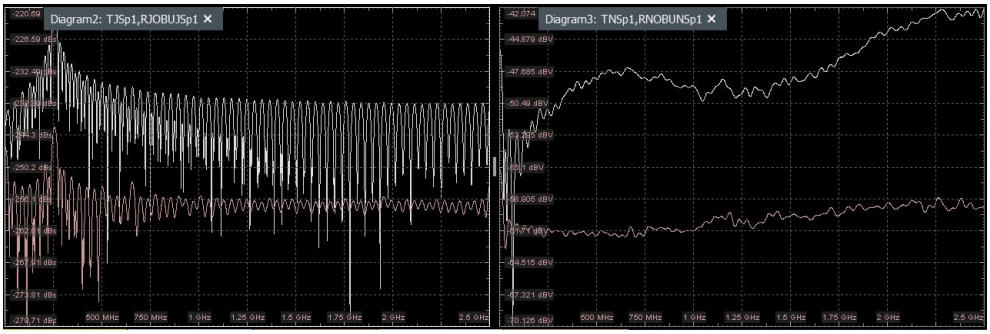
- "Histogram": displays a histogram of measurement results for the selected component - the cumulative occurrence distribution of measurement results in a graphic. You can enable a cursor and use it for further analysis.




- "Track": the measurement results over time for an acquired waveform. It is the graphical interpretation of all measurement values of a single acquisition. Unlike the histogram, it reveals trends of change in the analysis and preserves the timing relationship of the measurement results to the signal.




- "Spectrum": displays how the power of the signal is distributed over the frequency.




Remote command:
`ADVJitter<m>:RESult:COMPonents<n>:HISTogram` on page 2390
`ADVJitter<m>:RESult:COMPonents<n>:TRACk` on page 2391
`ADVJitter<m>:RESult:COMPonents<n>:SPECtrum` on page 2391

**Unit**
In the jitter results dialog, you can select the unit for the displayed results. The following values are available:

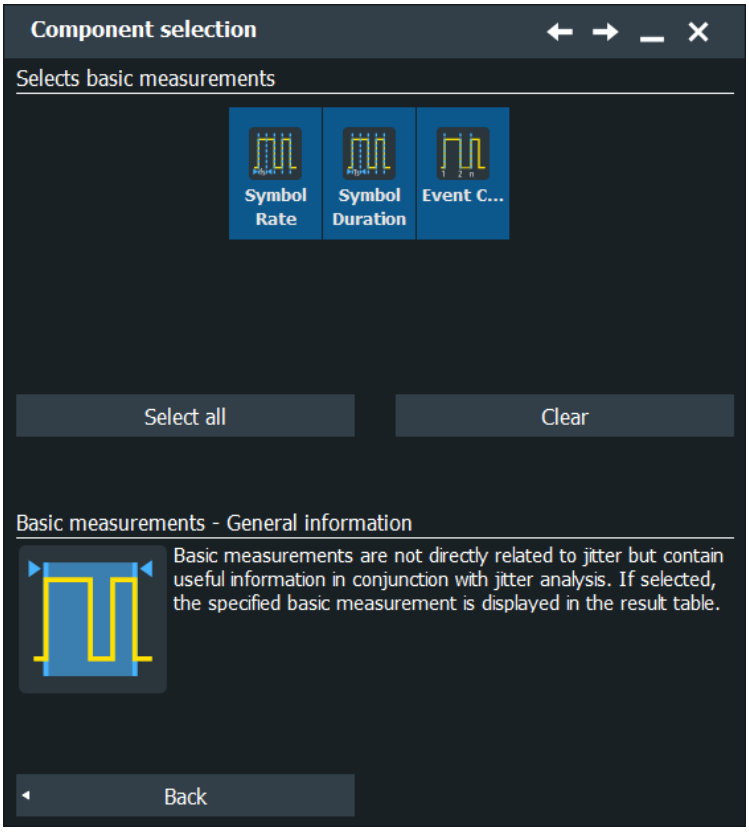
- For jitter measurements, you can select between absolute (seconds) and relative (unit intervals) values.
- For noise measurements, you can select between absolute (V) and relative (%) values.

Remote command:
`ADVJitter<m>:RESult:UNIT` on page 2391

### 18.2.5.2 Basic measurement settings

Access: "Analysis" menu > "Advanced Jitter" > "Result" tab > "Basic".

In the "Basic" tab, you can add measurements for better understanding of the collected data. You can then enable a histogram or track on them, see Component specific results.




**Add / remove components**
Opens a dialog to select the jitter and noise components.

**Basic components**
To display the measurement results, "Enable" the advanced jitter analysis in the "Results" tab.

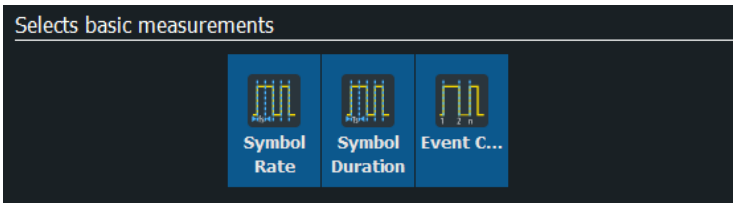


*Table 18-2: Advanced jitter basic measurements*

| Measurement | Abbreviation | Description | StdDev | ap/abs | Histogram | Track |
|---|---|---|---|---|---|---|
| Symbol rate | f_symb | Number of data symbols per second of the analyzed signal. It is equivalent to the baud rate.<br>For binary signals, the signal rate is equal to the bit rate. | | X | X | X |
| Symbol duration | T_symb | The length of time to transfer one signal symbol. Equals the inverse value of the symbol rate. | | X | X | X |
| Event Count | EC | Number of analyzed data transitions. | | X | | |

Remote command:
`ADVJitter<m>:DCOMposition:COMPonents<n>:ENABle` on page 2390

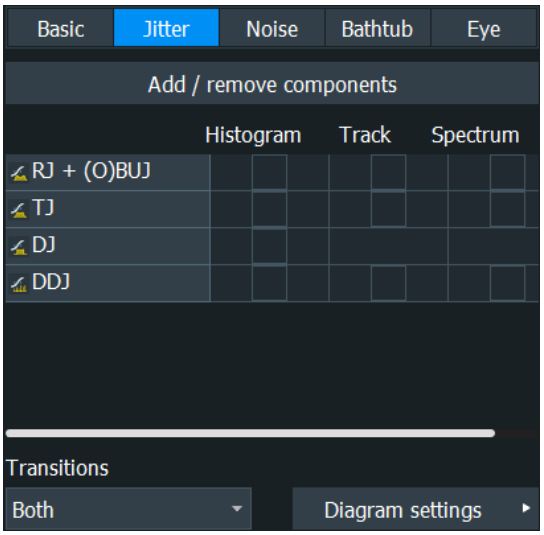
**Select all /Clear**
Selects/clears all components in the basic dialog.

Remote command:
`ADVJitter<m>:DCOMposition:SELBasicmeas` on page 2394
`ADVJitter<m>:DCOMposition:CLEBasicmeas` on page 2394

### 18.2.5.3 Jitter measurement settings

Access: "Analysis" menu > "Advanced Jitter" > "Result" tab > "Jitter".

In the "Jitter" tab, you can add the jitter components you want to measure. You can then enable a histogram, track or spectrum on them, see Component specific results.



**Add / remove components**
Opens a dialog to select the jitter components.

**Jitter Components**
In the "Jitter" tab, you can select which jitter components you want to measure. For the enabled measurements you can display the results in different forms, see Chapter 18.2.5.1, "General settings", on page 1045.

*Table 18-3: Advanced jitter components measurements*

| Measurement | Abbreviation | Description | Results | | Diagram | | |
|---|---|---|---|---|---|---|---|
| | | | StdDev | ap/abs | Histogram | Track | Spectrum |
| Total jitter at BER | TJ@BER | The eye closure at the user-defined target BER. | | x | | | |
| Total jitter | TJ | Measures the total Time Interval Error (TIE) of the input signal.<br>It is defined as the time between each transition of the input signal and its corresponding reference clock transition. | x | x | x | x | x |
| Deterministic jitter | DJ | The reproducible portion of the overall TIE of the input signal.<br>It consists of the Data Dependent Jitter(DDJ), Periodic Jitter (PJ) and the Other Bounded Uncorrelated Jitter (OBUJ) components. | | x | | | |
| Data-dependent jitter | DDJ | Consists of Inter-symbol Interference (ISI) and Duty Cycle Distortion (DCD).<br>It depends on the symbol sequence. ISI is an effect commonly caused by limited signal bandwidth in the transmitter and transmission line, which forces influence from past symbols onto future symbols. DCD is a mismatch between rising and falling signal transitions due to reference level mismatch or other imbalances between transitions. | x | x | x | x | x |
| Duty cycle distortion[3] | DCD | Measure of imbalance between rising and falling data transitions. It is defined as the time difference between the average rising and average falling edge. | | x | | | |
| Random plus (other) bounded uncorrelated jitter | RJ + (O)BUJ | Random jitter plus bounded uncorrelated jitter/ other bounded uncorrelated jitter is the combined effect of both jitter types.<br>If periodic components are activated, RJ + OBUJ is displayed. Otherwise RJ + BUJ.<br>RJ + (O)BUJ is typically used, if track or spectrum plots are desired. Both plots are not available for RJ only or (O)sBUJ only components as the time dependencies of these jitter types are unknown. RJ + (O)BUJ spectrum is a useful approximation of the RJ spectrum for negligible small (O)BUJ. | x | x | x | x | x |

[1] Available, if (Other) bounded uncorrelated > "Enabled".

[2] Available, if Periodic > "Enabled".

[3] No results available, if "Transitions" is set to "Rising"/"Falling".

| Measurement | Abbreviation | Description | Results | | Diagram | | |
|---|---|---|---|---|---|---|---|
| | | | StdDev | ap/abs | Histogram | Track | Spectrum |
| Periodic jitter(2) | PJ | Caused by crosstalk from periodic signal sources to the transmitter or transmission line.<br>It represents the harmonic portion of the overall time interval error (TIE). | x | x | x | x | x |
| Deterministic Dual-Dirac jitter | DJ (δ-δ) | Industry standard approximation to the deterministic jitter measurement, calculated by fitting a model of two overlapping, equally large Gaussian distributions to the observed TJ histogram. The calculated value is the time difference between the two distributions center points. | | x | | | |
| Inter-Symbol interference | ISI | The time between the earliest and the latest signal transition of a single transition type, i.e.rising and falling edges. The measurement result is the largest value across all transition types. | | x | | | |
| Data-dependent and periodic jitter(2) | DDJ + PJ | Consist of the data-dependent and periodic jitter<br>DDJ+PJ is the same as deterministic jitter minus other bounded uncorrelated jitter. In contrast to deterministic jitter, DDJ+PJ offers track and spectrum plots.<br>DDJ+PJ spectrum is a useful approximation of the deterministic jitter spectrum for negligible small other bounded uncorrelated jitter. | x | x | x | x | x |
| Other bounded uncorrelated jitter(1) | (O)BUJ | Composite impact of all unknown but deterministic portions of the overall time interval error of the input signal. It consists of any disturbance that comes from a deterministic source but cannot be reproduced from the input signal. Often it is caused by crosstalk from other data signals. | x | x | x | | |

(1) Available, if (Other) bounded uncorrelated > "Enabled".

(2) Available, if Periodic > "Enabled".

(3) No results available, if "Transitions" is set to "Rising"/"Falling".

| Measurement | Abbreviation | Description | Results | | Diagram | | |
|---|---|---|---|---|---|---|---|
| | | | StdDev | ap/abs | Histogram | Track | Spectrum |
| Other bounded uncorrelated Dual-Dirac jitter(1) | (O)BUJ (δ -δ) | Other bounded uncorrelated Dual-Dirac jitter (O)BUJ (δ- δ) is an industry standard approximation to the other bounded uncorrelated jitter measurement. It is calculated by fitting a model of two overlapping, equally large Gaussian distributions to the observed RJ+(O)BUJ histogram. The calculated value is the amplitude difference between the two distributions center points. | | x | | | |
| Random jitter (1) | RJ | Consists of any influence that cannot be predicted with model parameters. Usually, this jitter component follows a Gaussian probability distribution. The algorithm computes the standard deviation and the peak-peak value of the TIE caused by RJ within the analyzed waveform. | x | | x | | |

(1) Available, if (Other) bounded uncorrelated > "Enabled".

(2) Available, if Periodic > "Enabled".

(3) No results available, if "Transitions" is set to "Rising"/"Falling".

To display the measurement results, "Enable" the advanced jitter analysis in the "Results" tab.

Remote command:
`ADVJitter<m>:DCOMposition:COMPonents<n>:ENABle` on page 2390

**Select all /Clear**
Selects/clears all components in the jitter dialog.

Remote command:
`ADVJitter<m>:DCOMposition:CLEJitcomp` on page 2394
`ADVJitter<m>:DCOMposition:SELJitcomp` on page 2394

**Transitions**
Selects which transitions are considered for the display of the diagrams, rising, falling or both. This setting is mostly relevant for histograms, tracks and the measurement results.

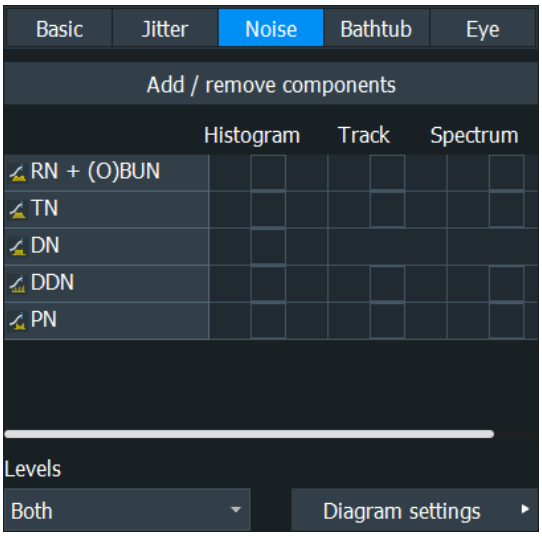
Remote command:
`ADVJitter<m>:RESult:TRANsitions` on page 2395

### 18.2.5.4 Noise measurement settings

Available for option R&S RTP- K134.

Access: "Analysis" menu > "Advanced Jitter" > "Result" tab > "Noise".

In the "Noise" tab, you can add the noise components you want to measure. You can then enable a histogram, track or spectrum on them, see Component specific results.




**Add / remove components**
Opens a dialog to select the noise components.

**Noise components**
In the "Noise" tab, you can select which jitter noise components you want to measure.

Component selection
Select noise components
EH@BER
RN + (O)BUN
TN
DN
DDN
LD
PN
ISIN
DDN + PN
(O)BUN
(O)BUN (δ-δ)
RN
Select all
Clear
Enable (Other) bounded uncorrelated in tab "Advanced"
Noise - General information
Noise describes the amplitude deviation of a data symbol at the sampling time with respect to a corresponding reference level. These deviations or Level Errors (LE) are caused by various sources.
The Total Noise (TN) is split into two main components: Random Noise (RN) and Deterministic Noise (DN). DN consists of Periodic Noise (PN) and non periodic components.
Back

| Measurement | Abbreviation | Description | Results | | Diagram | | |
|---|---|---|---|---|---|---|---|
| | | | StDev | ap/abs | Histogram | Track | Spectrum |
| Eye height @BER[3] | EH@BER | Returns the vertical eye opening at the user-defined BER. | | x | | | |
| Random plus (other) bounded uncorrelated noise | RN+ (O)BUN | Random noise plus (other) bounded uncorrelated noise or (other) bounded uncorrelated noise is the combined effect of both noise types.<br>If periodic components are activated RN + OBUN are displayed, otherwise RN + BUN.<br>RN + (O)BUN is typically used, if track or spectrum plots are desired. Both plots are not available for RN only or (O)BUN only components as the time dependencies of these noise types are unknown. RN + (O)BUN spectrum is a useful approximation of the RN spectrum for negligible small (O)BUN. | x | x | x | x | x |
| Total noise | TN | Measures the total level error of the input signal. It is defined as the amplitude deviation between the amplitude of the input signal at the decision time point and its corresponding reference level. | x | x | x | x | x |
| Deterministic noise | DN | The reproducible portion of the overall level error of the input signal. It consists of the data-dependent noise, the periodic noise components and the other bounded uncorrelated jitter components. | x | x | x | | |
| Data-dependent noise | DDN | Consists of inter-symbol interface noise and level distortion. It depends on the symbol sequence. ISIN is an effect whereby past symbols have an influence on future symbols.<br>DCD is a mismatch between rising and falling signal transitions due to reference level mismatch or other imbalances between transitions. | x | x | x | x | x |
| Level distortion[3] | LD | A measure of imbalance between different symbol levels. It is defined as the difference between the average level errors of the symbols one and zero. | | x | | | |
| Periodic noise[2] | PN | Periodic noise is typically caused by crosstalk from periodic signal sources to the transmitter or transmission line. It represents the harmonic portion of the overall level error. | x | x | x | x | x |

[1] Available, if (Other) bounded uncorrelated > "Enabled".

[2] Available, if Periodic > "Enabled".

[3] No results available, if "Levels" is set to "Rising"/"Falling".

| Measurement | Abbreviation | Description | Results | | Diagram | | |
|---|---|---|---|---|---|---|---|
| | | | StDev | ap/abs | Histogram | Track | Spectrum |
| Inter-symbol interference noise | ISIN | The level difference between the highest and the lowest level error of a symbol, i.e. one or zero. The largest value across all symbols is the measurement result. | | x | | | |
| Data-dependent noise and periodic noise[2] | DDN+PN | Data-dependent noise plus periodic noise is the combined effect of both jitter types.<br>Note that DDN+PN equals deterministic noise minus other bounded uncorrelated noise.<br>DDN+PN spectrum is a useful approximation of the deterministic noise spectrum for negligible small other bounded uncorrelated noise. | x | x | x | x | x |
| Other bounded uncorrelated noise [1] | (O)BUN | Composite impact of all unknown but deterministic portions of the overall time interval error of the input signal. It consists of any disturbance that comes from a deterministic source but cannot be reproduced from the input signal. Often it is caused by crosstalk from other data signals. | x | x | x | | |
| Other bounded uncorrelated Dual-Dirac noise [1] | (O)BUN (δ-δ) | Other bounded uncorrelated Dual-Dirac noise (O)BUN (δ- δ) is an industry standard approximation to the other bounded uncorrelated jitter measurement. It is calculated by fitting a model of two overlapping, equally large Gaussian distributions to the observed RN+(O)BUN histogram. The calculated value is the amplitude difference between the two distributions center points. | | x | | | |
| Random noise [1] | RN | Consists of any influence that cannot be predicted with model parameters. Usually, this noise component follows a Gaussian probability distribution. The algorithm computes the standard deviation and the peak-peak value of the level error (LE) caused by RN within the analyzed waveform. | x | | x | | |

[1] Available, if (Other) bounded uncorrelated > "Enabled".

[2] Available, if Periodic > "Enabled".

[3] No results available, if "Levels" is set to "Rising"/"Falling".

Remote command:
`ADVJitter<m>:DCOMposition:COMPonents<n>:ENABle` on page 2390

**Select all /Clear**
Selects/clears all components in the noise dialog.

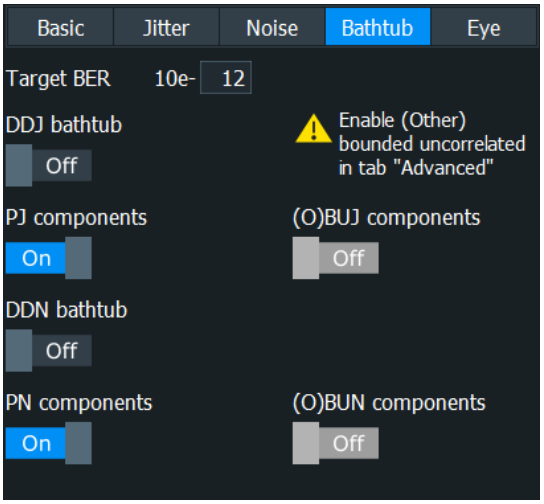
Remote command:
ADVJitter<m>:DCOMposition:CLENoisecomp on page 2395
ADVJitter<m>:DCOMposition:SELNoisecomp on page 2395

**Levels**
Selects which levels are considered for the display of the diagrams, rising, falling or both. This setting is mostly relevant for histograms, tracks and the measurement results.

Remote command:
ADVJitter<m>:RESult:LEVels on page 2395

### 18.2.5.5 Bathtub settings

Access: "Analysis" menu > "Advanced Jitter" > "Result" tab > "Bathtub".




**Target BER 10**
Sets the target bit error rate.

Remote command:
ADVJitter<m>:DCOMposition:TARBer on page 2396

**DDJ bathtub**
Enables a bathtub plot, a graph of BER in a log scale versus data-dependent jitter (DDJ).

Remote command:
ADVJitter<m>:RESult:BATHtub on page 2396

**PJ / (O)BUJ components**
Selects which jitter components are considered for the jitter bathtub.

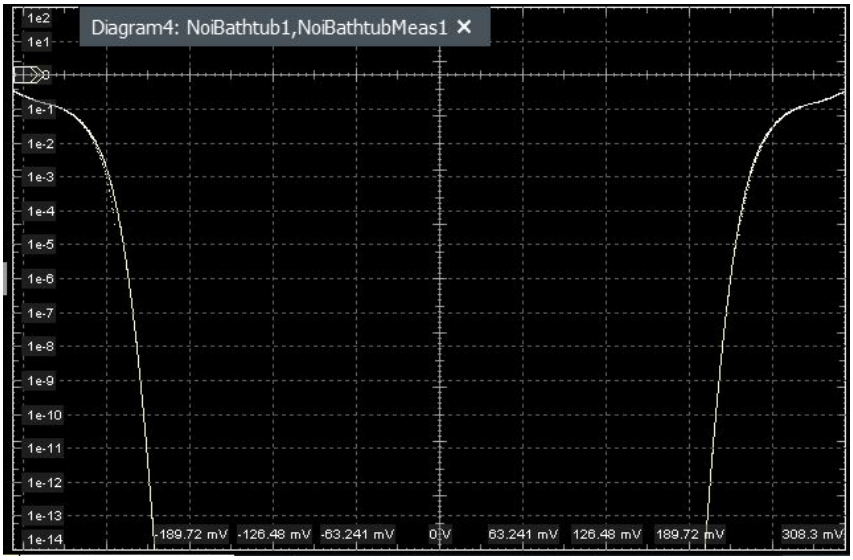
"PJ" is available, if Periodic > "Enabled".

"(O)BUJ"is available, if (Other) bounded uncorrelated > "Enabled".

Remote command:
ADVJitter<m>:RESult:PJ on page 2397
ADVJitter<m>:RESult:OBUJ on page 2397

**DDN bathtub**
Available for option R&S RTP- K134.

Enables a noise bathtub plot, a graph of BER in a log scale versus data-dependent jitter (DDN).




Remote command:
ADVJitter<m>:RESult:NBAThtub on page 2397

**PN / (O)BUN components**
Selects which noise components are considered for the noise bathtub.

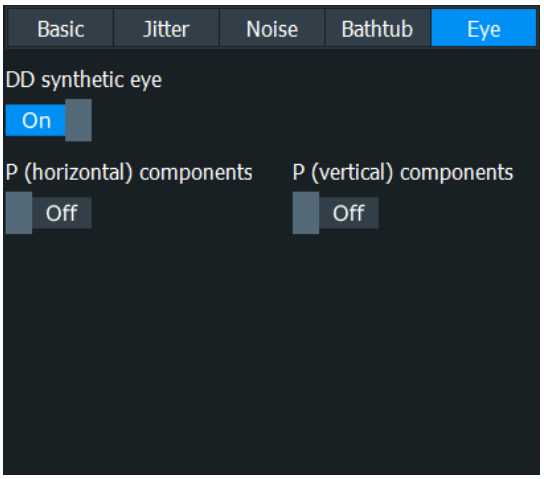
"PN" is available, if Periodic > "Enabled".

"(O)BUN" is available, if (Other) bounded uncorrelated > "Enabled".

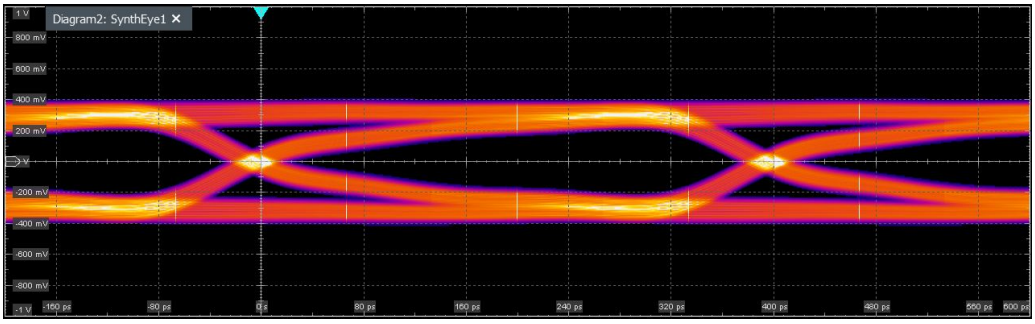
Remote command:
ADVJitter<m>:RESult:PN on page 2397
ADVJitter<m>:RESult:OBUN on page 2398

### 18.2.5.6 Eye settings

Access: "Analysis" menu > "Advanced Jitter" > "Result" tab > "Eye".



**Synthetic eye**
Enables a synthetic eye plot out of the synthesized data.




Remote command:
ADVJitter<m>:RESult:SYNTheticeye on page 2398

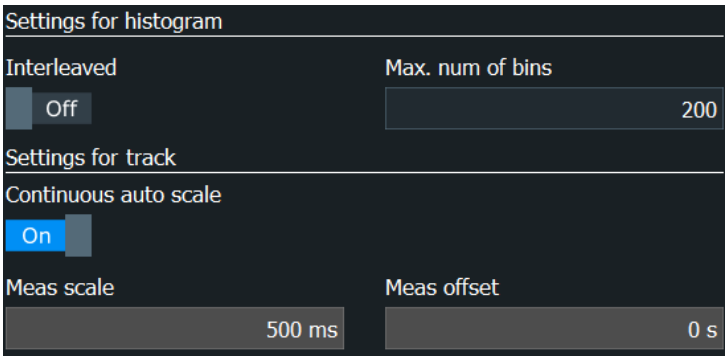
**Add periodic components to DD synthetic eye**
Adds horizontal/vertical periodic components to the data-dependent synthetic eye.

Remote command:
ADVJitter<m>:RESult:PHORizontal on page 2398
ADVJitter<m>:RESult:PVERtical on page 2398

### 18.2.5.7 Diagram settings

In the diagram settings dialog you can set a separate scale for the basic, jitter and noise parameters.

**Settings for histogram and track**
Defines the settings for the histogram and track diagrams.




**Interleaved ← Settings for histogram and track**
Displays interleaved histogram values.

Remote command:
ADVJitter<m>:DCOMposition:JINTerleaved on page 2403

**Max. numb of bins ← Settings for histogram and track**
Sets the number of bins, the number of data intervals, of the histogram.

Remote command:
ADVJitter<m>:RESult:VERTical:HBINs on page 2399

**Continuous auto scale ← Settings for histogram and track**
If enabled, the measurement scale for track is set automatically. If disabled you can manually set a Meas scale and Meas offset.

Remote command:
ADVJitter<m>:RESult:VERTical:CONTinuous on page 2400
ADVJitter<m>:RESult:VERTical:BHContinuous on page 2400
ADVJitter<m>:RESult:VERTical:NHContinuous on page 2401

**Measurement type ← Settings for histogram and track**
Selects the basic measurement type to apply the scale to.

Remote command:
ADVJitter<m>:RESult:VERTical:BMTYpe on page 2400

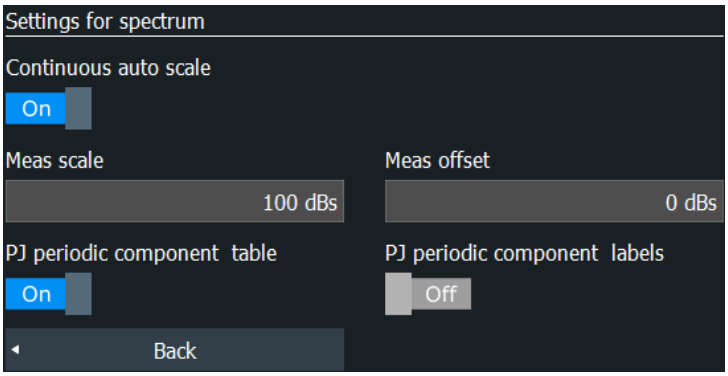
**Meas scale ← Settings for histogram and track**
Sets a measurement scale. The unit depends on the measurement type.

Remote command:
ADVJitter<m>:RESult:COMPonents<n>:SCALe on page 2401

**Meas offset ← Settings for histogram and track**
Sets a measurement offset. The unit depends on the measurement type.

Remote command:
ADVJitter<m>:RESult:COMPonents<n>:OFFSet on page 2402

**Settings for spectrum**
Defines the settings for the histogram and track diagrams.

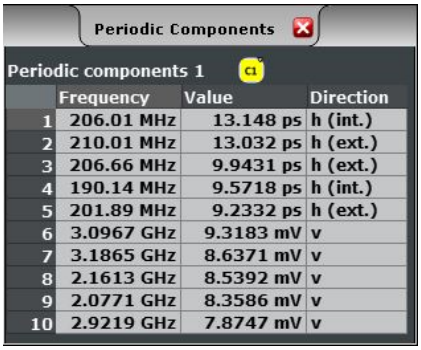



**Continuous auto scale ← Settings for spectrum**
If enabled, the measurement scale for the spectrum is set automatically. If disabled you can manually set a Meas scale and Meas offset.

Remote command:
ADVJitter<m>:RESult:VERTical:JSContinuous on page 2400
ADVJitter<m>:RESult:VERTical:NSContinuous on page 2401

**PJ periodic component table ← Settings for spectrum**
Enables the display of the periodic jitter components table. The table is displayed after the advanced jitter/ noise analysis is enabled.

The unit of the periodic components depends on the switching between "Absolute" and "Relative". Thus, times are represented in “s” and “UI” and voltages in “V” and “%”.




| | Frequency | Value | Direction |
|---|---|---|---|
| 1 | 206.01 MHz | 13.148 ps | h (int.) |
| 2 | 210.01 MHz | 13.032 ps | h (ext.) |
| 3 | 206.66 MHz | 9.9431 ps | h (ext.) |
| 4 | 190.14 MHz | 9.5718 ps | h (int.) |
| 5 | 201.89 MHz | 9.2332 ps | h (ext.) |
| 6 | 3.0967 GHz | 9.3183 mV | v |
| 7 | 3.1865 GHz | 8.6371 mV | v |
| 8 | 2.1613 GHz | 8.5392 mV | v |
| 9 | 2.0771 GHz | 8.3586 mV | v |
| 10 | 2.9219 GHz | 7.8747 mV | v |

Remote command:
`ADVJitter<m>:DCOMposition:SPJTable` on page 2402

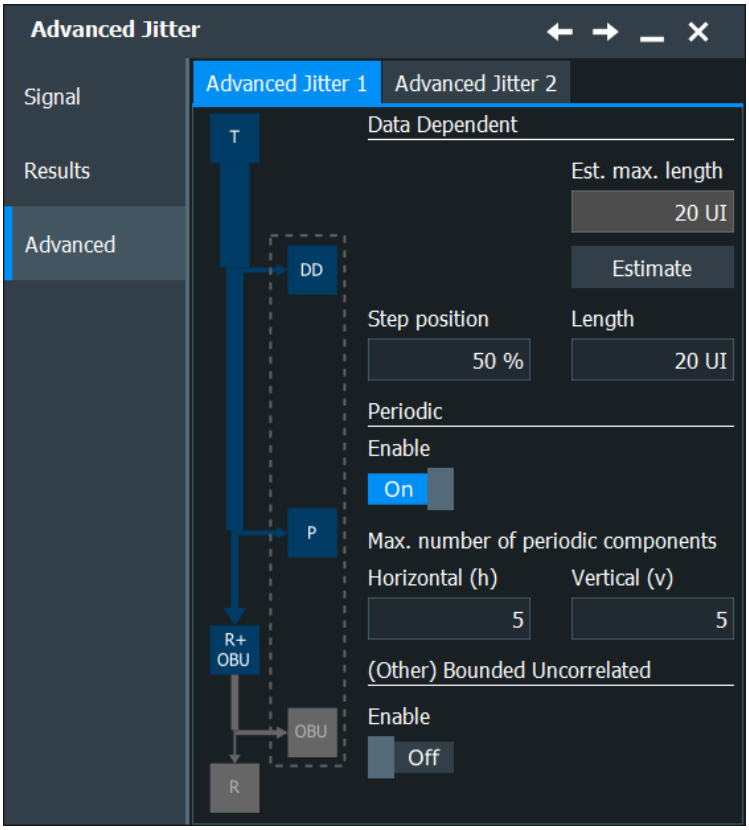
**PJ periodic component labels ← Settings for spectrum**
Displays the periodic jitter/noise labels on the spectrum diagrams.

Available, if in the "Result"> "Jitter" tab, "Spectrum" is enabled for the "Periodic Jitter" measurement.

Remote command:
`ADVJitter<m>:DCOMposition:SPJLabels` on page 2403
`ADVJitter<m>:DCOMposition:SPNLabels` on page 2403

## 18.2.6 Advanced settings

Access: "Analysis" menu > "Advanced Jitter" > "Advanced settings" tab.




**Length**
Sets the step response length in unit intervals. A higher step response value leads to longer measurement times. Hence it is best to set this value to the number of unit intervals that is roughly needed until the ringing has subsided.

You can set the step response manually or run an estimate.

The "Estimate" sets the maximum feasible value. If it is lower than the set "Length", then the "Length" value is overwritten.

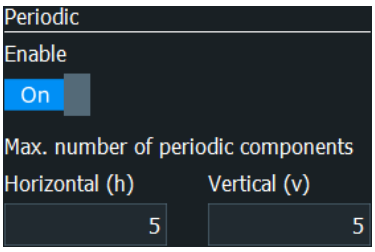
Remote command:
`ADVJitter<m>:DCOMposition:RESPlength` on page 2404
`ADVJitter<m>:DCOMposition:RLEStimate` on page 2404
`ADVJitter<m>:DCOMposition:MRESplength?` on page 2404

**Step position**
Sets the steps position. It defines what percentage of the unit lengths are displayed before the step response.

For example, if the "Length" is set to 20 UI and the "Step position" is set to 75%, the step response diagram displays 15 UI before the step response and 5 UI after it.

Remote command:
`ADVJitter<m>:DCOMposition:POSStep` on page 2404

**Periodic**
If enabled, allows the selection of the "Periodic jitter" / "Periodic noise" components: PJ, DDJ+PJ, PN and DDN+PN.




**Enable ← Periodic**
Enables the periodic components.

Remote command:
ADVJitter<m>:DCOMposition:ENP on page 2405

**Horizonta (h) direction ← Periodic**
Sets the maximum number of periodic components in the horizontal direction that are considered for the measurement. If the actual number of periodic components is higher than the set value, they are considered for the random jitter component.

The horizontal components differentiate also between intrinsic and extrinsic. Extrinsic periodic horizontal components describe timing variations likely caused by external interference on the clock generation system of the DUT. The interference source is free-running and not coupled to the generated clock. Intrinsic periodic horizontal components describe timing variations likely caused by fluctuations within the clock generation system of the DUT. The interference is tightly coupled to the generated clock and follows modulations in data rate, e.g. spread spectrum clocking.

Remote command:
ADVJitter<m>:DCOMposition:NHORizontal on page 2405

**Vertical (v) direction ← Periodic**
Sets the maximum number of periodic components in the vertical direction that are considered for the measurement. If the actual number of periodic components is higher than the set value, they are considered for the random jitter component.

Remote command:
ADVJitter<m>:DCOMposition:NVERtical on page 2405

**(Other) bounded uncorrelated**
If enabled, allows the selection of the (other) bounded uncorrelated jitter/noise: (O)BUJ, (O)BUJ(δ -δ), RJ, (O)BUN, (O)BUN(δ -δ), RN.

The **other** bounded uncorrelated jitter/noise includes the periodic jitter/noise components. It is displayed, if Periodic is enabled. Otherwise the bounded uncorrelated jitter/noise is displayed.

Remote command:
ADVJitter<m>:DCOMposition:ENOBu on page 2406

### 18.2.7 Saving the results

Once the advanced jitter analysis is enabled, you can save the results as a report or export the waveforms for future references.

#### Report

You can save a report of the advanced jitter analysis, which displays the "Adv. Jitter Results" table and the "Periodic components" table.

Advanced Jitter: Results of all Components

| | StdDev | abs/pp | Min(StdDev) | Max(StdDev) | Min(abs/pp) | Max(abs/pp) | Wave Count |
|---|---|---|---|---|---|---|---|
| AdvancedJitterResults1 | Analyzed source: Ch1Wfm1 | | Reference source: CDR | Use gate: Off | | | |
| Event Count | --- | 14.2 k | --- | --- | 14.2 k | 14.2 k | 1 |
| Random Jitter and Other Bounded Uncorrelated Jitter | 54.9 ps | 237 ps | 54.9 ps | 54.9 ps | 237 ps | 237 ps | 1 |
| Total Jitter (measured) | 55.5 ps | 196 ps | 55.5 ps | 55.5 ps | 196 ps | 196 ps | 1 |
| Data-Dependent Jitter | 5.53 ps | 53.5 ps | 5.53 ps | 5.53 ps | 53.5 ps | 53.5 ps | 1 |
| Periodic Jitter | 8.4 ps | 88.1 ps | 8.4 ps | 8.4 ps | 88.1 ps | 88.1 ps | 1 |
| Random Noise and Other Bounded Uncorrelated Noise | 78.9 mV | 605 mV | 78.9 mV | 78.9 mV | 605 mV | 605 mV | 1 |
| Total Noise | 72.9 mV | 486 mV | 72.9 mV | 72.9 mV | 486 mV | 486 mV | 1 |
| Data-Dependent Noise | 61.8 mV | 372 mV | 61.8 mV | 61.8 mV | 372 mV | 372 mV | 1 |
| Periodic Noise | 18.8 mV | 301 mV | 18.8 mV | 18.8 mV | 301 mV | 301 mV | 1 |

Advanced Jitter: Periodic Components of Periodic Jitter

| | Frequency | Value | Direction |
|---|---|---|---|
| AdvancedJitterPeriodicComponents1 | | Analyzed source: Ch1Wfm1 | |
| 1 | 206 MHz | 13.1 ps | Horizontal (intrinsic) |
| 2 | 210 MHz | 13 ps | Horizontal (extrinsic) |
| 3 | 207 MHz | 9.94 ps | Horizontal (extrinsic) |
| 4 | 190 MHz | 9.57 ps | Horizontal (intrinsic) |
| 5 | 202 MHz | 9.23 ps | Horizontal (extrinsic) |
| 6 | 3.1 GHz | 9.32 mV | Vertical |
| 7 | 3.19 GHz | 8.64 mV | Vertical |
| 8 | 2.16 GHz | 8.54 mV | Vertical |
| 9 | 2.08 GHz | 8.36 mV | Vertical |
| 10 | 2.92 GHz | 7.87 mV | Vertical |

See also Chapter 12.4, "Reports", on page 511.

#### Exporting the output signal

The following output signals from the advanced jitter analysis can be exported in the file formats `*.csv`, `*.bin`and `*.xml`:

- Step response
- Bathtub
- Histograms
- Tracks
- Spectrum

## 18.3 Clock data recovery

The process of clock data recovery (CDR) generates a reference clock from a high-speed serial data stream that is sent without a dedicated clock signal. The generated clock signal matches the frequency and is aligned to the phase of the data stream. The recovered clock can be used to sample the data stream, to obtain the sequence of transmitted bits, and to measure jitter parameters.

There are two ways to recover the clock signal:

- Using software algorithms that calculate the clock from data signal edges (option R&S RTP-K12):
  Chapter 18.3.1, "Software CDR", on page 1067
- Using hardware CDR
  See Chapter 6.3.16, "Hardware CDR trigger", on page 220

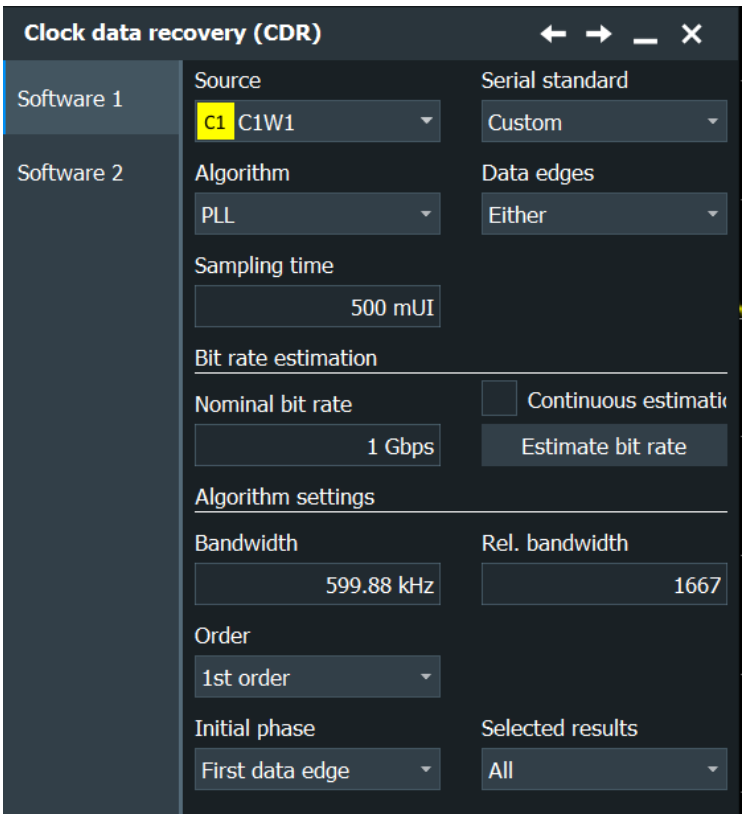
### 18.3.1 Software CDR

You can define two independent software CDR instances to recover clock signals. These clock signals are used for data timing measurements: time interval error, unit interval, and data rate (frequency of the clock signal). The lower the signal bandwidth and the higher the relative bandwidth, the longer is the settling time.

Software CDR is based on acquired data, needs a synchronization time to set up the clock. To get correct measurement results, the data acquired during synchronization time has to be discarded.

You can also display the recovered clock signal as a math waveform, see Chapter 18.3.2, "Displaying the recovered clock signal", on page 1070.

Access: [Apps] > "Analysis" tab > "CDR" > "Software1"/"Software2"




**Source**
Selects the source of the data stream.

Remote command:
`CDR:SOFTware<m>:ESBRate:SOURce` on page 2407

**Serial standard**
For some serial standards, the bit rate and bandwidth are known to the instrument. Select the standard, and the "Nominal bit rate" and bandwidth settings are set automatically, no estimation is required.

Remote command:
`CDR:SOFTware<m>:SERStandard` on page 2407

**Data edges**
Selects the edges of the data stream that are used for the clock recovery.

- "Either": Both positive and negative edges are used
- "Positive / Negative": Only one edge direction is used. Use one of these settings if the other edge might deliver unreliable results.

Remote command:
`CDR:SOFTware<m>:ESLope` on page 2409

**Algorithm**
Sets the software algorithm that is used for clock data recovery.

"Constant freq." CDR uses the nominal bit rate to generate the clock signal. The method assumes that the frequency of the signal is constant during the complete acquisition.

"PLL" PLL is the phase-locked loop control system. It can follow slow deviations in the frequency of the data stream. Thus, it acts as highpass filter regarding the jitter that remains on the signal.

"Feed Forward" The feed forward algorithm is a mathematical optimization method used for clock data recovery. Feed forward is available if none of the listed serial standards is used ("Serial standard" = "Custom"). It is faster than PLL, i.e., the settling time is shorter. Feed forward is also more precise than PLL. To set up the algorithm, set the bandwidth and the relative bandwidth.

Remote command:
`CDR:SOFTware<m>:ALGorithm` on page 2406

**Sampling time**
Sets a sampling time for the explicit clock signal, an offset for the clock edge in relation to the bit start. The clock edge sets the beginning of the unit interval. The sampling time is a number between 0 and 1. Value 0 sets the clock edge to the beginning of the bit period; value 0.5 sets the clock edge to the middle of the bit period.

Remote command:
`CDR:SOFTware<m>:SAMTime` on page 2412

**Bit rate estimation**
Bit rate estimation analyzes the source signal based on the given "Nominal bit rate" and corrects the value.

"Continuous estimation" Enables ongoing bit rate correction.

"Estimate" Estimates the bit rate once, for the current acquisition.

Remote command:
`CDR:SOFTware<m>:ESBRate:CONTinuous` on page 2408
`CDR:SOFTware<m>:ESBRate:BREStimate` on page 2408

**Nominal bit rate**
Sets the quiescent frequency of the PLL. It corresponds to the data rate of the data stream from which the clock is to be recovered.

Remote command:
`CDR:SOFTware<m>:BITRate` on page 2408

**Algorithm settings**
Phase-locked loop parameters are listed below.

**Note**: Nomial bit rate, bandwidth and relative bandwidth are interacting settings. Modifying one parameter also changes one of the dependent parameters.

"Order" Sets the order of the PLL: first or second order. PLL of higher order can compensate for more complex jitter behavior.

"Damping" Sets the damping factor, which is only relevant for second order PLL.

"Bandwidth" Sets the PLL bandwidth. It defines the part of the spectrum that the PLL can follow during synchronization. The PLL bandwidth is usually defined by the transmission standard.

"Rel. bandwidth" Sets the relative bandwidth, that is the ratio of the nominal bit rate to the PLL bandwidth.

Remote command:
`CDR:SOFTware<m>:PLL:ORDer` on page 2410
`CDR:SOFTware<m>:PLL:BWIDth` on page 2410
`CDR:SOFTware<m>:PLL:RELBwidth` on page 2411
`CDR:SOFTware<m>:PLL:DAMPing` on page 2411

**Reverse lock**
Available for "Algorithm" > "PLL".

Enables the reverse lock PLL algorithm. The signal is locked on the first edge.

Remote command:
`CDR:SOFTware<m>:REVLockpll` on page 2411

**Initial phase**
Available for "Algorithm" > "PLL".

Defines the phase reference for the first clock edge.

"First sample" The first clock edge matches the first sample of the waveform at the left border of the display.

"First data edge" The first clock edge matches the first edge of the data signal.

Remote command:
`CDR:SOFTware<m>:SYNC` on page 2409

**Selected results**
Available for "Algorithm" > "PLL".

The PLL requires some time to synchronize to the phase of the data stream. You can select when the CDR algorithm returns clock edges:

"After initial sync." The clock edges of the synchronization time are discarded; results are gathered after initial synchronization of the CDR. Thus, meaningful TIE measurement results can be obtained.

"All" All clock edges are used.

Remote command:
`CDR:SOFTware<m>:RESults` on page 2410

### 18.3.2 Displaying the recovered clock signal

The clock signal that is recovered by CDR can be displayed as a math waveform. The generated waveform is time-correlated to the data waveform.

1. Prerequisite: Configure the CDR method.

2. Press the [Math] key.
3. On the "Setup" tab, select the "Equation" subtab.
4. Double-tap the entry field to open the formula editor.
5. Tap "More", and again "More".
6. Tap "Clear".
7. Tap "CDR" and select the CDR method.
8. Complete the expression:
   SW CDR: e.g., CDR(sw1,Ch1Wfm1)
9. Tap "Enter".
10. Enable the math signal.

# 19 Advanced eye diagram analysis (option R&S RTP-K136/K137)

The main application of eye diagrams is the analysis of signal integrity and signal quality. In particular, eye diagram analysis is used for serial communication standards, for both high-speed serial protocols (e.g. USB) and low-speed serial protocols (e.g. CAN, FlexRay).

There are two ways to generate eye diagrams on the R&S RTP:

- Standard eye diagrams, which are a superposition of repetitively sampled digital data.
- To use advanced eye diagrams, option R&S RTP-K136/K137 or R&S RTP-K91/K93 is required.

## 19.1 Calculation and display of advanced eye diagrams

The R&S RTP generates advanced eye diagrams. To build advanced eyes, the oscilloscope acquires a long waveform containing many bits. The instrument uses a clock reference to cut the waveform into slices of 1-bit length, and superimposes the individual slices. The superimposed slices of consecutive bits build the advanced eye. The clock reference defines the position of the individual data bits on the time axis, and so it provides the necessary timing marks to align and superimpose the waveform slices.

On the contrary, the basic R&S RTP without the advanced eye option builds the eye from consecutive acquisitions. The trigger is set in a way that each acquisition is about 1 bit long. With long-persistence display settings, the consecutive acquisitions draw an eye on the screen. The bits that contribute to the eye are always non-consecutive due to the unavoidable oscilloscope's blind time between acquisitions.

The advanced eye approach combines consecutive bits in the eye diagram and does not miss any bit of the acquisition. Advanced eye generation also offers time control over the waveform slices contributing to the eye diagram. For example, you can select the bits contributing to the eye by filtering on dedicated data sequences or by excluding idle sequences.

You can define up to four advanced eye diagrams and configure them independently. Each eye diagram is shown in a separate display diagram, which has its own signal icon. You can move and minimize eye diagrams like any waveform diagram and analyze all eye diagrams in parallel.

The following analysis functions are available for advanced eye diagrams:

- Cursor measurements
- Automatic eye measurements
- Histograms
- Zoom
- Mask testing

## 19.2 Eye diagram setup

This chapter describes the configuration of eye diagrams.

The configuration of an eye diagram mainly consists of the selection and setup of the input source and timing reference source. You can also adjust display settings and restrict the segments that contribute to the eye generation by various means.

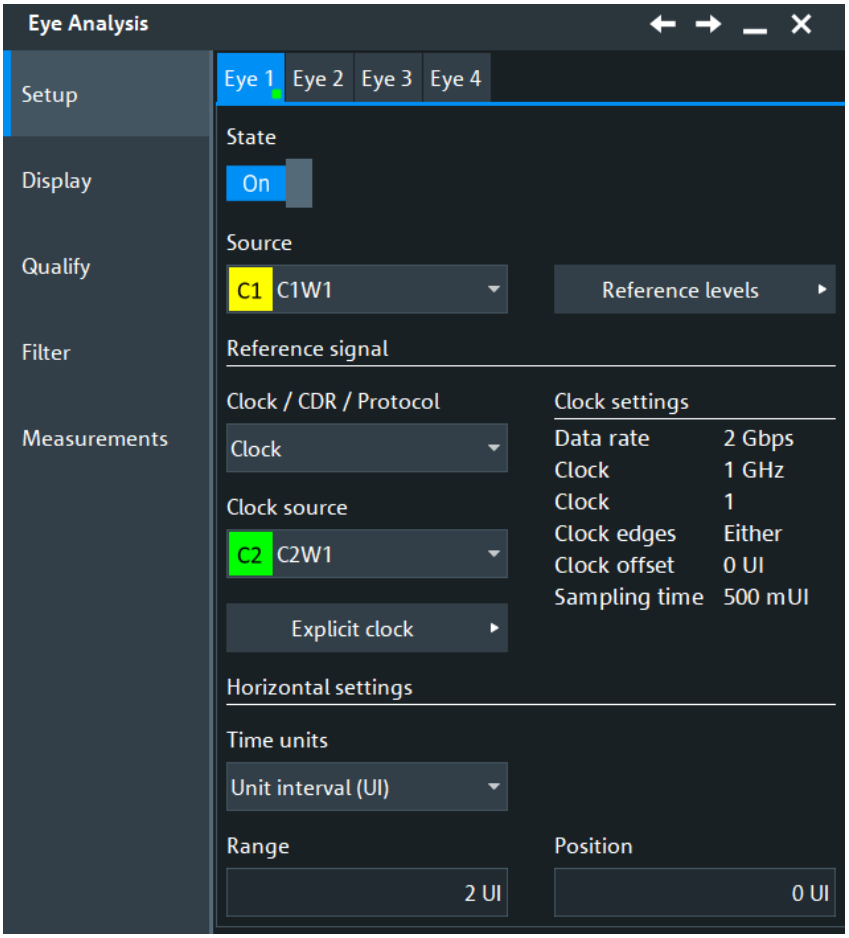
You can configure and analyze up to four independent eye diagrams in parallel.

The "Eye Diagram" dialog box provides four tabs with all settings that are required to configure eye diagrams.

### 19.2.1 Basic setup

Access: [Apps]> "Analysis" > "Eye Analysis" > "Setup" tab

In the "Setup" tab, you configure the source, the timing reference, and horizontal scale settings of the eye diagram. The vertical scale of the eye diagram is coupled to the vertical scale of the data source.



You can configure and analyze up to four independent eye diagrams in parallel (vertical subtabs Eye 1 to Eye 4). Make sure to select the correct eye diagram subtab on the left before you adjust the settings.

State....1074
Source....1074
Reference signal....1074
└ Clock / CDR / Protocol....1074
└ Clock....1075
└ CDR....1075
└ Protocol....1075
Horizontal settings....1076
└ Time units....1076
└ Range, Position....1076
└ Range UI, Position UI....1076
Explicit clock....1077
└ Vertical reference level....1077
└ Clock frequency....1077
└ Estimate clock frequency....1078
└ Clock multiplier....1078
└ Clock edge....1078
└ Data rate....1078
└ Clock offset....1078
└ Estimate clock offset....1078
└ Sampling time....1078

**State**
Activates or deactivates the eye diagram.

Remote command:
`EYE<m>[:STATe]` on page 2413

**Source**
Selects the waveform from which the eye diagram is generated (data source). You can select analog channels, reference and math waveforms.

If the source waveform is minimized or hidden, no measurement results for the eye diagram are shown.

Select the same channel for the DDR serial bus protocol configuration "DQ" data line and the input for the eye diagram.

Remote command:
`EYE<m>:SOURce` on page 2413

**Reference signal**
In this section you can define the reference signal used for the eye diagram.

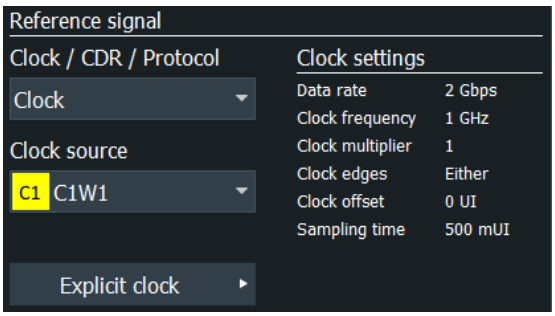
**Clock / CDR / Protocol ← Reference signal**
Selects the type of waveform that is used to obtain the timing information required to slice the data source waveform.

Remote command:
`EYE<m>:TIMReference[:SOURce]` on page 2418

**Clock ← Reference signal**
If a clock signal is available, the instrument can use it directly to set the segmentation timestamps. The signal has to be provided on an additional channel.

With "Clock source" you can select the source of the clock signal.

See also Explicit clock.

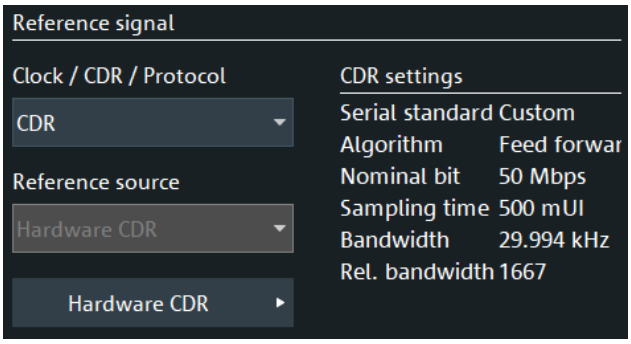



Remote command:
EYE<m>:TIMReference:CLOCk:SOURce on page 2418

**CDR ← Reference signal**
Available, if Type = "CDR".

Uses a hardware clock data recovery. You can also load a reference waveform that contains CDR information in the "Menu" > "Save/Recall" > "Recall" > "Waveform" dialog.

See Hardware CDR trigger.

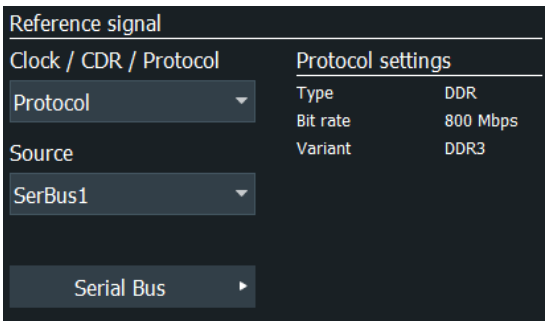



Remote command:
REFCurve<m>:CDR:ALGorithm? on page 1398
REFCurve<m>:CDR:BITRate? on page 1398
REFCurve<m>:CDR:BWIDth? on page 1399
REFCurve<m>:CDR:RELBwidth? on page 1399
REFCurve<m>:CDR:SAMTime? on page 1399

**Protocol ← Reference signal**
If "Protocol" is selected as the reference type signal, the DDR source signal settings are used for the timing information.

You can select the "Source" of the protocol signal. You can only select the protocols that are enabled.

Also some protocol-specific settings are displayed: "Type", "Bit rate", "Variant".




Remote command:
EYE<m>:TIMReference:PROTocol:SELect on page 2418

**Horizontal settings**
Defines the horizontal settings.

**Time units ← Horizontal settings**
Sets the method to define the horizontal scale of the eye diagram: as absolute time, or in unit intervals. You can use unit interval, if the nominal frequency of the clock signal is known.

Indirectly, the horizontal scale determines the size of one waveform slice. The waveform slice must be fully contained in the waveform acquisition to contribute to the eye diagram.

The time units value, effect the following parameters:

- The units of the "Eye diagram".
- Setup tab of the cursor measurements on "Eye diagram".
- Parameters of the eye measurement: "Eye width".
- Masks tests: user and eye
- Histogram: horizontal Histogram setup.
- "Settings" > "Appearance" > "Annotations" > "X position".

Remote command:
EYE<m>:HORizontal[:UNIT] on page 2414

**Range, Position ← Horizontal settings**
Available for Time units = "Absolute time"

Define the horizontal scale in absolute time.

"Range" Time range that is covered by the eye diagram.

"Position" Place of the zero point in the diagram, in seconds. The zero point is the alignment point on which the slice timestamps are superimposed.

Remote command:
EYE<m>:HORizontal:ABSolute:RANGe on page 2414
EYE<m>:HORizontal:ABSolute:POSition on page 2414

**Range UI, Position UI ← Horizontal settings**
Available for Time units = "Unit interval (UI)"

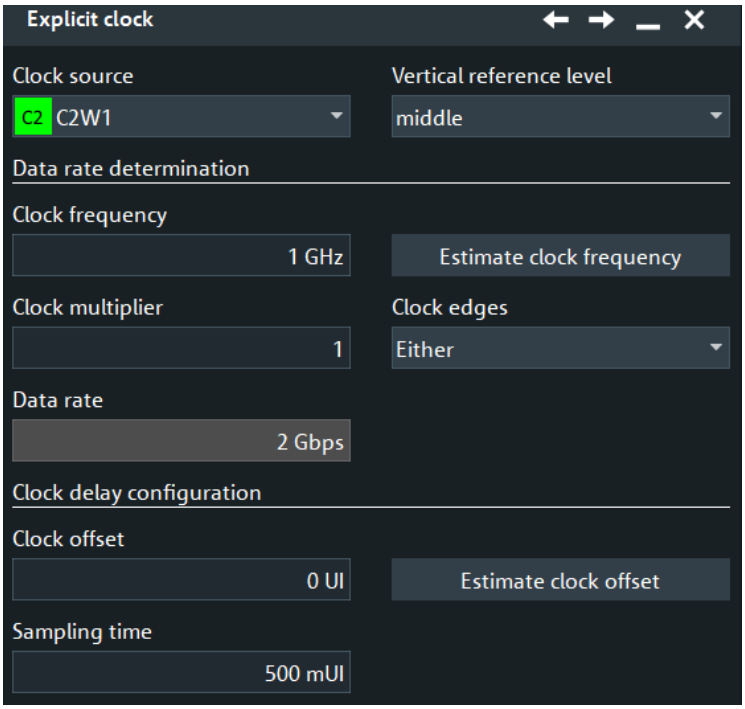
Define the horizontal scale in unit intervals. A unit interval typically corresponds to a bit in the input data signal.

"Range UI" Number of unit intervals that are visible in the eye diagram.

"Position UI" Place of the zero point in the diagram, in unit intervals. The zero point is the alignment point on which the segment timestamps are superimposed.

Remote command:
EYE<m>:HORizontal:UINTerval:RANGe on page 2415
EYE<m>:HORizontal:UINTerval:POSition on page 2415

**Explicit clock**
Opens a dialog to define the setting of the explicit clock for "Clock / CDR / Protocol" = "Clock".




**Vertical reference level ← Explicit clock**
Sets the reference level to define the timestamps for slicing the data waveform. A timestamp is set each time when the selected slope of the timing waveform crosses the selected reference level.

Remote command:
EYE<m>:TIMReference:CLOCk:LEVel on page 2416

**Clock frequency ← Explicit clock**
Sets the clock frequency of the explicit clock signal. If this value is unknown, you can use "Estimate clock frequency".

Remote command:
EYE<m>:TIMReference:CLOCk:FREQuency on page 2416

**Estimate clock frequency ← Explicit clock**
Calculates an estimate value for the clock frequency.

Remote command:
EYE<m>:TIMReference:CLOCk:FREStimation on page 2416

**Clock multiplier ← Explicit clock**
Sets a value for the clock multiplier, the ratio of an internal clock rate to the externally supplied clock.

Remote command:
EYE<m>:TIMReference:CLOCk:FACTor on page 2415

**Clock edge ← Explicit clock**
The clock edge sets the point in time when the state of the data signal is checked:

"Positive" Rising edge, a positive voltage change.

"Negative" Falling edge, a negative voltage change.

"Either" Both the rising and the falling edge.

Remote command:
EYE<m>:TIMReference:CLOCk:EDGE on page 2415

**Data rate ← Explicit clock**
Displays the calculated data rate. It is calculated as follows:

*"Data rate" = "Clock frequency" * "Clock multiplier"*X*

Where *X* is:

- 1 for "Clock edge"= "Positive" or "Negative"
- 2 for "Clock edge"= "Either"

Remote command:
ADVJitter<m>:SIGNal:DATRate? on page 2383

**Clock offset ← Explicit clock**
Sets a clock offset. If this value is unknown, you can use "Estimate clock offset".

Remote command:
EYE<m>:TIMReference:CLOCk:OFFSet on page 2417

**Estimate clock offset ← Explicit clock**
Calculates an estimate value for the clock offset.

Remote command:
EYE<m>:TIMReference:CLOCk:OFEStimation on page 2417

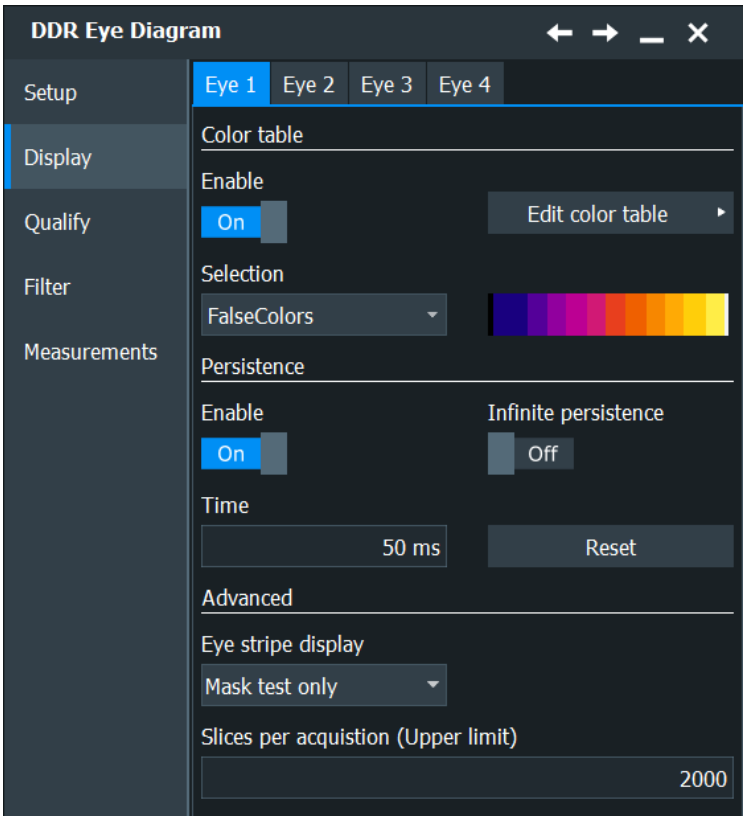
**Sampling time ← Explicit clock**
Sets a sampling time for the explicit clock signal, an offset for the clock edge in relation to the bit start. The clock edge sets the beginning of the unit interval. The sampling time is a number between 0 and 1. Value 0 sets the clock edge to the beginning of the bit period; value 0.5 sets the clock edge to the middle of the bit period.

Remote command:
EYE<m>:TIMReference:CLOCk:SAMTime on page 2417

### 19.2.2 Display settings

Access: [Apps]> "Analysis" > "DDR Eye" > "Display" tab

In the "Display" tab, you define the appearance of the individual eye diagrams.




You can configure and analyze up to four independent eye diagrams in parallel (vertical subtabs Eye 1 to Eye 4). Make sure to select the correct eye diagram subtab on the left before you adjust the settings.

**Color table**
Like any other waveform, you can show the eye waveform in solid color, or apply a color table.

For a description of the color table settings, see Chapter 4.3.2, "Color tables", on page 79.

**Persistence**
Each eye diagram has its specific persistence settings, independent of the general waveform persistence.

The persistence settings influence the way that the "Wave Count" and "Event Count" statistics on eye measurements are calculated:

- If "Enable" and "Infinite persistence" are disabled:
  - "Wave Count": for every image used in an eye measurement, sums up the number of acquisitions that contributed to the advanced eye image.
  - "Event Count": the number of UIs of images with successful eye measurement.

- If "Infinite persistence" is enabled:
  - "Wave Count": the number of acquisitions that contributed to the last advanced eye image used in an eye measurement.
  - "Event Count": the number of UIs of the last advanced eye image with successful eye measurement.

See also: Chapter 8.2.12.1, "Statistics", on page 384.

**Enable ← Persistence**
If enabled, each new data point in the diagram area remains on the screen for the duration that is defined using Time, or as long as Infinite persistence is selected.

If disabled, the waveform points are displayed only for the current acquisition.

Remote command:
`EYE<m>:DISPlay:PERSistence:STATe` on page 2420

**Infinite persistence ← Persistence**
If infinite persistence is enabled, each new waveform point remains on the screen until this option is disabled. Use infinite persistence to display rare events in the signal.

Remote command:
`EYE<m>:DISPlay:PERSistence:INFinite` on page 2420

**Time ← Persistence**
Sets a time factor that controls how long the waveform points fade away from the display. Thus, the R&S RTP emulates the persistence of analog phosphor screens.

Remote command:
`EYE<m>:DISPlay:PERSistence:TIME` on page 2420

**Reset ← Persistence**
Resets the display, removing persistent all waveform points.

Remote command:
`EYE<m>:DISPlay:PERSistence:RESet` on page 2420

**Eye stripe display**
Defines when the eye stripe is shown: always, never, or during mask tests.

The eye stripe is shown at the bottom of the main source waveform. It shows the part of the waveform that contributes to the eye. When a mask test is performed on the eye diagram, the eye stripe shows the position of the mask violations in red color.

Remote command:
`EYE<m>:DISPlay:MODE` on page 2419

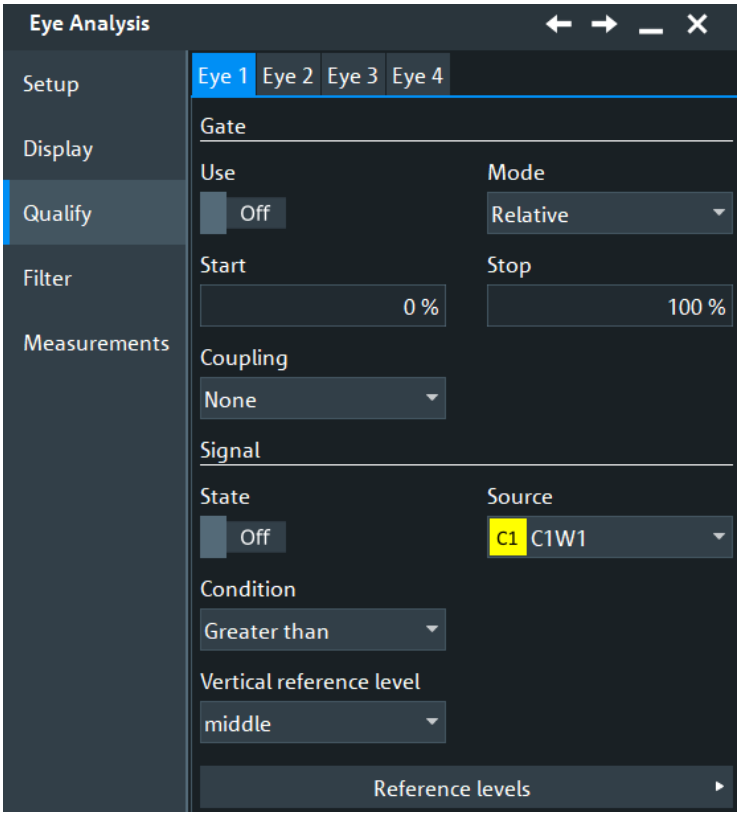
**Slices per acquisition**
Sets the number of waveform slices for a single acquisition.

Remote command:
`EYE<m>:MSLices` on page 2419

### 19.2.3 Qualify settings

Access: [Apps]> "Analysis" > "DDR Eye" > "Qualify" tab

Using the qualification settings, you can restrict the slices that contribute to the eye diagram. The restriction can be in horizontal direction (gate), or in vertical direction using the reference level.




You can configure and analyze up to four independent eye diagrams in parallel (vertical subtabs Eye 1 to Eye 4). Make sure to select the correct eye diagram subtab on the left before you adjust the settings.

**Gate**
Restricts the contributing slices in horizontal direction. Only the timestamps within the defined horizontal interval qualify for eye generation.

To apply the gate settings, enable the "State".

The gate definition is the same as gate definition for measurements, but the gates are independent. You can define the gate in one of these ways:

- Enter absolute time values for gate start and gate end: "Coupling mode = None" and "Mode = Absolute". Set "Start" and "Stop" time.
- Enter relative values for gate start and gate end: "Coupling mode = None" and "Mode = Relative". Set "Relative start" and "Relative stop" in percent.
- If a cursor measurement is active, you can use the vertical cursor lines as gate limits: "Coupling mode = Cursor". Select the cursor set.
- If a zoom is active, you can use the zoom limits as gate: "Coupling mode = Zoom". Select the zoom.

Remote command:
EYE<m>:QUALify:GATE[:STATe] on page 2421
EYE<m>:QUALify:GATE:COUPling on page 2421
EYE<m>:QUALify:GATE:MODE on page 2422
EYE<m>:QUALify:GATE:ABSolute:STARt on page 2422
EYE<m>:QUALify:GATE:ABSolute:STOP on page 2422
EYE<m>:QUALify:GATE:RELative:STARt on page 2422
EYE<m>:QUALify:GATE:RELative:STOP on page 2422
EYE<m>:QUALify:GATE:CURSor on page 2423
EYE<m>:QUALify:GATE:ZDIagram on page 2423

**Signal**
Sets a waveform-based qualification for the eye generation. The voltage level of a waveform is compared with a threshold, and only if this condition is fulfilled, the slices contribute to the eye generation.

**State ← Signal**
Applies the signal qualification.

Remote command:
EYE<m>:QUALify:SIGNal[:STATe] on page 2424

**Source ← Signal**
Selects the waveform to be compared.

Remote command:
EYE<m>:QUALify:SIGNal:SOURce on page 2424

**Condition ← Signal**
Selects the comparison condition for the waveform.

Remote command:
EYE<m>:QUALify:SIGNal:CONDition on page 2424

**Vertical reference level ← Signal**
Selects the threshold. You can use the lower, middle, or upper reference level. These levels are the reference levels that are defined for measurements.

See Chapter 8.2.4, "Reference levels", on page 332.

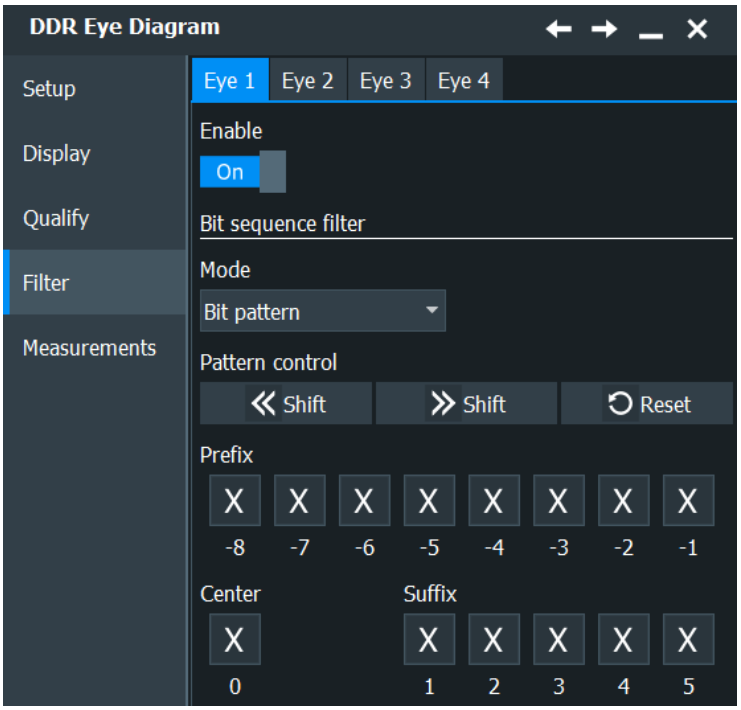
Remote command:
EYE<m>:QUALify:SIGNal:LSELect on page 2425

**Reference levels**
Opens the reference level dialog. For details, see Chapter 8.2.4, "Reference levels", on page 332.

### 19.2.4 Filter settings

Access: [Apps]> "Analysis" > "DDR Eye" > "Filter" tab

Filter settings allow you to select waveform slices for eye generation dependent on the bit status.




You can configure and analyze up to four independent eye diagrams in parallel (vertical subtabs Eye 1 to Eye 4). Make sure to select the correct eye diagram subtab on the left before you adjust the settings.

**Enable**
Applies the filter settings.

Remote command:
`EYE<m>:FILTer:STATe` on page 2425

**Mode**
Selects which bits contribute to the eye generation.

| | |
|---|---|
| "All bits" | All bits contribute to the eye diagram, same as "State" = off. |
| "Level transition" | Only bits after a signal transition contribute (0 to 1, or 1 to 0). |
| "Constant level" | Only bits without a preceding signal transition contribute (0 to 0, or 1 to 1). |
| "Bit pattern" | Only bits contribute that match the defined pattern before and after the center eye. Use the bit pattern, for example, to measure the effect of data-dependent jitter on the eye diagram. |

Remote command:
`EYE<m>:FILTer:BPATtern:MODE` on page 2426

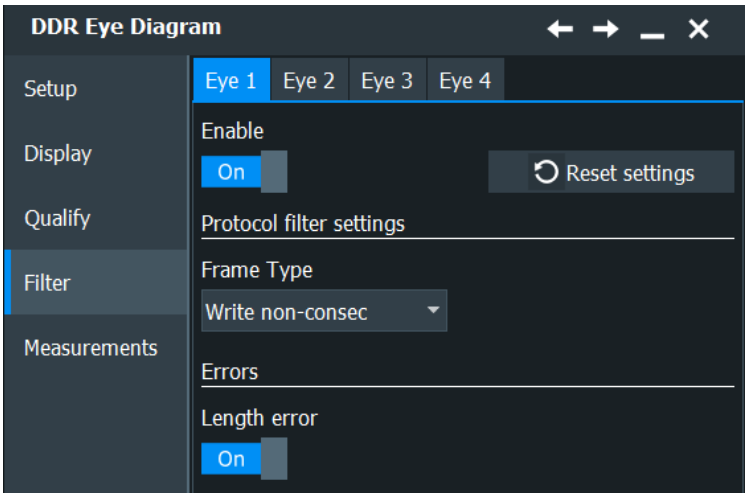
**Pattern**
Defines the bit pattern if "Mode" = "Bit pattern". You can set up to 8 bits before and 5 bits after the center eye (reference bit).

Remote command:
EYE<m>:FILTer:BPATtern:PREFix<1..8> on page 2426
EYE<m>:FILTer:BPATtern:CENTer on page 2426
EYE<m>:FILTer:BPATtern:SUFFix<1..5> on page 2426

**Pattern control**
Shift the pattern bit by bit to the left or to the right, and reset to the default (X for all bits).

Remote command:
EYE<m>:FILTer:BPATtern:SLEFt on page 2426
EYE<m>:FILTer:BPATtern:SRIGht on page 2427
EYE<m>:FILTer:BPATtern:RESet on page 2427

**Protocol filter settings**
This setting is only available, if the Source is set to "SBx" and the "DDR" serial bus analysis is enabled for the same bus.




**Reset settings ← Protocol filter settings**
Resets the current filter settings.

**Frame Type ← Protocol filter settings**
Selects which frame types contribute to the eye diagram, write frames, read frames or any of them.

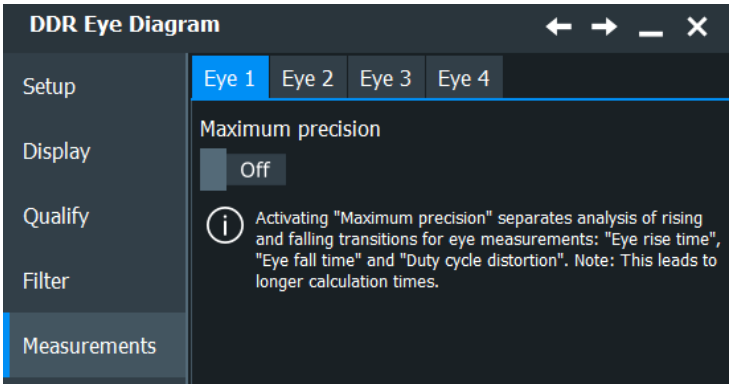
Remote command:
EYE<m>:FILTer:DDR:FTYPe on page 2427

**Length Error ← Protocol filter settings**
If enabled, the length errors contribute to the eye diagram.

Remote command:
EYE<m>:FILTer:DDR:LENerror on page 2427

### 19.2.5 Measurements settings

Access: [Apps]> "Analysis" > "DDR Eye" > "Measurement" tab




**Maximum precision**
If this function is enabled, the rising and falling transitions for the eye measurements "Eye rise time", "Eye fall time" and "Duty cycle distortion" are separated. This leads to longer calculation times.

See also: Chapter 8.2.6, "Eye diagram measurements", on page 355.

Remote command:
`EYE<m>:MEASurement:MAXPrecision` on page 2419

## 19.3 Measurements and zoom on eye diagrams

To analyze eye diagrams, you can use the following basic functions of the R&S RTP:

- **Cursor measurements**
  Cursors can be used on eye diagrams in the usual way. "Track waveform" is not supported.
  For details, see Chapter 8.1, "Cursor measurements", on page 308.
- **Automatic eye measurements**
  Eye measurements can be used on advanced eye diagrams in the usual way.
  For details, see Chapter 8.2.6, "Eye diagram measurements", on page 355.
- **Histograms**
  You can create histograms on eye diagrams as usual.
  For details, see Chapter 8.2.8, "Histograms and histogram measurements", on page 368.
- **Zoom**
  You can zoom into eye diagrams using the zoom functions.
  For details, see Chapter 7.1, "Zoom", on page 261.

## 19.4 Mask Testing on Eye Diagrams

Mask testing on eye diagrams allows you to test data signals against eye shapes that are required in the standards.

You can select the shape of the eye, enter its dimensions and position the eye on the display. For mask configuration details, see Chapter 10.2.2, "Mask definition", on page 427. The fail criteria for the test is defined as usual for R&S RTP mask tests, see Chapter 10.2.3, "Define Test", on page 434.

The eye mask definition cannot be saved as a mask test. You can save the settings as user defined preset and recall them by loading the preset file. See: Chapter 4.6, "Preset setup", on page 101.

# 20 Deembedding (option R&S RTP-K121/-K122)

Deembedding removes the parasitic effects of the measurement setup from the measured signal. A simple measurement setup consists of a probe only, but more complex setups include also cables, fixtures and other components. The effects of these components on the measurement are typically increasing when signal frequency increases. Thus, deembedding is useful or even necessary when measuring signals of 4 GHz frequency or higher. Furthermore, you can virtually move the measurement point to a point in a circuit that cannot be reached by probing. In this case, the effects of the components between the real and ideal probe are deembedded.

The components of a measurement setup are usually multiports, and each multiport can be described by a scattering matrix. The elements of a scattering matrix are the S-parameters. The S-parameters of a multiport are usually measured using vector network analyzers (VNA), and they are saved in Touchstone files.

The R&S RTP also can measure probes and cables and determine their S-parameters. These components are called proven probe or proven cable, and their characteristics are saved in `*.rs2p` files. The measurements require hardware option R&S RTP-B7. For proven probe, also the R&S RT-ZF30 probe test fixture is needed

From the S-parameters of the measurement components, the deembedding option determines the transfer function for the measurement setup. Based on the transfer function, filter coefficients are calculated, and the filter is applied to the measured signal.

Option R&S RTP-K121 realizes the deembedding process in software. As the process requires some time, triggering on the corrected signal is not possible, and the acquisition rate decreases.

Option R&S RTP-K122 realizes the deembedding process in hardware. This process is fast, so you can trigger on the corrected signal, and the acquisition rate remains unchanged. However, deembedding by hardware is typically less precise but the accuracy is sufficient for most measurements.

## 20.1 Configuring the measurement setup for deembedding

The R&S RTP provides an easy way to model the measurement setup on the instrument, and to enter the characteristics of the components. If you use an R&S RT-ZM, R&S RT-ZD or R&S RT-ZS probe with Rohde & Schwarz interface, the setup is easy: The probe is recognized by the oscilloscope, and the oscilloscope knows the characteristics of the probe components.

Before you start, you need to know the following:

- Probing method: probing or terminating
- S-parameter files of the components of the measurement setup (Touchstone files).

S-parameters can be listed as complex numbers (real and imaginary parts), or as magnitude and phase in the order of increasing frequency. The phase unit has to be degree (°).

Proceed as follows:

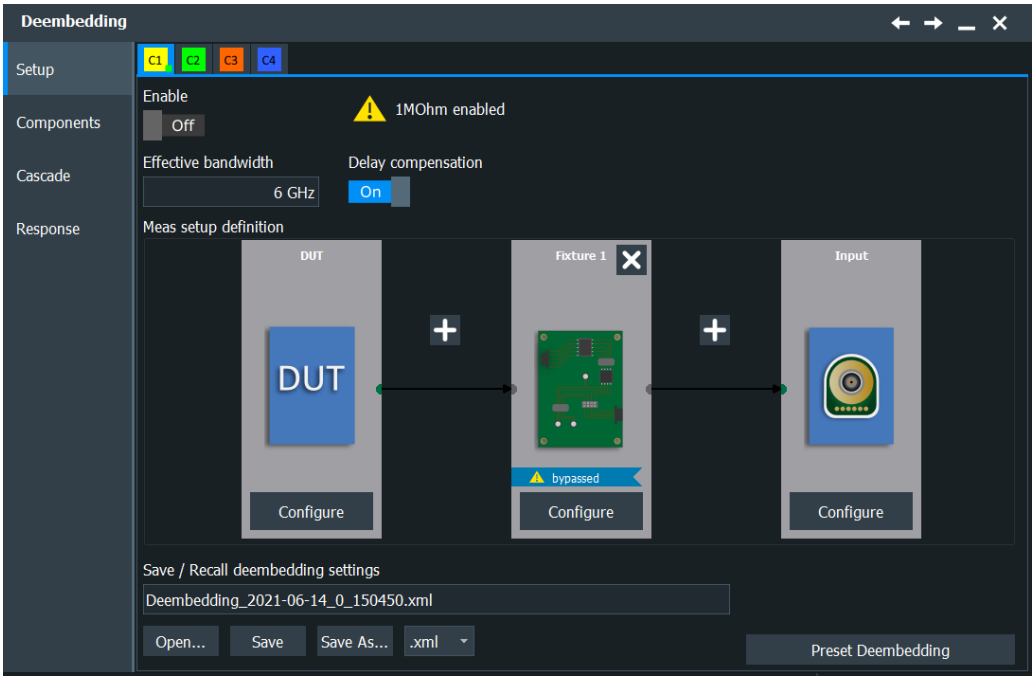
1. Select [Apps] > "Analysis".
2. Select "Deembedding".
3. Select the input channel to which the measurement setup is connected.
4. Add the components of the measurement setup:
   a) Tap the "+" icon.
   b) Select the component type.

   The component is added to the setup, and it is disabled by default (bypassed). The DUT and a probe or other input component are included by default.
5. Configure each component:
   a) Tap the "Configure" button.
   b) Enter the settings as required.
   All settings are described in Chapter 20.2, "Deembedding settings", on page 1088.
   c) If necessary, tap "Open" and load the S-parameters from file.
   As long as the configuration is incomplete or conflicting, the message "Invalid configuration" is shown.
   d) Enable the component to consider it during deembedding.
   e) Tap the subtab of the next component, and configure it in the same way.
6. On the "Setup" tab, set the Effective bandwidth and the Delay compensation.
7. Enable the deembedding.
8. If you want to keep the configuration for later use, save it to file.

## 20.2 Deembedding settings

### 20.2.1 Measurement setup

In the "Deembedding" > "Setup" tab, you add all components of the measurement setup and define all settings that are related to the complete path.




**Enable**
Activates the deembedding - the correction of parasitic effects of the measurement setup on the measured signal.

Remote command:
`DEEMbedding<m>[:STATe]` on page 2432

**Effective bandwidth**
Sets the maximum bandwidth until which the signal is corrected. This maximum value is the minimum bandwidth value of the probe bandwidth, tip module bandwidth (R&S RT-ZM) and oscilloscope bandwidth, and can not be higher than the highest frequency in a used S-parameter file. The setting is also coupled with the digital filter bandwidth if the cut-off frequency of the digital filter is at least 1 GHz.

Remote command:
`DEEMbedding<m>:BANDwidth` on page 2434

**Delay compensation**
Enables the compensation for the group delay of the complete measurement setup. If enabled, the calibration time reference point is the start point of the measurement setup. If disabled, the calibration time reference point is the oscilloscope's channel input.

Remote command:
`DEEMbedding<m>:CMPDelay` on page 2435

**Realtime deembedding**
Requires real-time extension option R&S RTP-K122.

If enabled, the deembedding calculation is done by the hardware in real time. You can trigger on the corrected signal, and the acquisition rate remains unchanged.

If disabled, the deembedding calculation is done by software. This method is slower, therefore, triggering on the corrected signal is not possible.

Remote command:
DEEMbedding<m>:REALtime on page 2434

**Meas setup definition**
Simulates the complete measurement setup with all components.

To add a component, tap the **+** icon and select the component type: cable, fixture, adapter, proven probe, proven cable, or others ("Custom"). The DUT component is always included in the deembedding. Also, a connected R&S RT-ZM, R&S RT-ZD or R&S RT-ZS probe with Rohde & Schwarz interface is automatically recognized and included in the deembedding.

The remark "bypassed" indicates that the component is disabled and not considered in the deembedding calculation as long as it is not configured correctly.

Remote command:
DEEMbedding<m>:ADD on page 2432
DEEMbedding<m>:REMove on page 2433
DEEMbedding<m>:CCOunt? on page 2433

**Save / recall deembedding settings**
Saves the complete deembedding configuration, or loads a saved deembedding configuration. The file format is XML.

To save the deembedding configuration, enter a filename and tap "Save". Alternatively, tap "Save as" to open the file selection dialog box, see Chapter 12.5, "File selection dialog", on page 513.

To load a deembedding configuration, tap "Open" and select the required file.

Remote command:
DEEMbedding<m>:NAME on page 2435
DEEMbedding<m>:SAVE on page 2436
DEEMbedding<m>:OPEN on page 2436
DEEMbedding<m>:DELete on page 2436

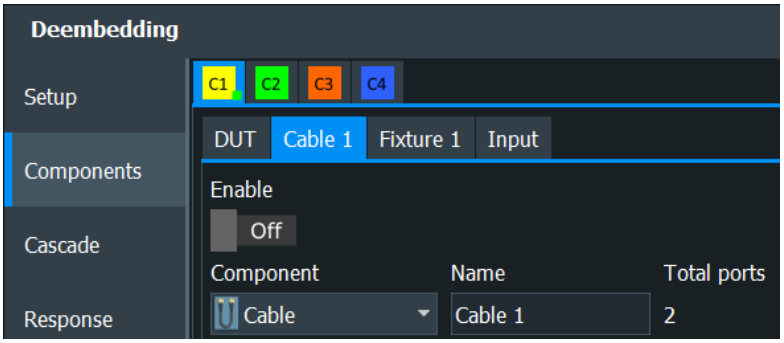
**Preset Deembedding**
Sets the complete deembedding configuration to the default values.

Remote command:
DEEMbedding<m>:PRESet on page 2435

### 20.2.2 Common settings for components

The "Components" tab has several subtabs to configure each component of the measurement setup that was defined on the "Setup" tab.

Settings that are available on all subtabs are described in this chapter. Other settings are component-specific and explained in the following chapters.




**Enable**
Enables the component for deembedding. As long as the configuration is incomplete or conflicting, the message "Invalid configuration" is shown, and enabling is not possible.

Remote command:
`DEEMbedding<m>:COMPonent<n>[:STATe]` on page 2437

**Component**
Selects the component type. The "DUT", "Input" and "Probe" types are predefined and shown for information. The other types ("Fixture", "Cable", "Adapter", and "Other") are selectable.

Remote command:
`DEEMbedding<m>:COMPonent<n>:TYPE` on page 2437

**Name**
Enter a name for the component that helps you identifying it.

Remote command:
`DEEMbedding<m>:COMPonent<n>:NAME` on page 2438

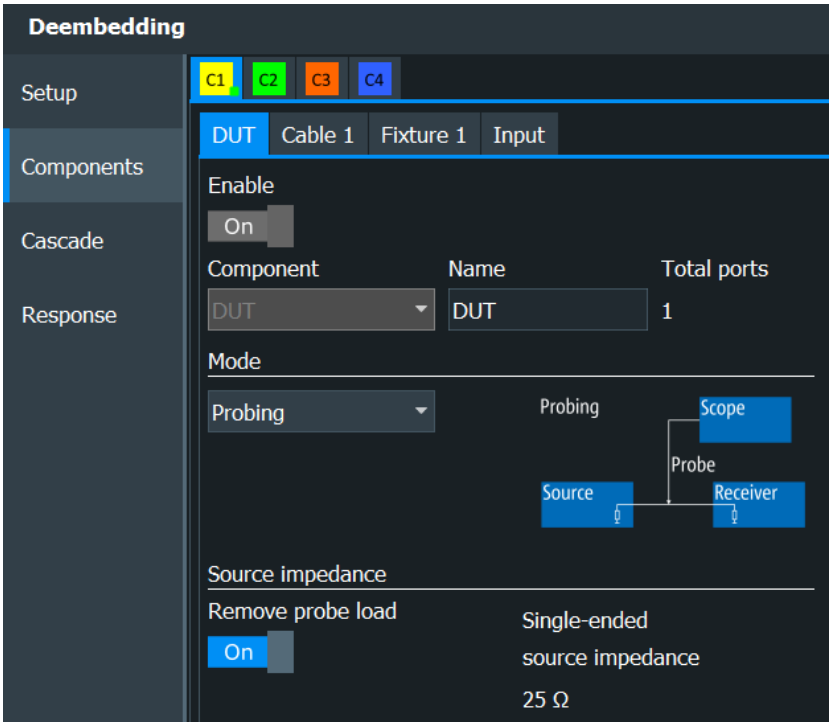
**Total ports**
Shows the number of all ports of the selected component (input ports and output ports).

The number of ports depends on the attached probe. If a modular probe R&S RT-ZM or differential probe R&S RT-ZD is used, the number of ports is 4. Otherwise, 2 ports are available.

Remote command:
`DEEMbedding<m>:COMPonent<n>:PCOunt?` on page 2438

### 20.2.3 DUT settings

In the "Deembedding" > "Components" > "DUT" tab, you set the characteristics of the DUT.




**Probing, Terminating**
Sets the probing mode.

Select "Probing" if you use a high-ohmic probe and measure on an existing line in parallel to the load.

Select "Terminating" if the measuring equipment is the load of the line.

Remote command:
`PROBe<m>:DEEMbedding:MODE` on page 1273
`DEEMbedding<m>:COMPonent<n>:MODE` on page 2438

**Remove probe load**
If "Probing" is selected, you can remove the loading of the probe.

If the probe loading is removed, you see the signal that would be at the measurement point if the probe's input impedance is ideal (infinite impedance).

Without removing the loading, you see the real signal at the measurement point, including the probe loading.

Remote command:
`PROBe<m>:DEEMbedding:REMProbeload` on page 1273
`DEEMbedding<m>:COMPonent<n>:LOAD:REMProbeload` on page 2439

**Source impedance**
Shows the source impedance of the DUT. If an R&S RT-ZM probe is connected, the value depends on the selected probe mode: common mode, differential, or single-ended measurement.

Remote command:
DEEMbedding<m>:COMPonent<n>:LOAD:IMPedance? on page 2439
PROBe<m>:DEEMbedding:LOAD:IMPedance on page 1274

### 20.2.4 R&S RT-ZM probe settings

If an R&S RT-ZM modular probe is connected to a channel input, the probe is detected by the oscilloscope and automatically added to the measurement path. The S-parameters of the probe, the tip modules, and the probe group delay are known to the oscilloscope and used for deembedding.

Probe settings that are important for deembedding can be set directly in the deembedding dialog.

To get all probe deembedding settings:

1. In the "Deembedding" > "Components" > "RT-ZMxx" tab, select "More Settings".
2. In the "Probes Setup" dialog box, select "Probe Deembedding".

The "Probe Deembedding" dialog box is described in Chapter 5.6.5.2, "Deembedding for modular probes", on page 160.

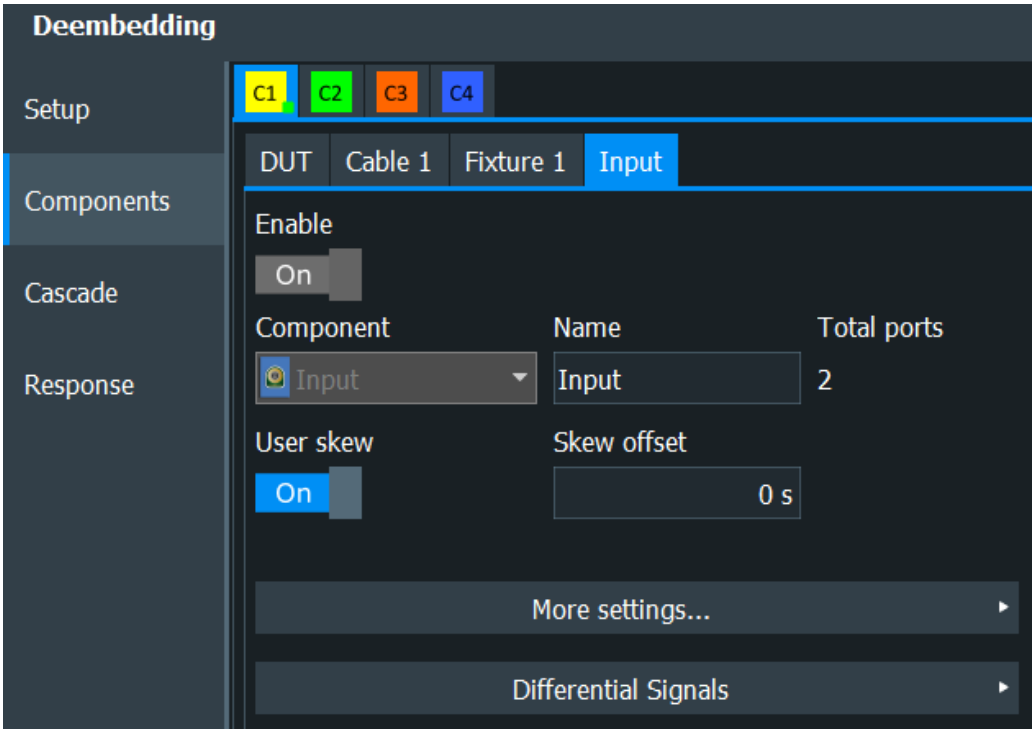
For details on probe parameters, see:

- "Select probe tip module, Pin spacing" on page 160
- "Probe mode" on page 159
- "DM Offset, CM Offset, P Offset, N Offset" on page 160

### 20.2.5 Input settings for other probes

The "Input" component is used for all probes except for R&S RT-ZM, R&S RT-ZD and R&S RT-ZS probes with Rohde & Schwarz interface, and for cables or other components that are used to connect the measurement setup to the oscilloscope. For these input components, the horizontal skew and vertical settings are considered during deembedding. You can set the skew directly in the "Deembedding" > "Components" > "Input" tab, or in "Horizontal" menu > "Skew". The button "More Settings" opens the "Channels" settings.

The input component is also used for deembedded differential signals, see Chapter 5.8.3, "Deembedding of differential signals", on page 187.




For details, see:

- "User skew" on page 126
- "Skew offset" on page 126

### 20.2.6 Cable, adapter, fixture and custom settings

If you have added a cable, adapter, test fixture or other component to the measurement setup, you set the characteristics in the corresponding subtabs of the "Deembedding" > "Components" tab.

The characteristics are the same for these component types. Configuration differs between 2-port and 4-port components, because 4-ports need additional settings. Generally, these components are configured by loading S-parameter files and analyzing its characteristics.

**To configure a component in a single-ended 2-port setup**

1. Tap "Open".
2. Select and load the `*.s2p` Touchstone file that contains the S-parameters of the component. The filetype is always `*.s2p`.
3. Assign the ports to the input and output of the component according to the data in the Touchstone file. See also: "Input 1, Input 2, Output 1, Output 2" on page 1096.
4. Analyze the response curves of each S-parameter. The response curves are functions of magnitude, phase, and group delay over frequency.
   a) Select the S-parameter.

b) Check the response curves. The display is based on a reference impedance of Z0 = 50 Ω.

The response curves show:
- The correctness of the S-parameter file.
- The correct setting of the input and output ports ("Input 1", "Output 1")
- The characteristics of the component.

5. If the component is configured correctly, tap "Enable" to include it in the deembedding setup.

**To configure a component in a single-ended 4-port setup**

1. If the component is a 4-port, i.e. a fixture or custom block, load the `*.s4p` Touchstone file that contains the S-parameters of the component:
   a) Tap "Open".
   b) Select and load the `*.s4p` Touchstone file. The filetype is always `*.s4p`.
2. If the component is a 2-port, i.e. a cable or adapter, typically 2 cables or adapters are used. These components can be different or identical.
   a) If the components are identical, enable "Identical Paths" on page 1096. You need 1 `*.s2p` Touchstone file.
   If the components are different, disable "Identical Paths" on page 1096. You need 2 `*.s2p` Touchstone files, one for each component.
   b) Select the filetype `*.s2p`.
   c) Tap "Open".
   d) Select and load the `*.s2p` Touchstone file.
   e) If the components are different, load the second `*.s2p` Touchstone file in the same way.
3. Assign the ports to the input and output of the component according to the data in the Touchstone file. See also: "Input 1, Input 2, Output 1, Output 2" on page 1096.
4. Analyze the response curves of each S-parameter. You can analyze magnitude, phase and group delay response, which are functions of frequency.
   a) Select the S-parameter.
   b) Check the response curves. The display is based on a reference impedance of Z0 = 50 Ω.

   The response curves show:
   - The correctness of the S-parameter file.
   - The correct setting of the input and output ports ("Input 1", "Output 1")
   - The characteristics of the component.
5. If the component is configured correctly, tap "Enable" to include it in the deembedding setup.

Response curves are described in Chapter 20.3.1, "Responses of the component cascade", on page 1097.

**Input 1, Input 2, Output 1, Output 2**
Assign the ports to the input and output of the component according to the data in the Touchstone file. This information is not directly written in the file. You need background information how the S-parameters were determined, i.e. which port was input and which port was output. If in doubt, the response curves help assigning the ports correctly.

Remote command:
`DEEMbedding<m>:COMPonent<n>:IPONe` on page 2442
`DEEMbedding<m>:COMPonent<n>:IPTWo` on page 2442
`DEEMbedding<m>:COMPonent<n>:OPONe` on page 2442
`DEEMbedding<m>:COMPonent<n>:OPTWo` on page 2442

**Identical Paths**
The setting is relevant if the measurement setup is a 4-port, and the component is a 2-port. Usually, these components are cables and adapters, which are described by `*.s2p` files. If there is a 4-port setup, 2 cables or 2 adapters are used. If there is no crosstalk in-between, each cable or adapter can be described by a `*.s2p` file.

If both components are different, each is described by its own `*.s2p` file. If they are identical, for example, when a matched-pair cable is used, they can be described by the same file. In this case, enable "Identical Paths".

Remote command:
`DEEMbedding<m>:COMPonent<n>:IDENtical` on page 2441

**S-Parameters, Filetype**
Loads the Touchstone file that contains the S-parameters of the component.

If the measurement setup is a 2-port, the filetype is always `*.s2p`.

If the measurement setup is a 4-port, the selection depends on the component. Fixtures and custom blocks are usually 4-ports, and they are described by `*.s4p` S-parameter files.

Cables and adapters are usually 2-ports, and they are described by `*.s2p` S-parameter files. If there is a 4-port setup, 2 cables or 2 adapters are used. You need 2 different files if the components are different, or only 1 file if they are identical (see also "Identical Paths" on page 1096).

Remote command:
`DEEMbedding<m>:COMPonent<n>:SPONe` on page 2442
`DEEMbedding<m>:COMPonent<n>:SPTWo` on page 2442

# 20.3 Responses

## 20.3.1 Responses of the component cascade

The "Deembedding" > "Cascade" tab combines the S-parameters of all included components and shows the response curves of the measurement setup for each S-parameter:

- Magnitude response
- Phase response
- Group delay response

While the response curves on the component tabs show the S-parameter characteristics of a single component, the cascade shows the combined characteristics for all components including the mismatch between the components. The DUT is not considered.

**Port selection**
Selects the S-parameter. The response curves (magnitude, phase, and group delay response) of the setup cascade for the selected parameter are shown. You can refresh each graphic, and display it on full screen.

## 20.3.2 Responses of the measurement setup

On the "Deembedding" > "Response" tab, you see the resulting transfer function characteristics of the deembedded measurement setup. These response curves include also the characteristics of the DUT and its interaction with the input connector of the oscilloscope.

You can see the following characteristics:

- Magnitude response
- Phase response
- Group delay response
- Step response

Using these response curves, you can assess the correction of the signal by deembedding. In particular, you can compare realtime deembedding with software deembedding and check whether the accuracy of the realtime deembedding is sufficient for the measurement task.

► To see more details in a response graphic, tap the upper right button in the graphic.

The graphic opens in full view, where you can analyze the response function: zoom in and out, use the cursor.

On the "Response" tab, you see also some parameters of the uncorrected measurement setup including the DUT. All parameters are related to a frequency range from DC to the configured effective bandwidth.

**$Att_{max}$**
Maximum attenuation of the measurement setup before deembedding, with reference to a frequency response normalized at DC to 0 dB. The deembedding filter must amplify the amplitude response by the "$Att_{max}$" value, and thus also increases the noise by this value. The value also helps to decide whether deembedding is useful for the defined effective bandwidth.

Remote command:
`DEEMbedding<m>:MATTenuation?` on page 2443

**$Att_{DC}$, $Gain_{DC}$**
Maximum attenuation or gain of the measurement setup before deembedding at DC. This value is the expected basic attenuation or gain, which is corrected by deembedding.

Remote command:
`DEEMbedding<m>:CADC?` on page 2443

**$Gain_{max}$**
Maximum gain of the measurement setup before deembedding, with reference to a frequency response normalized at DC to 0 dB. The deembedding filter must attenuate the amplitude response by this value. The value helps also to avoid an overload of the ADC.

Remote command:
`DEEMbedding<m>:MGAin?` on page 2444

## 20.4 Proven probe, proven cable

The R&S RTP can measure probes and cables and determine their S-parameters. These components are called proven probe or proven cable, and their characteristics are saved in `*.rs2p` files. The measurements require hardware option R&S RTP-B7. For proven probe, also the R&S RT-ZF30 probe test fixture is needed

### 20.4.1 Characterization of proven probes and cables

To get the S-parameters of proven probes and cable, several measurements are needed. A wizard guides you through the process, and graphics show how to connect the setup. You can also save the result of each step and speed up further measurements by saved step results.

The wizard steps are the following:

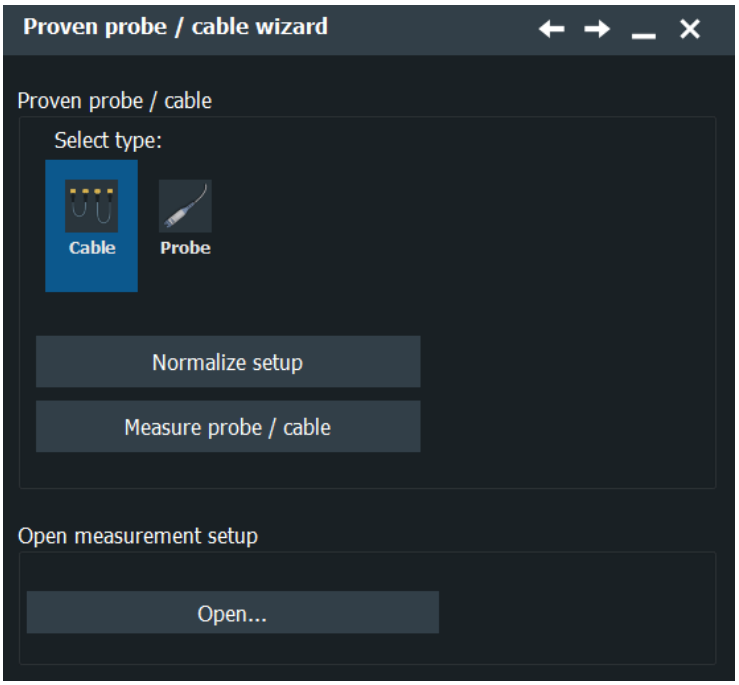
- Proven probe / cable setup ..... 1099
- Normalization ..... 1100
- Measurement ..... 1101
- Response ..... 1102

#### 20.4.1.1 Proven probe / cable setup

Access: [ Apps] > "Analysis" > "Deembedding" > "Setup" tab: include component "Proven Probe" or "Proven Cable" > "Configure" > "Start wizard"

Depending on the steps that you have done before, you can:

- Start the complete characterization procedure.
- Start the probe or cable measurement in a normalized setup.
- Load measurement results.




**Select type**
Select the DUT to be characterized: probe or cable

**Normalize setup**
Startes the wizard for the complete normalization and measurement procedure.

**Measure probe / cable**
Starts the measurement of the probe or cable directly. If the measurement setup has been normalized just before, the "Measurement" dialog box opens. Otherwize, you are asked to open a normalization file, or to start with normalization.

**Open measurement setup**
Opens a file with previously saved measurement results. The "Response" dialog box opens, and you can check the response plots, and save or apply the S-parameters to the deembedding setup.

### 20.4.1.2 Normalization

Open [Apps]> "Analysis" tab > "Proven Probe / Cable" > "Normalize setup"

1. Connect the cables to the pulse source R&S RTP-B7 and to the input channels.
   The probe or cable to be measured is *not* connected.
   Use only high-quality double-shielded RF cables. Two suitable cables are delivered with the pulse source option R&S RTP-B7.

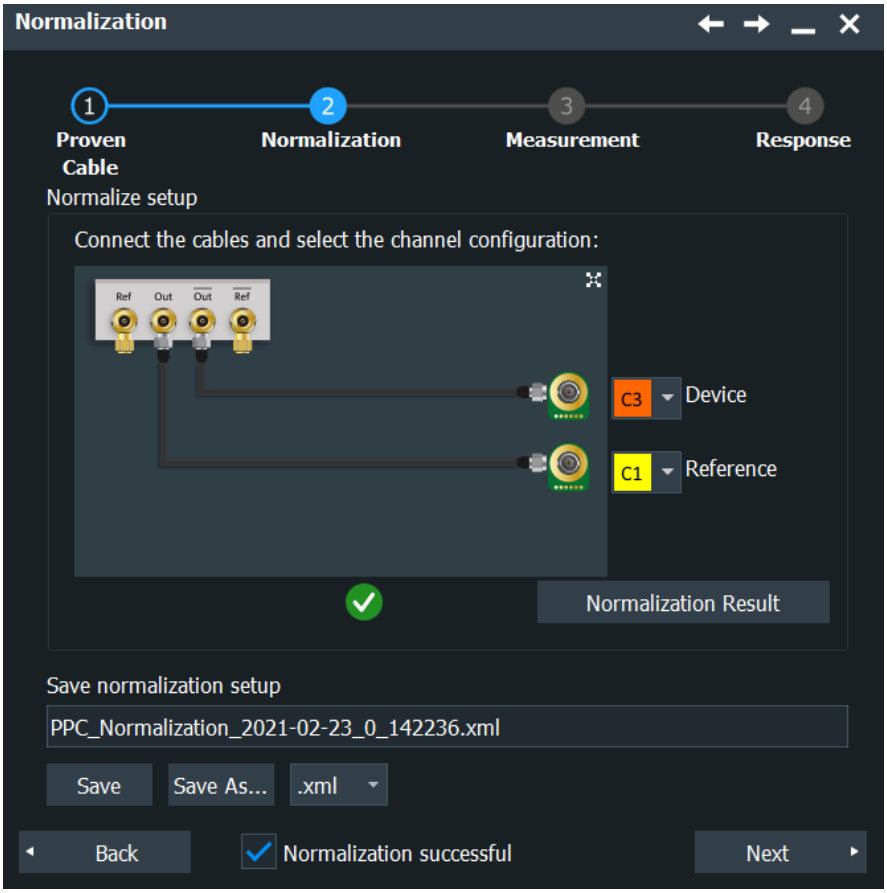



***Figure 20-1: Normalization settings***

2. Start measurements with "Normalize".

   The basic setup is characterized.

**Device**
Selects the input channel that is connected to the $\overline{\text{Out}}$ connector of the pulse source R&S RTP-B7 during normalization, and to the measured probe or cable during measurements.

**Reference**
Selects the input channel that is connected to Out connector of the pulse source.

**Normalize**
Starts the normalization of the setup without the probe or cable to be measured. The frequency correction and deskew are determined and applied in the next measurement step.

If normalization was successful, it is indicated by an icon, and you can display the "Normalization Result" - the normalized magnitude response.

**Save normalization setup**
Saves the results of the normalization step to an XML file.

### 20.4.1.3 Measurement

Open [Apps]> "Analysis" tab > "Proven Probe / Cable" > "Normalize setup"> "Next"

1. To characterize a cable, connect it to the cable coming from the Out connector of the pulse source and the "Device" input channel. Use an adapter if necessary.

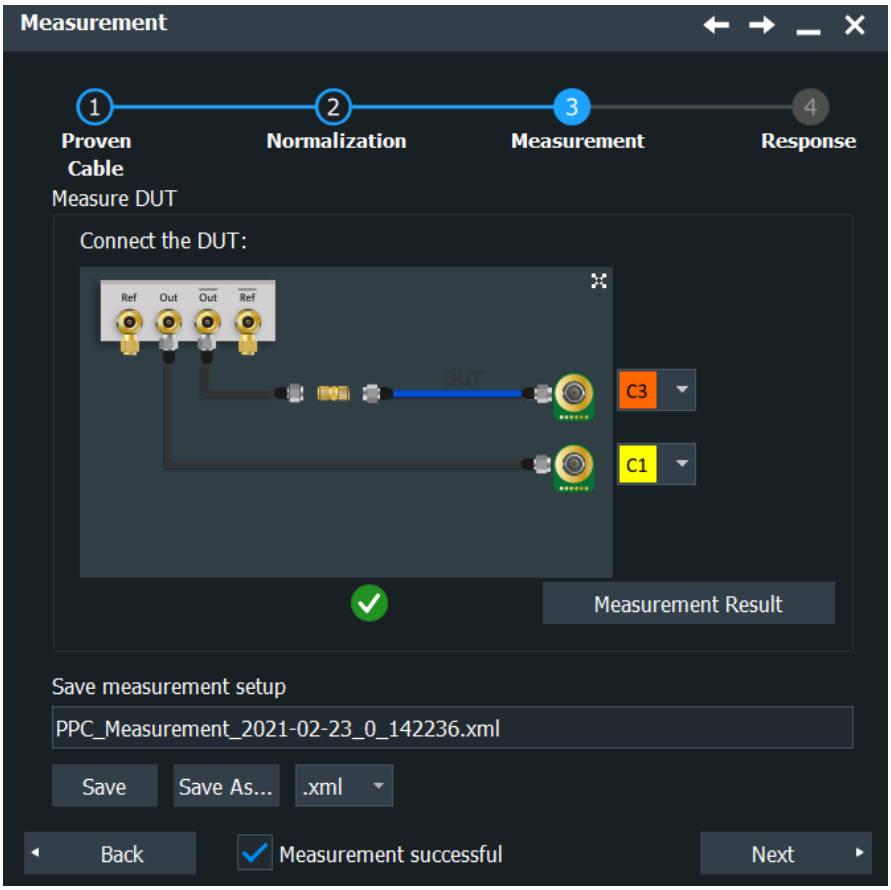



*Figure 20-2: Cable measurement*

2. To characterize a probe, you need the R&S RT-ZF30 probe test fixture.
   a) Connect the fixture to the cable coming from the Out connector of the pulse source.

b) Connect the probe to the fixture.
For details, see the "R&S RT-ZF30 Probe Test Fixture Manual".
c) Connect the probe to the "Device" input channel.

3. Select "Measure".

**Measure**
Starts the measurement of the probe or cable to be characterized.

If the measurement was successful, it is indicated by an icon, and you can display the "Measurement Result" - the magnitude response of the proved probe or cable.

**Save measurement setup**
Saves the measurement results of the proven probe or cable to an XML file.

#### 20.4.1.4 Response

Open [Apps]> "Analysis" tab > "Proven Probe / Cable" > > "Normalize setup" > "Next" > "Measure" > "Next"

In the last step, you see the response curves, the charcteristics of deembebbed proven probe or cable. The characteristics are the same as on the "Deembedding Response" tab. For more information, see Chapter 20.3.2, "Responses of the measurement setup", on page 1097.

**Apply result**
Includes the proven probe or proven cable component in the deembedding setup, and opens the "Deemebedding Components" tab for configuration.

**Save result**
Saves the S-parameters of the proven probe or cable to a rs2p or rs4p file. Using this file, you can include and configure this proven probe or cable as a component in a deembedded measurement setup.

### 20.4.2 Deembedding with proven probe or proven cable

If a probe or cable is characterized as described in Chapter 20.4.1, "Characterization of proven probes and cables", on page 1098, and its S-parameters are saved, you can use this probe or cable in any deembedding measurement setup.

1. In the deembedding measurements setup, add the component "Proven Probe" or "Proven Cable".
2. Select "Configure" on the new component.
3. Load the S-parameters file (`*.rs2p`).
If no file is available, you can start the characterization wizard here.

See also: Chapter 20.1, "Configuring the measurement setup for deembedding", on page 1087.

# 21 External frontend control (option R&S RTP-K553)

If the External Frontend Control option (R&S RTP-K553) is installed, you can use one or more external frontends to analyze signals with the R&S RTP.

In addition to the R&S RTP-K553 option, you need at least one of these options to process the signal data in the oscilloscope:

- R&S RTP-K11 I/Q software interface
- R&S RTP-K121 Deembedding

To analyze the signal data, you need a suitable software on a connected computer:

- R&S VSE Vector Signal Explorer Software
- Another software that can receive data via remote control and analyze it.

## 21.1 External frontend basics

New technology standards, such as 5G and 6G, increase the demand of higher carrier frequencies and higher bandwidth. For example, the D band of 6G requires carrier frequencies up to 170 GHz and bandwidth up to 10 GHz. To measure these signals in a flexible and cost efficient way, the R&S RTP oscilloscope can be combined with external frontends.

The external frontend downconverts the high-frequency RF input signal to a lower intermediate frequency IF signal using its own internal local oscillator. For all subsequent signal processing steps in the R&S RTP, this process is transparent. Thus, the external frontend extends the frequency range of the oscilloscope, for example, up to 170 GHz with external frontend R&S FE170SR and R&S RTP with 13 GHz or 16 GHz bandwidth.

In complex measurement setups, long RF cables between the DUT and the analyzer are often necessary due to the physical dimensions of high-performance oscilloscopes. When measuring high frequency signals, the long RF cable can lead to unwanted loss. An external frontend with smaller physical dimensions allows for measurements closer to the DUT. The measurement signal is transferred from the DUT to the oscilloscope on a lower intermediate frequency (IF), for which loss does not have a major effect.

The R&S RTP supports the following external frontend types:

- R&S FE44S
- R&S FE50DTR (Rx only)
- R&S FE170SR

Refer to the oscilloscope's data sheet for an up-to-date list of supported external frontends.

For details on setting up the external frontend, see the documentation of the used device.

The external frontends can support several frequency bands. In connection with oscilloscope, only the lower band is supported.

Depending on the frontend type and oscilloscope bandwidth, up to 4 frontends can be connected.

- R&S FE44S and R&S FE50DTR: max. 4 external frontends
- R&S FE170SR: max. 2 frontends, requires R&S RTP with 13 GHz or 16 GHz bandwidth

The external frontend is controlled by the R&S RTP via a LAN connection. Up to 2 LAN/SSL connections are supported by the frontend. The data is transferred to the oscilloscope using an IF cable. In addition, a reference signal is provided from the R&S RTP to one or all connected frontends.

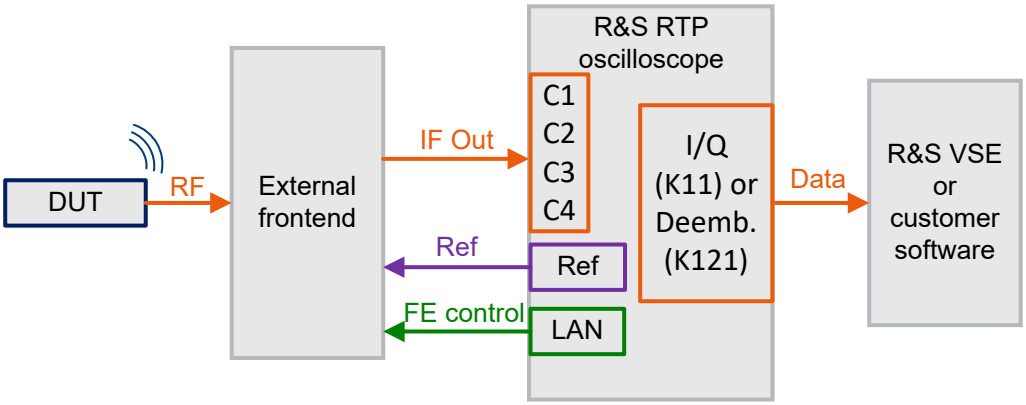



*Figure 21-1: Schematic test setup with external frontend*

The oscilloscope receives the downconverted signals and provides I/Q data or corrected raw data for further analysis. For data analysis, you can use the R&S VSE Vector Signal Explorer software, or another suitable analysis software.

## 21.2 Test setups

You can connect several external frontends to the R&S RTP in different ways. To analyze the data that is acquired, you need a computer with an analyzing software, for example, R&S VSE.

The common points of all test setups are the following:

- The frontends are controlled by the R&S RTP.
- The software on the computer controls the R&S RTP. From this perspective, frontends and oscilloscope look like one unit.
- The frontends have encrypted control interface.
- IF cables with good quality up to 16 GHz are required.

The basic setup uses one external frontend to analyze one channel.

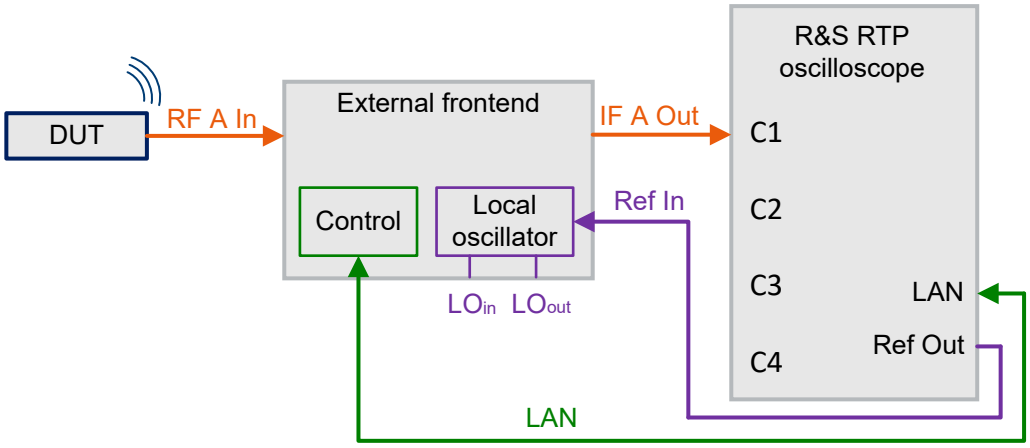



*Figure 21-2: Simple test setup for analysis of one channel*

### Coupled references

In test setup with coupled references, each frontend gets the reference frequency from the oscilloscope. Coupling the references avoids frequency offset.

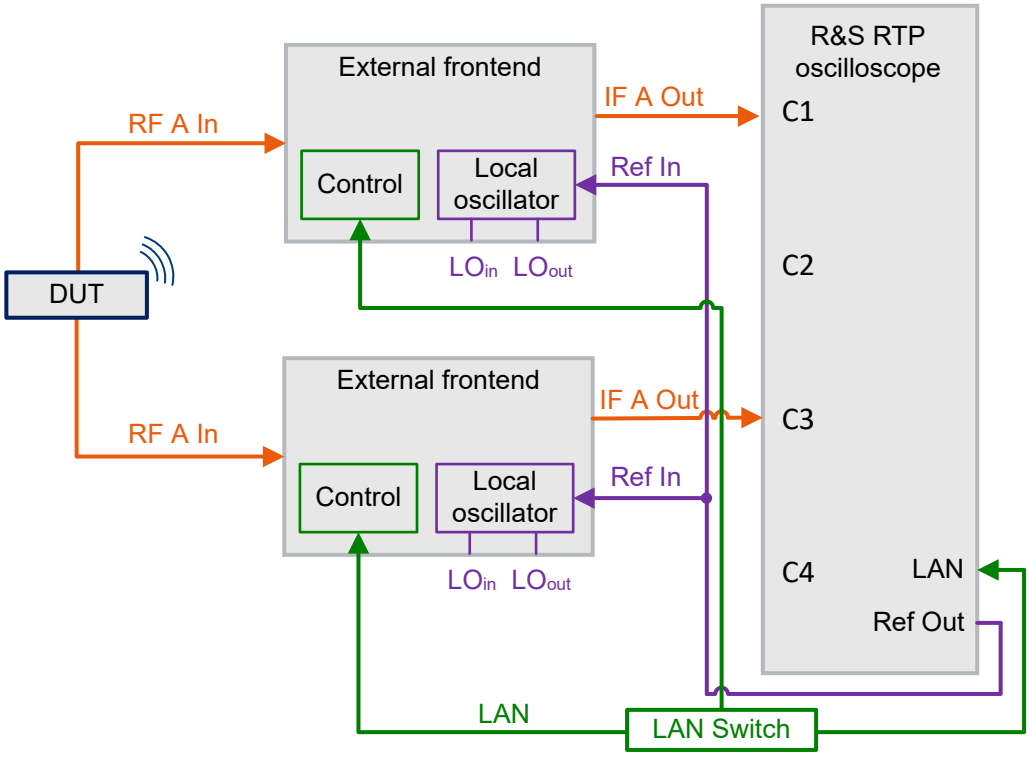



*Figure 21-3: Test setup for analysis of 2 channels with 2 external frontends and coupled reference frequency*

### Coupled local oscillators

In test setups with coupled local oscillators, the LO output of one frontend is connected to the LO input of the next frontend. The first frontend receives the reference signal from the oscilloscope. Coupling local oscillators results in minimum phase noise in the setup.

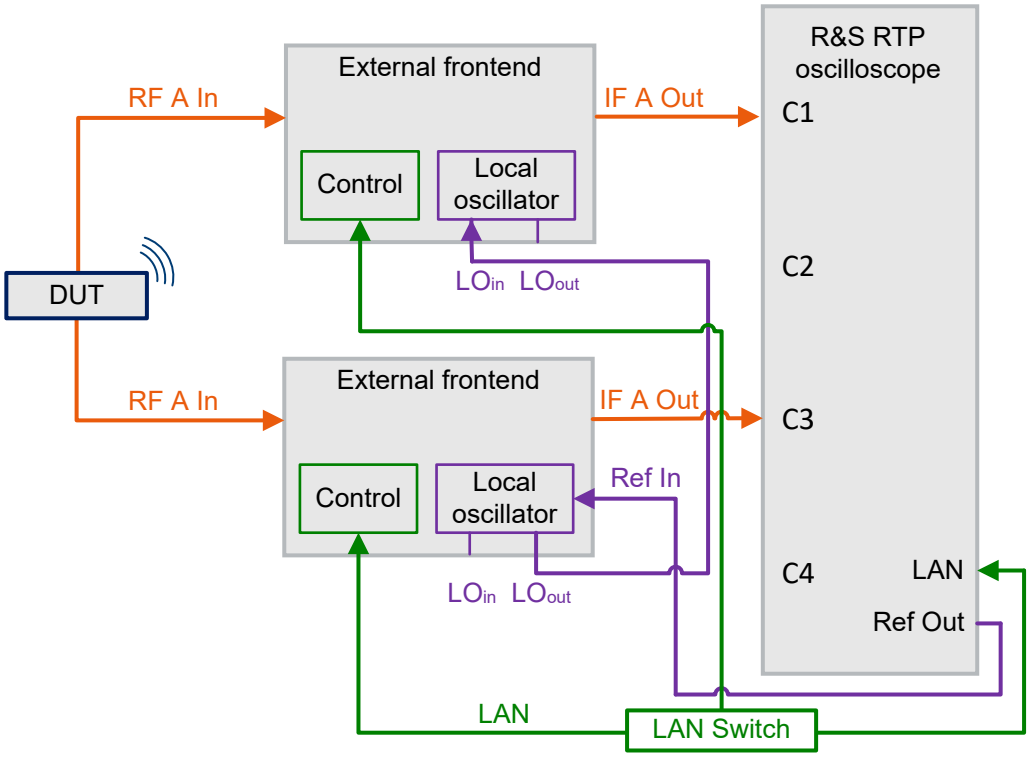



*Figure 21-4: Test setup for analysis of 2 channels with 2 external frontends and coupled local oscillators*

The test setup with 4 channels and 2 dual-channel frontends is suitable for 4x4 MIMO tests. The local oscillators are connected in this setup.

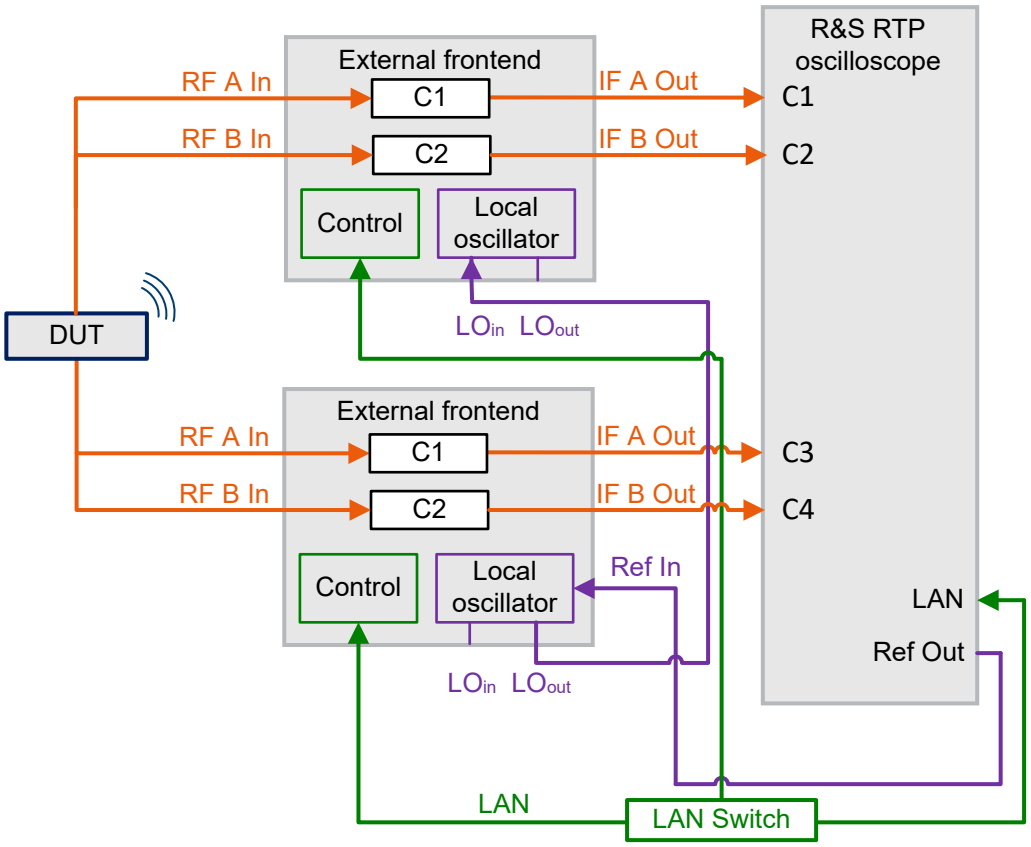



*Figure 21-5: Test setup for analysis of 4 channels with 2 external frontends and coupled local oscillators*

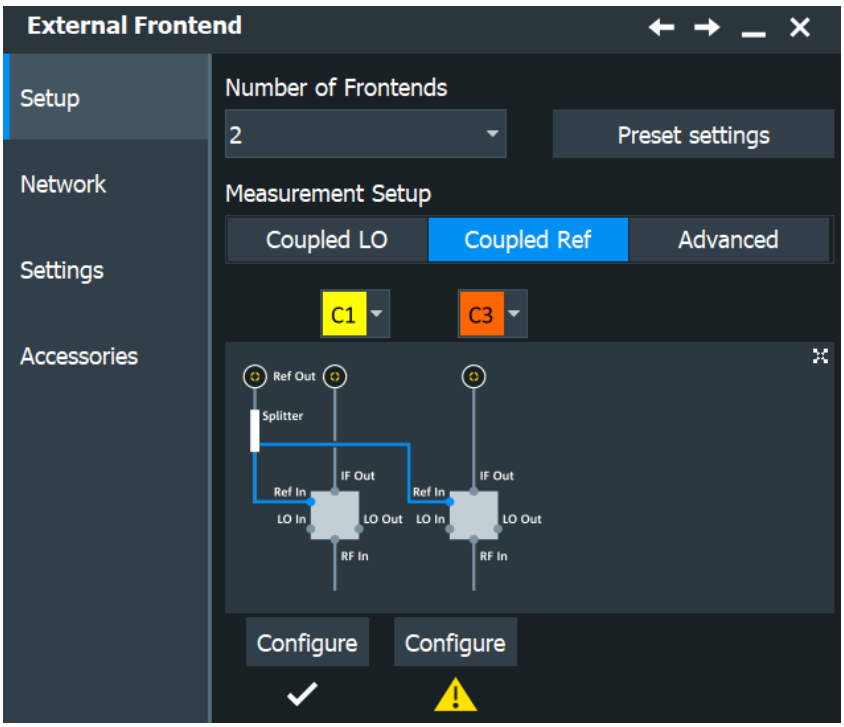
## 21.3 Settings for frontend control

Access: "Menu" > "Apps" > "RF" tab > "External Frontend"


- Setup ..... 1107
- Network ..... 1109
- Settings ..... 1111
- Accessories ..... 1114


### 21.3.1 Setup

Access: "Menu" > "Apps" > "RF" tab > "External Frontend" > "Setup"




Number of Frontends....1108
Preset settings....1108
Measurement Setup....1108
Configure....1109

**Number of Frontends**
Sets the number of connected frontends.

Depending on the frontend type and oscilloscope bandwidth, up to 4 frontends can be connected.

- R&S FE44S and R&S FE50DTR: max. 4 external frontends
- R&S FE170SR: max. 2 frontends, requires R&S RTP with 13 GHz or 16 GHz bandwidth

Remote command:
`EFRontend:COUNt` on page 2446

**Preset settings**
Resets the complete configuration of the external frontends to the default values. External frontends have their own preset, the general preset does not affect the frontend settings.

Remote command:
`EFRontend:PRESet` on page 2447

**Measurement Setup**
Selects the test setup and the input channels to which the frontends are connected.

- "Coupled LO": local oscillators of the frontends are connected.

- "Coupled Ref": the reference frequency "Ref In" for each frontend comes from the oscilloscope.
- "Advanced": user-defined test setup.

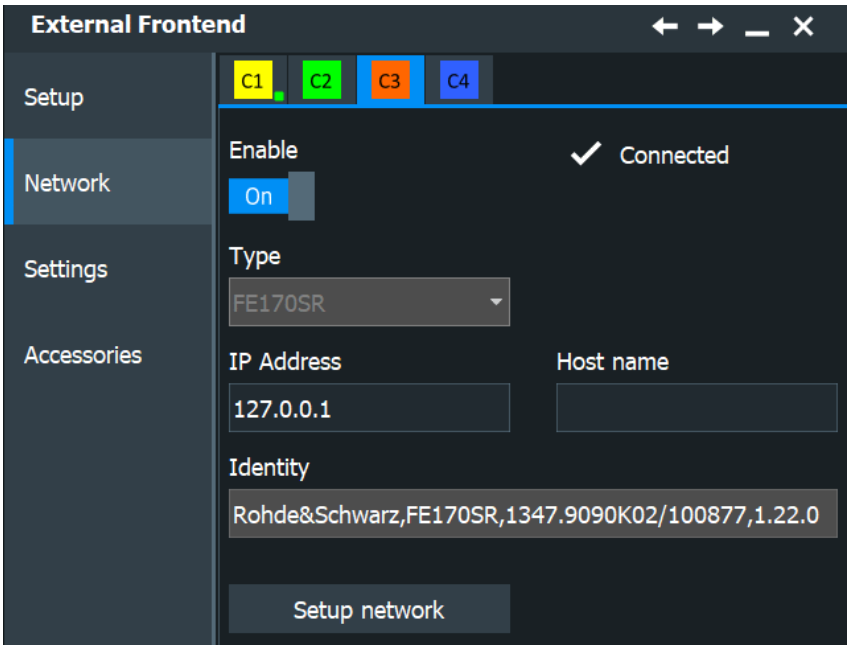
See also: Chapter 21.2, "Test setups", on page 1104.

Remote command:
`EFRontend:SETup` on page 2447
`EFRontend:CSELect<m>` on page 2446

**Configure**
Opens the network configuration settings for the connected channel.

## 21.3.2 Network

Access: "Menu" > "Apps" > "RF" tab > "External Frontend" > "Network" > select channel




Enable ..... 1109
Type ..... 1110
IP Address ..... 1110
Host name ..... 1110
Identity ..... 1110
Setup network ..... 1110

**Enable**
Activates the frontend on the selected channel.

Remote command:
`EFRontend<m>:CONNection:STATe` on page 2446
`EFRontend<m>:CONNection:CSTate?` on page 2445

**Type**
Selects the type of the connected frontend, for example, R&S FE44S, R&S FE50DTR or R&S FE170SR.

Remote command:
`EFRontend<m>:TYPE:LIST?` on page 2447
`EFRontend<m>:CONNection:CONFig` on page 2445

**IP Address**
Enter the IP address of the connected frontend. The frontend must be connected to the R&S RTP via LAN. The IP address is indicated on the electronic ink display on the frontend. The IP address is alternative to host name, you can enter the one or the other.

Remote command:
`EFRontend<m>:CONNection:CONFig` on page 2445

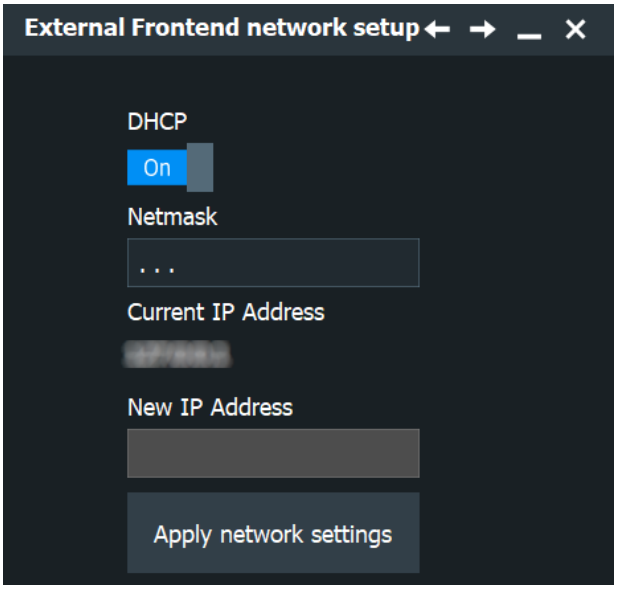
**Host name**
Enter the name of the connected frontend. The host name is alternative to IP address, you can enter the one or the other.

Remote command:
`EFRontend<m>:CONNection:CONFig` on page 2445

**Identity**
Shows the IDN string of the connected frontend, which contains the frontend type, part number, serial number and version.

Remote command:
`EFRontend<m>:IDN?` on page 2446

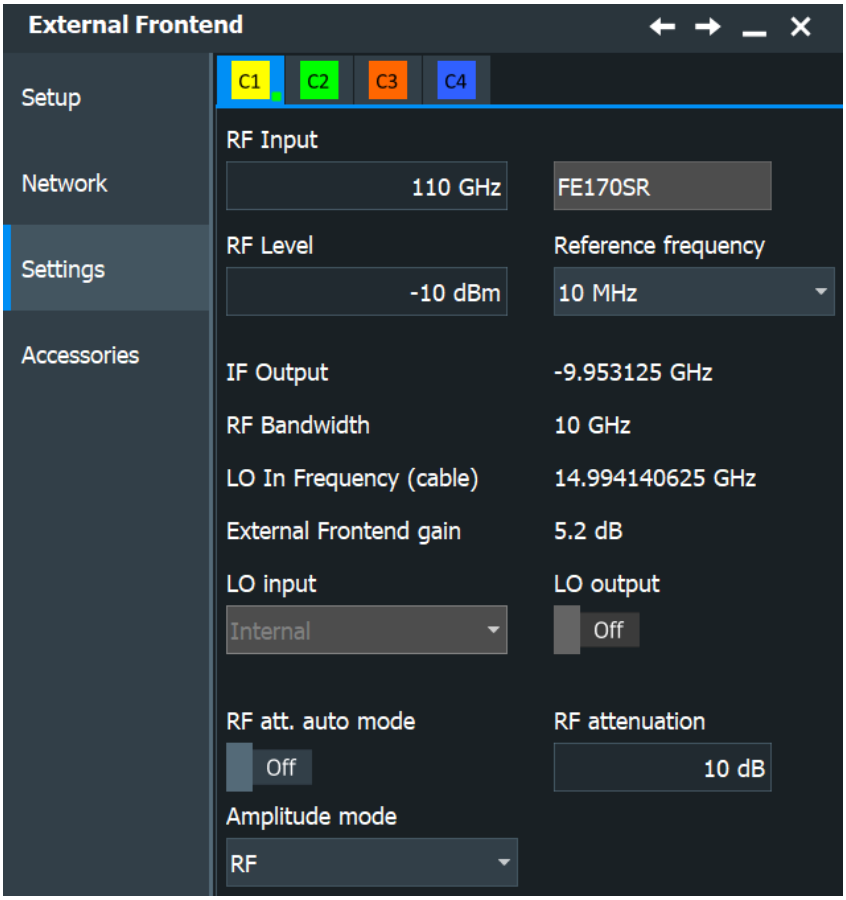
**Setup network**
Opens a dialog to define specific network settings, in particular, if DHCP is not applicable:



| | |
|---|---|
| "DHCP" | Switch off DHCP for manual IP configuration of the external frontend. Contact your network administrator to obtain a valid IP address. The automatic configuration mode ( DHCP = ON) uses DHCP server to obtain the instrument IP address. |
| "Netmask" | Subnet mask of the network. |
| "New IP address" | IP address of the external frontend. |
| "Apply network settings" | Activates the settings. |

### 21.3.3 Settings

Access: "Menu" > "Apps" > "RF" tab > "External Frontend" > "Settings"




The selected frontend type is shown for information. To change the frontend, select the "Network" tab.

RF Input ..... 1112
RF Level ..... 1112
Reference frequency ..... 1112
IF Output ..... 1112
RF Bandwidth ..... 1112

LO In Frequency (cable)....1112
External Frontend gain....1113
LO input, LO output....1113
RF attenuation auto mode, RF attenuation....1113
Amplitude mode....1113

**RF Input**
Sets the RF frequency of the signal that the external frontend receives from the DUT. The value is the center frequency that determines the LO frequency generation.

The instrument suggests a value that is supported by the connected external frontend.

Remote command:
`EFRontend<m>:FREQuency:RFRequency:VALue` on page 2458

**RF Level**
Sets the reference level of the signal at the RF input of the external frontend. If this level is entered correctly, the external frontend is configured such that the level at the input of the R&S RTP is optimal.

Remote command:
`EFRontend<m>:RFINput:LEVel` on page 2459

**Reference frequency**
Sets the reference frequency that is used for frequency conversion on the external frontend.

In "Coupled LO" and "Coupled Ref" measurement setup, the reference frequency is the frequency of the Ref Out signal of the oscilloscope, 10 MHz.

In "Advanced" setup, also other sources of the reference frequency are possible, e.g. from a signal analyzer. You can select the value from a list of reference frequencies that are supported by the connected external frontend.

Remote command:
`EFRontend<m>:FREQuency:REFerence[:VALue]` on page 2456

**IF Output**
Shows the intermediate frequency to which the signal is downconverted by the frontend.

Remote command:
`EFRontend<m>:FREQuency:IFRequency:VALue?` on page 2453

**RF Bandwidth**
Shows the maximum analysis bandwidth of the connected external frontend.

Remote command:
`EFRontend<m>:FREQuency:IFBWidth:LIMit?` on page 2453

**LO In Frequency (cable)**
In "Coupled LO" mode, the local oscillators of the external frontends are connected with a cable from "LO Out" of one frontend to "LO In" of the next frontend. The field displays the frequency of the signal that goes through this cable, which is between 8 GHz and 16.4 GHz.

Remote command:
EFRontend<m>:FREQuency:LOSCillator:INPut:FREQuency? on page 2454

**External Frontend gain**
Displays the gain of the external frontend for information.

Remote command:
EFRontend<m>:GAIN? on page 2458

**LO input, LO output**
For measurement setups "Coupled LO" and "Coupled Ref", the status of the LO input and LO output is shown. The usage is defined by the setup definition.

If the "Advanced" measurement setup is selected, set "LO input" and "LO output" according to the connection of the local oscillator in your setup.

"LO input = Internal" The LO input of the selected frontend is NOT connected to the LO output of another frontend. The reference frequency is received via "Ref In" from the oscilloscope.

"LO input = External" The LO input of the selected frontend is connected to the LO output of another frontend.

Remote command:
EFRontend<m>:FREQuency:LOSCillator:MODE on page 2454
EFRontend<m>:FREQuency:LOSCillator:OUTPut:STATe on page 2455

**RF attenuation auto mode, RF attenuation**
The R&S RTP sets the signal level based solely on the "RF Level" setting. The RF attenuation settings are meant to protect the mixer of the external frontend from high RF input levels. Increasing the RF attenuation reduces intermodulation in the mixer, but increases noise because the signal needs reamplification afterwards.

If "RF attenuation auto mode" is "On", the instrument provides a good compromise between intermodulation and noise, and the optimal vertical scale is set. If "Off", you can fine-tune this trade-off by adjusting "RF attenuation".

Remote command:
EFRontend<m>:ATTenuation:AUTO on page 2449
EFRontend<m>:ATTenuation[:VALue] on page 2449
EFRontend<m>:ATTenuation:STEP:LIST? on page 2449

**Amplitude mode**
Defines the vertical scale of the signal.

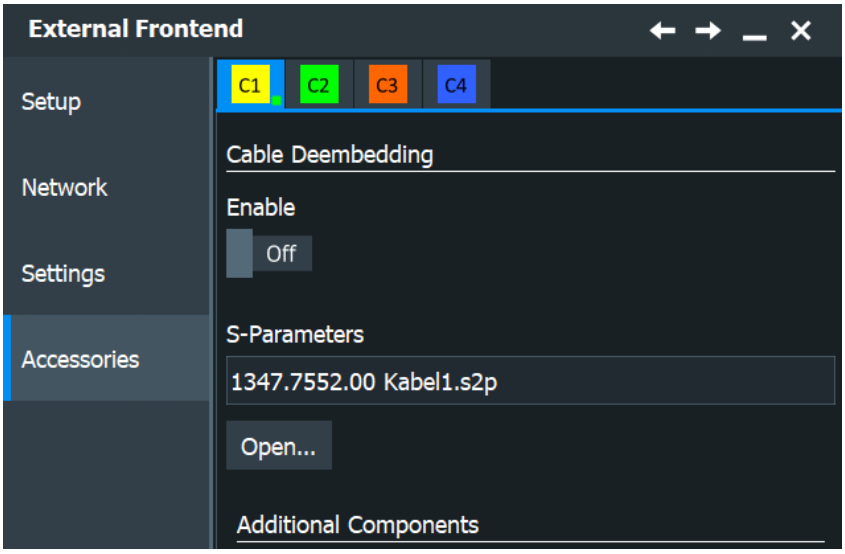
"RF" The vertical scale is calculated out of the specified "RF Level" and the "External Frontend gain" so that the signal uses the complete height of the diagram. Both values are visible in the dialog.

"IF" The vertical scale is set to 50 mV/div.

Remote command:
EFRontend<m>:AMPLmode on page 2448

### 21.3.4 Accessories

Access: "Menu" > "Apps" > "RF" tab > "External Frontend" > "Accessories"




Cable Deembedding: Enable....1114
S-Parameters....1114
Additional components....1114

**Cable Deembedding: Enable**
Activates the correction of the IF signal due to IF cable loss from the frontend to the oscilloscope.

Remote command:
`EFRontend<m>:ALIGn:STATe` on page 2460

**S-Parameters**
Defines the S-parameter file that contains the correction values for the cable that connects the frontend to the oscilloscope.

You can type the filename, or tap "Open" to select the path and file in a file selection dialog. The file is loaded automatically when cable deembedding is enabled.

Remote command:
`EFRontend<m>:ALIGn:FILE` on page 2460

**Additional components**
For future use.

## 21.4 Using frontends to get I/Q data

You can use external frontends with option R&S RTP-K553 together with option R&S RTP-K11 I/Q software interface. To analyze the data, you can use the R&S VSE software or another analysis software.

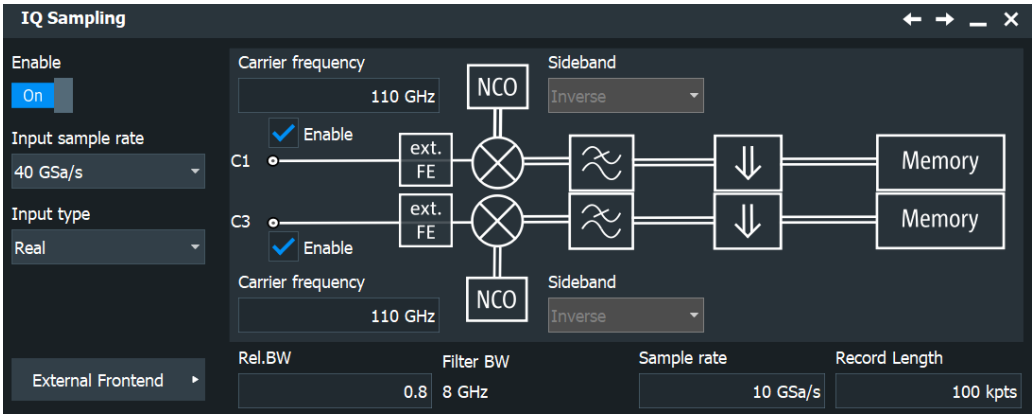
Typical use cases are:

- Analysis of RF chips with 2x2 polarization MIMO and bandwidth 10 GHz
- Analysis of RF chips with differential analog I/Q output and bandwidth 10 GHz with R&S VSE software

If both options are installed on the instrument, and the external frontends are configured and enabled, they are included in the signal path. The frequency response of the frontend is corrected automatically.

1. Start the R&S RTP-K11 I/Q application: [Apps] > "RF" tab > "IQ".
2. Select "External Frontend" to configure the frontends.
3. Set up and enable the external frontends.
4. Tap the "Back" arrow in the dialog title bar to return to the "IQ Sampling" dialog.

   You can see the active frontends in the signal path of I/Q data analysis.
5. For "Carrier frequency", enter the frequency of the DUT at the input of the external frontend.
6. Set the "Input type" = "Real". The complex input type is not supported when external frontends are used.




## 21.5 Using frontends to get raw data

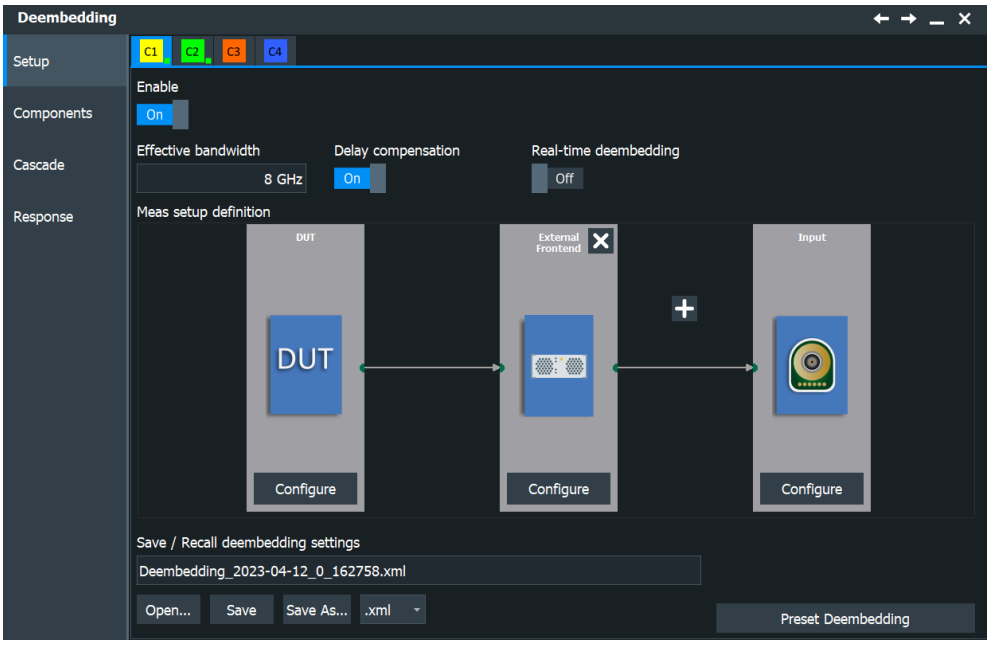
You can use external frontends with option R&S RTP-K553 together with deembedding option R&S RTP-K121. To analyze the acquired data, you can use the R&S VSE software. Real-time deembedding with option R&S RTP-K122 is *not* supported in this scenario.

Correcting the frequency response of the external frontends becomes more important if the bandwidth of the signal increases. Signals with higher bandwidth are used, for example, in various radar applications. In addition, the potential gain in having smaller files for baseband signals via I/Q application (R&S RTP-K553) diminishes when the

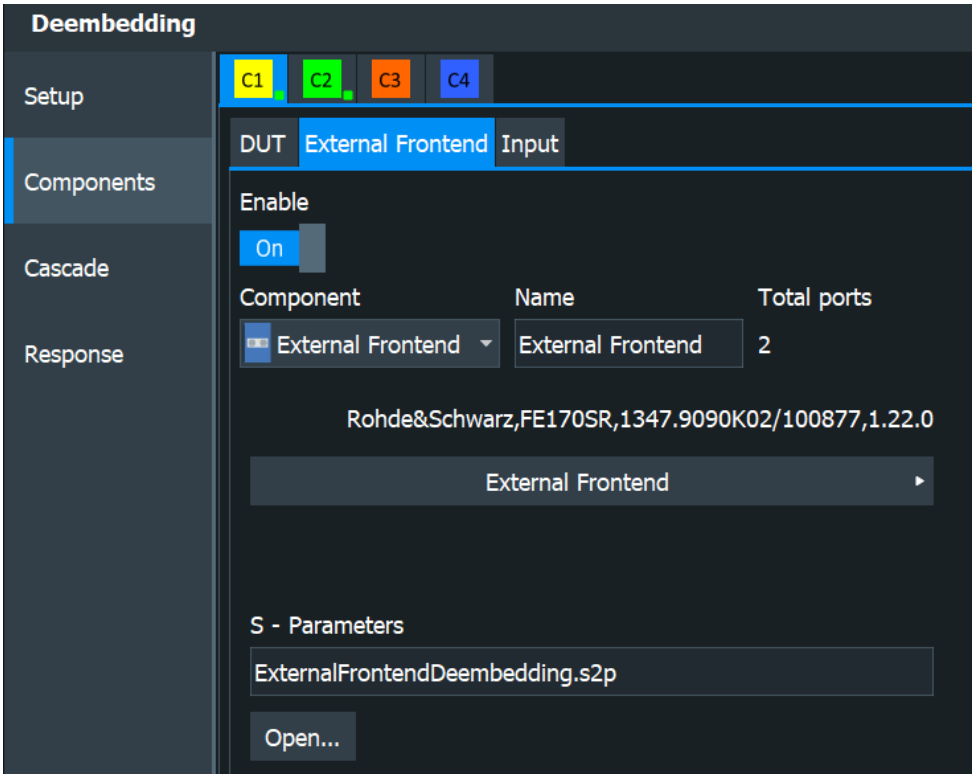
bandwidth increases. In such a scenario, a waveform-based analysis method on the de-embedded IF signal can be more suitable.

The external frontend is an additional component in the deembedding cascade. The S-parameters of the connected frontends are loaded, and the deembedding removes the parasitic effects of the component from the measured signal.

1. Start the R&S RTP-K121 deembedding application: [Apps] > "Analysis" tab > "Deembedding".
2. Select the input channel to which the external frontend is connected.
3. Under "Meas setup definition", tap the "+" icon.
4. Select the "External Frontend" component.




5. Tap "Configure".
6. If the external frontend is already connected to the channel, the identity of the frontend is displayed. If not, select "External Frontend" to configure the frontends.
7. Set up and enable the external frontends.
8. Tap the "Back" arrow in the dialog title bar to return to the "Components" dialog.




The R&S RTP automatically creates an S-parameter file from the correction data of the external frontend and compensates it. Do not change the "S-Parameters" setting.
As long as the configuration is incomplete or conflicting, the message "Invalid configuration" is shown.

9. Start the R&S VSE software, either directly on the oscilloscope if installed ([Apps] > "R&S Apps" tab > "VSE", or on a connected computer.
Refer to the R&S VSE documentation for details on configuration and analysis.

## 21.6 Troubleshooting frontend control

If problems arise during external frontend control, try the following measures to optimize the connection or setup. For additional information, see the external frontend documentation.

### 21.6.1 Error messages

If errors occur, status messages in the external frontend configuration dialog box and in the status bar of the display inform you about them.

Query the error message queue using the remote command `SYST:ERR?` to determine if any or which errors occurred during operation.

If any errors occur that are not described here, contact the Rohde & Schwarz customer support.

***Table 21-1: Status and error messages***

| Status bar message | Dialog status | SYST:ERR? result | Description, possible solution |
|---|---|---|---|
| External Frontend [x]: Connecting... | Connecting... | | Connection to the external frontend is being established.<br>If a frontend is connected for the first time, connecting takes about 30 sec. Repeated connection takes only 2 sec. |
| Connected | Connected | | Connection established, ready for operation. |
| No External Frontend Connected | Not connected | | No external frontend detected.<br>Connect an external frontend to the R&S RTP. |
| External Frontend [x]: Invalid Host IP | Invalid Host IP | `300, "Invalid Host IP"` | Verify the specified IP address of the external frontend. |
| External Frontend [x]: LAN Connection Failed | Not connected | `301, "LAN Connection Failed"` | Verify that the external frontend is connected to the same network as the R&S RTP.<br>Verify the specified IP address of the external frontend. |
| External Frontend [x]: Incompatible ??? Firmware | Incompatible Oscilloscope Firmware | `302, "Incompatible ??? Firmware"` | The frontend hardware is not compatible with the installed oscilloscope firmware.<br>Update the oscilloscope firmware, see the R&S RTP release notes. |
| External Frontend [x]: Incompatible Frontend Firmware | Incompatible Frontend Firmware | `303, "Incompatible Frontend Firmware"` | The firmware of the external frontend is not up to date.<br>Perform a firmware update. |
| External Frontend [x]: Device Locked | Device Locked | `304, "Device Locked"` | External frontend is accessible via ping, but no SSL-connection could be established.<br>Disconnect another device that is accessing the external frontend and try reconnecting.<br>A single frontend supports up to 2 LAN/SSL connections. |
| External Frontend [x]: Wrong Device Type | Wrong Device Type | `306, "Wrong Device Type"` | The connected external frontend is not of the specified type.<br>Disconnect the device, select the correct external frontend type in the configuration dialog box, and reconnect. |
| External Frontend [x]: LAN Connection Lost | LAN Connection Lost | `314, "LAN Connection Lost"` | The connection to the external frontend has been lost.<br>The R&S RTP attempts to reconnect periodically. Wait for the "Connecting…" message. |
| External Frontend [x]: Firmware Update Failed | Firmware Update Failed | `315, "Firmware Update Failed"` | The firmware of the external frontend was not updated successfully. |
| External Frontend [x]: <Frontend_Name> is not allowed, RF input frequency of at least <Max_RF> GHz is required. | Connected | `316, "External Frontend Not Allowed"` | The R&S RTP does not support any IF frequencies required by the external frontend. |

| Status bar message | Dialog status | SYST:ERR? result | Description, possible solution |
|---|---|---|---|
| External Frontend [x]: Current RF frequency is out of valid range. Please check Frontend Config dialog for more information | Connected | `317, "RF frequency is out of valid range."` | The selected measurement (RF) frequency is not supported by the selected frequency band of the external frontend. Select a different frequency on the R&S RTP, or a different frequency band for the frontend. |
| External Frontend [x]: Settings Mismatch - reinitiate frontend by changing a measurement parameter | Connected | `332, "Settings Mismatch"` | The conversion settings on the external frontend were changed by another instrument during the measurement on the R&S RTP. The measurement results no longer match the original measurement settings. To restore the original measurement settings on the frontend, change a frontend setting in the R&S RTP dialog. |
| External Frontend [x]: Not Connected – Device Busy | Device busy | `333, "Device Busy"` | The external frontend is blocked by another instrument, e.g. for initial connection or a firmware update. After the other instrument finishes its task, try reconnecting the R&S RTP. Switch the "Connection" state off and on again. |
| External Frontend [x]: Device Busy | Connected | `333, "Device Busy"` | During a running measurement, the external frontend is blocked by another instrument, e.g. for initial connection or a firmware update. The current measurement is aborted. After the other instrument finishes its task, start a new measurement on the R&S RTP. |
| External Frontend [x]: Touchstone file insufficient frequency range | | `350, "Touchstone file frequency range insufficient"` | The `.s2p` file does not cover the used frequency.<br>Correct the file or select a different correction file. |
| | "<filename>.s2p has wrong file format" | `-150, "String data error"` | The `.s2p` file does not have the correct syntax.<br>Correct the file. |
| | | `-256, "File name not found"` | The specified `.s2p` file does not exist.<br>Correct the path and filename and make sure that the file is stored on the instrument under `C:\Users\Public\Public Documents\Rohde-Schwarz\RTx\ExternalFrontends\<frontend_type>\touchstonefiles`. |
| External Frontend [x]: Touchstone file insufficient frequency range | | `350, "Touchstone file frequency range insufficient"` | The `.s2p` file does not cover the used frequency.<br>Correct the file or select a different correction file. |

**Touchstone file errors**

Some errors concerning `.s2p` correction files do not cause warnings in the status bar and do not provide results in the system error queue, but do display messages or return errors in remote control.

## 21.6.2 Problems with connection

If you cannot connect to the external frontend, check the status bar or error queue of the R&S RTP for error messages, see also Table 21-1.

If no messages are available, check the "LAN" and "POWER" status LEDs on the external frontend. Permanent red lights indicate an error, usually hardware problems, e.g. if the fans are off. In this case, the instrument is automatically put in a safe (error) state to avoid damage. A LAN connection is no longer possible. Try switching the external frontend off and on again.

If the problem persists, contact Rohde & Schwarz customer support, see Chapter 25.3, "Contacting customer support", on page 2476.

### 21.6.3 Problems during operation

**Problem: When using the R&S FE170SR together with the I/Q software interface (R&S RTP-K11), the received signal has aliasing or has an unexpected shape**
**Possible solution:** The R&S FE170SR uses an intermediate frequency close to 10 GHz. Make sure that the "Input sample rate" in the I/Q software option is set to 40 GSa/s, and that no warning is shown. Make sure that the R&S FE170SR is connected to channel 1 or channel 3 of the R&S RTP, and that there is no signal on channels 2 and 4.

# 22 Compliance tests

Thanks to the high measurement accuracy of the R&S RTP, the oscilloscope can be used for reliable compliance testing on the physical layer of high-speed bus interfaces. For fast and easy compliance testing, you need the free-of-charge R&S ScopeSuite. The software guides you step-by-step through the test sequences, configures the oscilloscope, automatically performs measurements and compiles the results in a measurement report. Detailed, image-based instructions make it easy to connect the oscilloscope and the probes to the test fixture set and the device under test.

The R&S ScopeSuite can be installed on a test computer or directly on the R&S RTP. If the R&S ScopeSuite is installed on the oscilloscope, you can start it directly in the firmware menu.

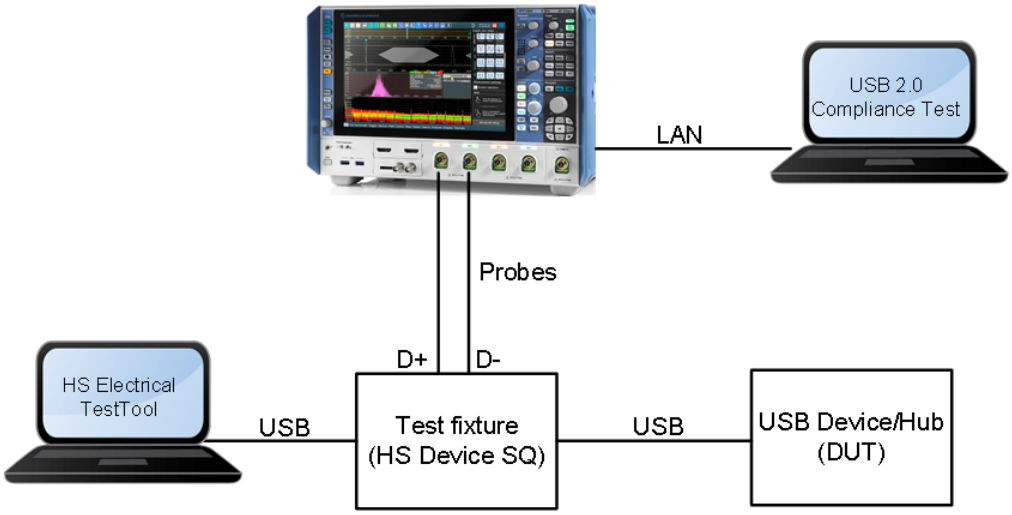
For best operation results, we recommend that the installed firmware versions of the R&S ScopeSuite and the oscilloscope are the same.

Access: [Apps] > "Compliance" tab

# 22.1 USB compliance test

The R&S RTP-K21 option enables the automatic testing of USB 2.0 compliance (high-speed) as well as USB 1.1 (full-speed) and USB 1.0 (low-speed). Additionally you can perform USB 3.2 compliance tests with option R&S RTP-K101.

As an example, the test setup for USB device tests is shown below.




The test fixture and the USB compliance test procedures are described in separate manuals:

- "R&S®RTO-K21, R&S®RTP-K21/-101 R&S®USB 2.0/3.2 Compliance Test, User Manual"
- "R&S®RT-ZF1 USB 2.0 Compliance Test Fixture Set, Manual"
- "R&S®RT-ZB2 USB 3.1 Demo Board 1 (Compliance Mode), Manual"

## 22.1.1 Test equipment

**USB 2.0 test equipment**

For USB 2.0 compliance tests, the following test equipment is recommended:

- R&S RTP oscilloscope with 4 channels and at least 2 GHz bandwidth.
- R&S RT-ZF1 USB 2.0 compliance test fixture set, which consists of the load board and the signal quality board.
- Probes:
  - 2x single-ended active probes with at least 2 GHz bandwidth
  - 1x differential active probe with at least 2 GHz bandwidth
- R&S RTO-B6/R&S RTP-B6/Tabor WX2182B/Tabor WX2182C arbitrary waveform generator for automatic receiver sensitivity tests.
  For manual receiver sensitivity tests, any AWG can be used.

- R&S ScopeSuite software, which can be installed on a computer or directly on the R&S RTP.
- R&S RTP-K21 USB 2.0 compliance test option (required option, installed on the oscilloscope)
- Make sure that you use the latest version of the USBHSET tool. To find downloads, search for *USBHSET* on the following websites:
  - USB 3.0 host: www.usb.org/developers/tools/
  - USB 2.0 host: www.usb.org/developers/tools/usb20_tools/

  Regularly check the USB-IF website compliance.usb.org/index.asp?UpdateFile=Electrical&Format=Standard for USB-IF compliance updates.

**USB 3.2 Test Equipment**

For USB 3.2 compliance tests, the following test equipment is recommended:

- R&S RTP oscilloscope with 4 channels and at least 16 GHz bandwidth.
- At least one of the following test fixture sets:
  - R&S RT-ZB2 USB 3.1 Demo Board 1
  - USB3ET - USB 3.0 Electrical Test Fixture Kit
  - USB31AET – USB 3.1 USB (10 GT/s) Type-A and Micro-B Electrical Test Fixture Kit
  - USB31CET – USB 3.1 USB (10 GT/s) Type-C Electrical Test Fixture Kit
- Cables / adapters:
  - 2x R&S RT-ZA16 BNC to SMA adapter
  - 2x SMA cable
- R&S RTP-B6/R&S RTO-B6/R&S RTP-B6/Tabor WX2182B/Tabor WX2182C arbitrary waveform generator to generate `Ping.LFPS` to toggle compliance pattern.
- R&S ScopeSuite software, which can be installed on a computer or directly on the R&S RTP).
- R&S RTP-K101 USB 3.2 compliance test option (required option, installed on the R&S RTP)

**USB3.2-Rx Test Equipment**

For USB 3.2-Rx compliance tests, the following test equipment is recommended:

- R&S RTP oscilloscope with 4 channels and at least 16 GHz bandwidth.
- At least one of the following test fixture sets:
  - USB 3.1 Compliance Load Board (5.6”, 7.1”, 8.1”)
  - USB 3.1 Mock Host (7.2”, 8.7”, 9.7”)
  - USB 3.1 Mock Device (7.2”, 8.7”, 9.7”)
  - Host PHY Tx Fixture USB 3.1
  - Dev PHY Tx Fixture USB 3.1
  - USB 3.1 Host Fixture 1A
  - USB 3.1 Dev Fixture 1A
  - Rx CAL Fixture

  - USB 3.1 Captive Cable Device Fixture Type-A
  - USB 3.1 Captive Cable Device Fixture Type-C
- Cables / adapters:
  - 3 m Type A - B cable
  - 1 m (6dB) cable
  - 2 m (7dB) cable
  - 2 Pair of matched SMA cables
  - Short A-B or A-uB Cable
  - 2 R&S RT-ZA16 adapters
- R&S RTP-B6/R&S RTO-B6/R&S RTP-B6/Tabor WX2182B/Tabor WX2182C arbitrary waveform generator to generate `Ping.LFPS` to toggle compliance pattern.
- Bit error rate tester (BERT): Anritsu MP1900A
- Anritsu MX183000A: high-speed serial data test software
- R&S ScopeSuite software, which can be installed on a computer or directly on the R&S RTP).
- R&S RTP-K102 USB 3.2-Rx compliance test option (required option, installed on the R&S RTP)

## 22.2 Ethernet compliance tests (options R&S RTP-K22/K23/K25/K86)

The Ethernet compliance test options enable the automatic testing of various Ethernet standards:

- 10/100/1000BASE-T (R&S RTP-K22)
- 2.5/5/10GBASE-T (R&S RTP-K23)
- 100BASE-T1: (R&S RTP-K24)
- 1000BASE-T1: (R&S RTP-K87)
- 1000BASE-T1: (R&S RTP-K88
- 10BASE-T1: (R&S RTP-K89)

The test fixture and the Ethernet compliance test procedures are described in the separate manuals:

- R&S®RTO, R&S®RTP Ethernet Compliance Test, User Manual"
- "R&S®RT-ZF2 Ethernet Compliance Test Fixture Set Manual"
- "R&S®RT-ZF3 Frequency Converter Manual
- "R&S®RT-ZF4 10BASE-Te Fixture Manual
- R&S®RT-ZF5 Ethernet Probing Fixture Manual
- R&S®RT-ZF6 Frequency Converter Manual

### 22.2.1 Test equipment

For Ethernet compliance tests, the following test equipment is needed:

- The free-of-charge R&S ScopeSuite software, which can be installed on a computer or directly on the oscilloscope.
- **10/100/1000BASE-T Ethernet compliance tests:**
  - R&S RTO/R&S RTP oscilloscope with at least 600 MHz bandwidth
  - Differential probe with at least 1 GHz bandwidth
  - R&S RTO/R&S RTP-K22 10/100/1000BASE-T Ethernet compliance test option (required option, installed on the oscilloscope)
  - R&S RT-ZF2 Ethernet test fixture set
  - For energy-efficient Ethernet tests, in addition:
    R&S RT-ZF5 Ethernet test fixture set for 100/1000BASE-T EEE tests
    R&S RT-ZF4 test fixture for 10BASE-Te tests
- **2.5GBASE-T Ethernet compliance tests:**
  - R&S RTO/R&S RTP oscilloscope with at least 600 MHz bandwidth
    For precise power spectral density and power level measurements up to 1 GHz, it is recommended to use an oscilloscope and a differential probe with 1 GHz bandwidth each.
  - Differential probe with at least 1GHz bandwidth
  - R&S RTO/R&S RTP-K23 GBASE-T Ethernet compliance test option (required option, installed on the R&S RTP)
  - R&S RT-ZF2 Ethernet test fixture set
- **5GBASE-T Ethernet compliance tests:**
  - R&S RTO/R&S RTP oscilloscope with at least 1 GHz bandwidth
    For precise power spectral density and power level measurements up to 1.5 GHz, it is recommended to use an oscilloscope and a differential probe with 1.5 GHz bandwidth each.
  - Differential probe with at least 1.5 GHz bandwidth
  - R&S RTO/R&S RTP-K23 GBASE-T Ethernet compliance test option (required option, installed on the R&S RTP)
  - R&S RT-ZF2 Ethernet test fixture set
- **10GBASE-T Ethernet compliance tests:**
  - R&S RTO/R&S RTP oscilloscope with at least 2 GHz bandwidth
    For precise power spectral density and power level measurements up to 3 GHz, it is recommended to use an oscilloscope and a differential probe with 3 GHz bandwidth each.
  - Differential probe with at least 3 GHz bandwidth
  - R&S RTO/R&S RTP-K23 GBASE-T Ethernet compliance test option (required option, installed on the R&S RTP)
  - R&S RT-ZF2 Ethernet test fixture set
- **100BASE-T1 compliance tests:**
  - R&S RTO/R&S RTP oscilloscope with at least 600 MHz bandwidth

  - Differential probe with 1 GHz bandwidth
  - R&S RTO/R&S RTP-K24 100BASE-T1 compliance test option (required option, installed on the R&S RTP)
  - R&S RT-ZF2 Ethernet test fixture set
  - R&S RT-ZF3 frequency converter for transmitter distortion tests
- **1000BASE-T1 compliance tests:**
  - R&S RTO/R&S RTP oscilloscope with at least 2 GHz bandwidth
  - Differential probe with 2 GHz bandwidth
  - R&S RTO/R&S RTP-K87 1000BASE-T1 compliance test option (required option, installed on the R&S RTP)
  - R&S RT-ZF2 Ethernet test fixture set
  - R&S RT-ZF6 frequency converter for transmitter distortion tests
- **MGBASE-T1 compliance tests:**
  - R&S RTO / R&S RTO6 /R&S RTP oscilloscope, with a bandwidth of:
    - for 2.5 GBASE-T1 tests: at least 3.5 GHz
    - for 5 GBASE-T1 tests: at least 7 GHz
    - for 10 GBASE-T1 tests: at least 14 GHz
  - 2 BNC/SMA cables
  - R&S RTO/R&S RTP-K88 MGBASE-T1 compliance test option (required option, installed on the R&S RTP)
- **10BASE-T1 compliance tests:**
  - R&S RTO/R&S RTP oscilloscope
  - Optional Differential probe with 1 GHz bandwidth
  - R&S RTO/R&S RTP-K89 10BASE-T1 compliance test option (required option, installed on the R&S RTP)
- R&S RTO-B6/R&S RTP-B6/Tabor WX2182B/Tabor WX2182C arbitrary waveform generator for automatic disturber tests. For some disturber tests, also HMF2550 can be used.
- R&S ZNB/ZNC/ZND/ZVL vector network analyzer for automatic return loss measurements. For manual measurements, also other AWGs and VNAs can be used.

## 22.3 D-PHY compliance tests (option R&S RTP-K26)

The option R&S RTP-K26 enables the automatic processing of D-PHY compliance tests.

The D-PHY compliance test procedures are described in a separate manual:

- R&S®RTO-K26, R&S®RTP-K26, D-PHY Compliance Test, User Manual"

### 22.3.1 Test equipment

For D-PHY compliance tests, the following test equipment is needed:

- R&S RTP oscilloscope with 4 channels and at least 4 GHz bandwidth
- For measuring the clock signal (+ and -): either 1 differential probe or 2 single-ended probes with at least 4 GHz bandwidth. However, note that D-PHY Group 2 and Group 4 tests require 2 probes for the clock signal.
- For measuring the data signal (+ and -): 2 probes with at least 4 GHz bandwidth
- R&S RTP-K26 D-PHY compliance test option (required option, installed on the oscilloscope)
- R&S RTP-K27 D-PHY compliance test option (required option, installed on the oscilloscope)
- For Eye Tests, the following options are needed:

  For R&S RTP:
  - Option R&S RTP-K136 or R&S RTP-K137 Advanced eye analysis
  - OptionR&S RTP-K140/K141 High-speed serial pattern trigger
- Recommended test fixture for LP-TX tests: MIPI D-PHY Capacitive Load ($C_{LOAD}$) fixture from The University of New Hampshire InterOperability Laboratory (UNH-IOL)
- Recommended termination board for HS-TX tests: MIPI D-PHY Reference Termination Board (RTB) from The University of New Hampshire InterOperability Laboratory (UNH-IOL)
- The free-of-charge R&S ScopeSuite software, which can be installed on a computer or directly on the oscilloscope.

## 22.4 eMMC compliance tests (option R&S RTP-K92)

The option R&S RTP-K92 enables the automatic processing of eMMC compliance tests.

The eMMC compliance test procedures are described in a separate manual:

- eMMC Compliance Tests, Test Procedures Manual"

### 22.4.1 Test equipment

For eMMC compliance tests, the following test equipment is needed:

- R&S RTP oscilloscope with at least 1 GHz bandwidth
- A probe with at least 1.5 GHz bandwidth
- R&S RTP-K92 eMMC compliance test option (required option, installed on the R&S RTP)
- The free-of-charge R&S ScopeSuite software, which can be installed on a computer or directly on the R&S RTP.

## 22.5 PCIe compliance tests (option R&S RTP-K81)

The option R&S RTP-K81 enables the automatic processing of PCIe 1.1 / PCIe 2.0 compliance tests.

Also, option R&S RTP K83 enables the automatic processing of PCIe 3.0 compliance tests.

The PCIe compliance test procedures are described in a separate manual:

- R&S®RTO-K81, R&S®RTP-K81/-K83, PCIe Compliance Test, User Manual"

### 22.5.1 Test equipment

**PCIe 1.1 / PCIe 2.0 Test Equipment**

For PCIe compliance tests, the following test equipment is needed:

- R&S RTP oscilloscope with at least 6 GHz bandwidth, 2 channels and 20 GS/s available
- Probes/cables:
  - Two SMP to SMA cables or
  - Two single-ended probes with at least 6GHz bandwidth bandwidth or
  - One differential probe with at least 6GHz bandwidth bandwidth
- R&S RTP-K81 PCIe compliance test option (required option, installed on the oscilloscope)
- Recommended test fixture:
  - For Add-In card testing: PCI Express Compliance Base Board (CBB2.0) from PCI-SIG
  - For System Board testing: PCI Express Compliance Load Board (CLB2.0) from PCI-SIG
- The free-of-charge R&S ScopeSuite software, which can be installed on a computer or directly on the oscilloscope.

**PCIe 3.0 Test Equipment**

For PCIe compliance tests, the following test equipment is needed:

- R&S RTP oscilloscope with at least 16 GHz bandwidth, 2 channels and 40 GS/s available
- Probes/cables:
  - Two SMP to SMA cables or
  - Two single-ended probes with at least 6GHz bandwidth bandwidth or
  - One differential probe with at least 6GHz bandwidth bandwidth
  - Two R&S RTP-ZMA40 Amplifier probe tips.
- R&S RTP -K83 PCIe compliance test option (required option, installed on the oscilloscope)

- Recommended test fixture:
  - For Add-In card testing: PCI Express Compliance Base Board (CBB3.0) from PCI-SIG
  - For System Board testing: PCI Express Compliance Load Board (CLB3.0) from PCI-SIG
- The free-of-charge R&S ScopeSuite software, which can be installed on a computer or directly on the oscilloscope.

## 22.6 DDR3 compliance tests (option R&S RTP-K91)

The option R&S RTP-K91 enables the automatic processing of DDR3 compliance tests.

The DDR3 compliance test procedures are described in a separate manual:

- R&S®RTO-K91, R&S®RTP-K91 DDR3 Compliance Test, User Manual"

### 22.6.1 Test equipment

For DDR3 compliance tests, the following test equipment is needed:

- R&S RTP with 4 channels and minimum 8 GHz bandwidth, or
  R&S RTO2000 with 4 channels and minimum 4 GHz bandwidth
  R&S RTO6 with 4 channels and minimum 4 GHz bandwidth
- R&S RTP-K91 DDR3 compliance test option (required option, installed on the R&S RTP)
- 4 modular probes, R&S ZM90 with 9 GHz bandwidth. See also Chapter 22.6.1.1, "Soldering guide for modular probes", on page 1129.
- 4 modular probe tips R&S RT-ZMA10
- The free-of-charge R&S ScopeSuite software, which can be installed on a computer or directly on the R&S RTP.

#### 22.6.1.1 Soldering guide for modular probes

All single-ended signals such as ADD, CMD, DQ, DM, CS, CKE should be soldered in such a way that the signal is connected to the + part on the probe tip. The ground of the signal is connected to the - part on the same probe tip.

All differential signals such as CK, DQS should be soldered differentially:

- The + signal is connected to the + part on the probe tip.
- The - signal is connected to the - part on the same probe tip.
- The ground of the signal is connected to the ground of the same probe tip.

## 22.7 DDR4 compliance tests (option R&S RTP-K93)

The option option R&S RTP-K93 enables the automatic processing of DDR4 compliance tests.

The DDR4 compliance test procedures are described in a separate manual:

- R&S®RTP-K93 DDR4 Compliance Test, User Manual"

### 22.7.1 Test equipment

For DDR4 compliance tests, the following test equipment is needed:

- R&S RTP with 4 channels and minimum 8 GHz bandwidth
- R&S RTP-K93 DDR4 compliance test option (required option, installed on the R&S RTP)
- 4 modular probes, R&S ZM90 with 9 GHz bandwidth. See also Chapter 22.7.1.1, "Soldering guide for modular probes", on page 1130.
- 4 modular probe tips R&S RT-ZMA10
- The free-of-charge R&S ScopeSuite software, which can be installed on a computer or directly on the R&S RTP.

#### 22.7.1.1 Soldering guide for modular probes

All single-ended signals such as ADD, CMD, DQ, DM, CS, CKE should be soldered so that the signal is connected to the + part on the probe tip. The ground of the signal should be connected to the - part on the same probe tip.

All differential signals such as CK, DQS should be soldered differentially:

- The + signal is connected to the + part on the probe tip.
- The - signal is connected to the - part on the same probe tip.
- The ground of the signal is connected to the ground of the same probe tip.

## 22.8 C-PHY compliance tests (option R&S RTP-K28)

The option R&S RTP-K28 enables the automatic processing of C-PHY compliance tests.

The C-PHY compliance test procedures are described in a separate manual:

- R&S®RTP-K28, C-PHY Compliance Test, User Manual"

### 22.8.1 Test equipment

For C-PHY compliance tests, the following test equipment is needed:

- R&S RTP oscilloscope with 4 channels and at least 13 GHz bandwidth

- The free-of-charge R&S ScopeSuite software, which can be installed on a computer or directly on the oscilloscope.
- Differential probes
- Three singled-ended active probes or SMA cables.
- R&S RTP-K28 C-PHY compliance test option (required option, installed on the oscilloscope)
- Recommended test fixture for LP-TX tests: MIPI C-PHY Capacitive Load ($C_{LOAD}$) fixture
- Recommended termination board for HS-TX tests: MIPI C-PHY Reference Termination Board (RTB)

# 23 Network operation and remote control

The operating system is the basis of the instrument's firmware, it provides basic functions such as logon, password protection, virus protection, and connection to a network.

In addition to working with the R&S RTP directly, using the touchscreen and the keys, you can also operate the instrument from a remote PC. Various methods of remote control are supported:

- Using the web interface in a LAN network
- Using Remote Desktop Connection in a LAN network
- Remote control with SCPI commands


- Operating system ..... 1132
- Setting up a network (LAN) connection ..... 1137
- Web interface ..... 1142
- Remote Desktop connection ..... 1147
- Remote control ..... 1149


## 23.1 Operating system

The R&S RTP has a Windows 10 operating system.

The operating system has been configured according to the instrument's features and needs. To ensure that the instrument software functions properly, certain rules must be observed when using the operating system.

**NOTICE**

**Risk of rendering instrument unusable**

The instrument is equipped with the Windows operating system. You can install additional software on the instrument, however, additional software can impair instrument function. Thus, run only programs that Rohde & Schwarz has tested for compatibility with the instrument software.

The drivers and programs used on the instrument under Windows are adapted to the instrument. Only install update software released by Rohde & Schwarz to modify existing instrument software.

Changes in the system setup are only required if the network configuration does not comply with the default settings (see Chapter 23.2.1, "Connecting the instrument to the network", on page 1137).

### 23.1.1 Logon

Windows requires that users identify themselves by entering a user name and password in a logon window.

If the instrument is connected to the network, you are automatically logged on to the network when you log on to the operating system. As a prerequisite, the user name and the password must be identical under Windows and on the network. The instrument provides an auto-logon function that can be configured for standard user and administrator access. The configuration requires the user name and password.

See also: "Log on as" on page 74.

The R&S RTP provides two user accounts:

- "Instrument": an administrator account with unrestricted access to the computer/ domain (default user)
- "NormalUser": a standard user account with limited access

For both users, the initial password is "894129". It is recommended that you change the password for both users after initial logon. You can change the password in the Windows configuration :

"Start" menu > "Settings" > "Accounts" > "Sign-in options".

Make sure to change the password also in the auto-logon of the R&S RTP, see below.

By default, the instrument logs on with administrator account and standard password. To restrict the access to system functions, you can change the autologon to "NormalUser".

The following tasks require administrator rights:

- Configuration of a LAN settings, network connection, and firewall settings
- Date and time setup
- Printer installation
- Firmware update
- BIOS update
- Using the web browser
- Touchscreen calibration
- Installation of options
- Moving portable licenses

**To configure the auto-logon for a standard user**

Default situation: the auto-logon is configured for the administrator ("Instrument").

1. Open "Menu" > "Settings" > "System" > "Users"
2. Set "Logon as" to "User autologon".
3. Enter the "User name": *NormalUser*.
4. Enter the password of the standard user.
5. Restart the instrument.

**To configure the auto-logon for administrator**

Starting situation: the auto-logon is configured for a standard user ("NormalUser").

1. Open "Menu" > "Settings" > "System" > "Users"
2. Set "Logon as" to "None".
3. Exit the firmware.
4. Sign out from the operating system.
5. Log on as administrator ("Instrument").
6. Set the "Logon as" to "Admin autologon". Enter the "User name": *Instrument* and the administrator's password.
7. Restart the instrument.

### 23.1.2 Service packs and updates

Microsoft regularly creates security updates and other patches to protect Windows-based operating systems. These are released through the Microsoft Update website and associated update server. Instruments using Windows, especially those that connect to a network, should be updated regularly.

For details and recommendations, see the following Rohde & Schwarz white paper:

- 1EF96: Malware Protection Windows 10

### 23.1.3 Virus protection

Take appropriate steps to protect your instruments from infection. Use strong firewall settings and scan any removable storage device used with a Rohde & Schwarz instrument regularly. It is also recommended that you install anti-virus software on the instrument. Rohde & Schwarz does NOT recommend running anti-virus software in the background ("on-access" mode) on Windows-based instruments, due to potentially degrading instrument performance. However, Rohde & Schwarz does recommend running it during non-critical hours.

For details and recommendations, see the following Rohde & Schwarz white paper:

- 1EF96: Malware Protection Windows 10

### 23.1.4 Backup and recovery

The R&S RTP has a backup partition with an application for backup and restore the firmware. The backup of the factory default state is always available. You can save additional backups, for example, before a firmware update or to provide different system configurations for different environments.

The backup saves the current instrument installation and its configuration. Data are not saved in the backup.

**To save a system backup**

1. Before starting the backup, unplug the LAN cable and USB storage devices to minimize the risk of computer virus attacks.
2. Restart the R&S RTP.

   The boot screen is displayed for about 2 seconds.
3. When the boot screen is displayed, immediately select "Backup", and press [ENTER].
4. Select "Create Backup".
5. Enter a name for the backup and the date. If necessary, you can add information to the description. By default, the description contains the current version of the firmware.
6. Select "Start Backup".

   A progress information dialog is displayed. You can terminate at any time with "Cancel". After the process has been finished, the dialog is closed automatically, and the main dialog is displayed again.
7. In the main dialog, select "Exit and Reboot".

   The backup and restore application is closed, and the R&S RTP is restarted.

**To restore a backup version**

When restoring a backup, the Windows operating system and installed anti-malware software are probably outdated. To minimize the risk of malware threats after restoring a backup, verify and adjust the "Windows Update" settings. Follow the recommendations from Rohde & Schwarz applicable to your instrument. Also, install all Windows security updates that have been published in the meanwhile.

1. Before starting the recovery, unplug the LAN cable and USB storage devices to minimize the risk of computer virus attacks.
2. Restart the R&S RTP.

   The boot screen is displayed for about 2 seconds.
3. When the boot screen is displayed, immediately select "Backup", and press [ENTER].
4. Select the backup that you want to restore.
5. Select "Restore Selected".
6. Confirm with "Yes".

A progress information dialog is displayed. You can terminate at any time with "Cancel".

During the restore process, the system partition is deleted, formatted and written newly. The data partition is not affected. After the process has been finished, the application is closed automatically, and the R&S RTP is restarted.

**To delete a backup version**

To provide space for new backups, you can remove older backups. The factory default cannot be deleted.

1. Restart the R&S RTP.
2. When the boot screen is displayed, immediately select "Backup", and press [ENTER].
3. Select the backup that you want to delete.
4. Select "Delete Selected".
5. Confirm with "Yes".
6. In the main dialog, select "Exit and Reboot", or continue with "To save a system backup" on page 1135.

### 23.1.5 Accessing windows functionality

All required Windows settings can be changed using the touchscreen and the on-screen keyboard that is part of the Windows system. However, modification is easier if you connect a mouse and/or keyboard to the instrument.

**To access Windows**

► Open "Menu" > 🖳.

The application is minimized to the taskbar and the "Start" menu becomes available.

**To access Windows using an external keyboard**

1. To open the "Start" menu, press the Windows key or the CTRL + ESC key combination on your keyboard.
2. To access the desktop, press the Windows key + D on your keyboard.

## 23.2 Setting up a network (LAN) connection

**Network environment**

Before connecting the product to a local area network (LAN), consider the following:

- Install the latest firmware to reduce security risks.
- For internet or remote access, use secured connections if applicable.
- Ensure that the network settings comply with the security policies of your company. Contact your local system administrator or IT department before connecting your product to your company LAN.
- When connected to the LAN, the product may potentially be accessed from the internet, which may be a security risk. For example, attackers might misuse or damage the product.

**Applications of LAN**

The LAN connection settings can be configured directly in the Windows operating system. Provided the network administrator has assigned you the appropriate rights and adapted the Windows firewall configuration, you can use the interface, for example:

- To transfer data between a controlling device and the test device, e.g. to run a remote control program.
- To access or control the measurement from a remote computer using the "Remote Desktop" application (or a similar tool)
- To transfer data from a remote computer and back, e.g. using network folders

This section describes how to configure the LAN interface. It includes the following topics:

- Chapter 23.2.1, "Connecting the instrument to the network", on page 1137
- Chapter 23.2.2, "Assigning the IP address", on page 1138

Note that only user accounts with administrator rights can configure LAN networks.

### 23.2.1 Connecting the instrument to the network

There are two methods to establish a LAN connection to the instrument:

- A non-dedicated network (Ethernet) connection from the instrument to an existing network.
- A dedicated network connection (Point-to-point connection) between the instrument and a single computer.

► To establish a non-dedicated network connection, connect a commercial RJ-45 cable to one of the LAN ports.
To establish a dedicated connection, connect a (crossover) RJ-45 cable between the instrument and a single PC.

If the instrument is connected to the LAN, Windows automatically detects the network connection and activates the required drivers.

The network card can be operated with a 10/100/1000 Mbps Ethernet IEEE 802.3u interface.

### 23.2.2 Assigning the IP address

Depending on the network capacities, the TCP/IP address information for the instrument can be obtained in different ways.

- If the network supports dynamic TCP/IP configuration using the Dynamic Host Configuration Protocol (DHCP), all address information can be assigned automatically.
- If the network does not support DHCP, or if the instrument is set to use alternate TCP/IP configuration, the addresses must be set manually.

By default, the instrument is configured to use dynamic TCP/IP configuration and obtain all address information automatically. This means that it is safe to establish a physical connection to the LAN without any previous instrument configuration.

**NOTICE**

**Risk of network errors**

Connection errors can affect the entire network. If your network does not support DHCP, or if you choose to disable dynamic TCP/IP configuration, you must assign valid address information before connecting the instrument to the LAN. Contact your network administrator to obtain a valid IP address.

**Assigning the IP address on the instrument**

1. Open "Menu" > "Settings".
2. Select the "System" tab.
3. In the "Network" tab, select "Advanced".
4. Select "Ethernet".
5. Select "Properties".
6. On the "Networking" tab, select "Internet Protocol Version 4 (TCP/IPv4)", and then select "Properties".
7. Select "Use the following IP address".
8. Enter the address information as obtained from the network administrator.

### 23.2.3 Using computer names

In a LAN that uses a DNS server (Domain Name System server), each PC or instrument in the LAN can be accessed via an unambiguous device name (or host name, computer name) instead of the IP address. The DNS server translates the host name to the IP address. Device names are useful when a DHCP server is used, as a new IP address can be assigned each time the instrument is restarted.

Each instrument is delivered with an assigned computer name, but this name can be changed.

The default instrument name is a non-case-sensitive string with the following syntax:

<Type><variant>-<serial_number>

The serial number can be found on the rear panel of the instrument. It is the third part of the device ID printed on the bar code sticker:

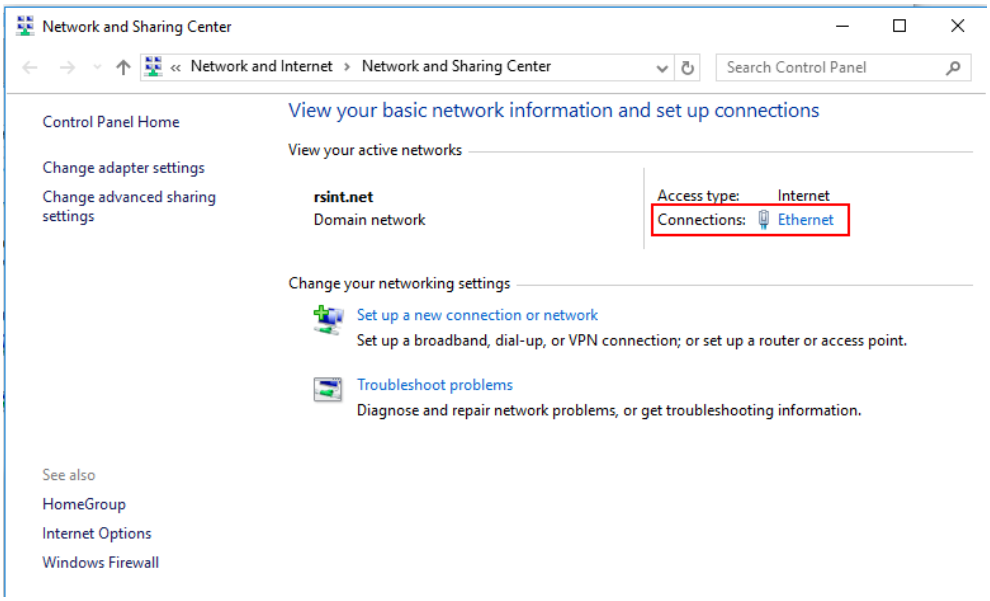
**To change the device name**

1. Open the "Menu" > "Settings" dialog.
2. Open the "System" dialog.
3. Select the "Network" tab.
4. Enter the new "Device name" and confirm the entry.

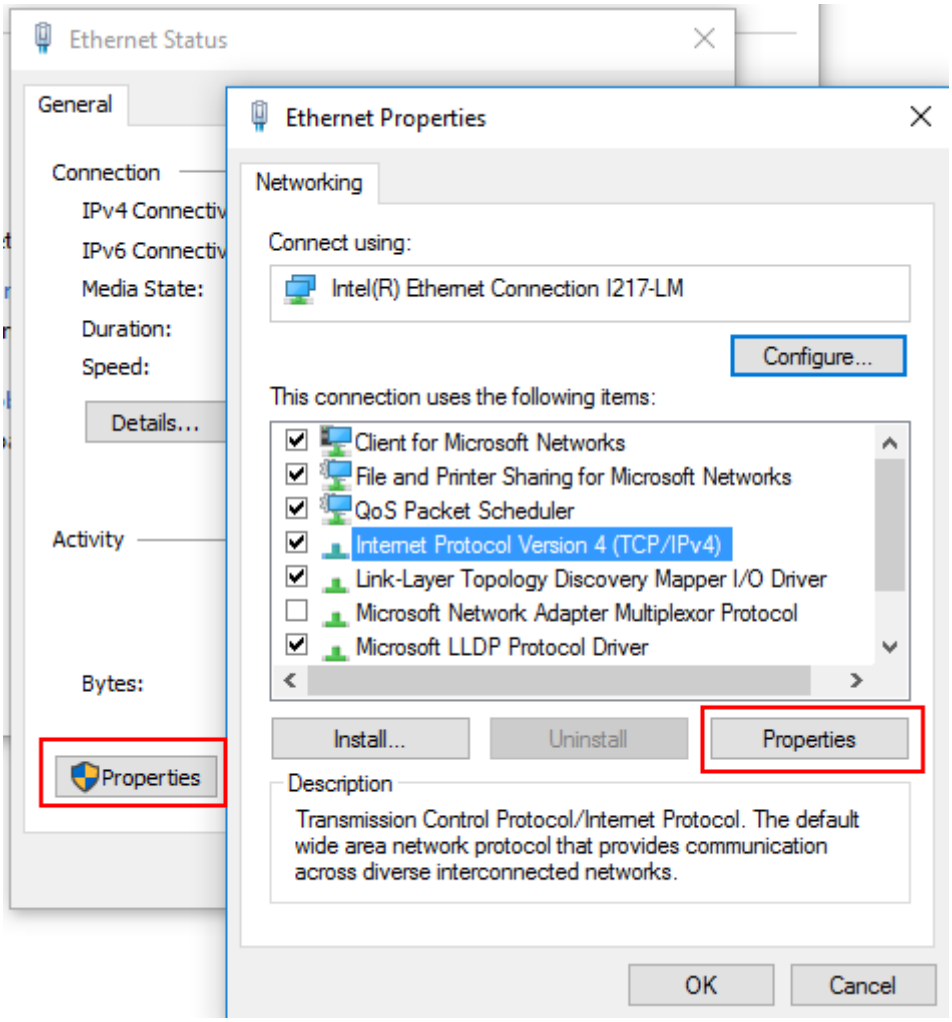
**Using a DNS server to determine the IP address**

If a DNS server is configured on the R&S RTP, the server can determine the current IP address for the connection using the permanent computer name.

1. Obtain the name of your DNS domain and the IP addresses of the DNS and WINS servers on your network.
2. Open the "Menu" > "Settings" dialog.
3. Select the "System" tab.
4. Tap "Network".
5. Select "Ethernet".




6. On the "Networking" tab, select "Internet Protocol Version 4 (TCP/IPv4)", and then select "Properties".




7. On the "General" tab, select "Use the following DNS server addresses" and enter your own DNS addresses.

   For more information, refer to the Windows operating system help.

### 23.2.4 Changing the windows firewall settings

A firewall protects an instrument by preventing unauthorized users from gaining access to it through a network. Rohde & Schwarz highly recommends the use of the firewall on your instrument. Rohde & Schwarz instruments are shipped with the Windows firewall enabled and preconfigured in such a way that all ports and connections for remote control are enabled.

For more details on firewall configuration, see the following Rohde & Schwarz White Paper:

- 1EF96: Malware Protection Windows 10

Note that changing firewall settings requires administrator rights.

## 23.3 Web interface

If the R&S RTP is connected to a computer via LAN, you can operate the instrument from a computer. No additional tools are required, you need only a web browser.

### 23.3.1 Settings on the R&S RTP

Access: "Menu" > "Settings" > "System" > "Network"

The LAN status icon on the toolbar of the R&S RTP indicates the status of the LAN connection. A green icon indicates that the instrument is connected to the LAN; a red symbol indicates an error - mostly the LAN cable is not connected.

The "Network" tab of the "System" dialog box provides network information. See Chapter 4.1.2, "Network settings", on page 71.

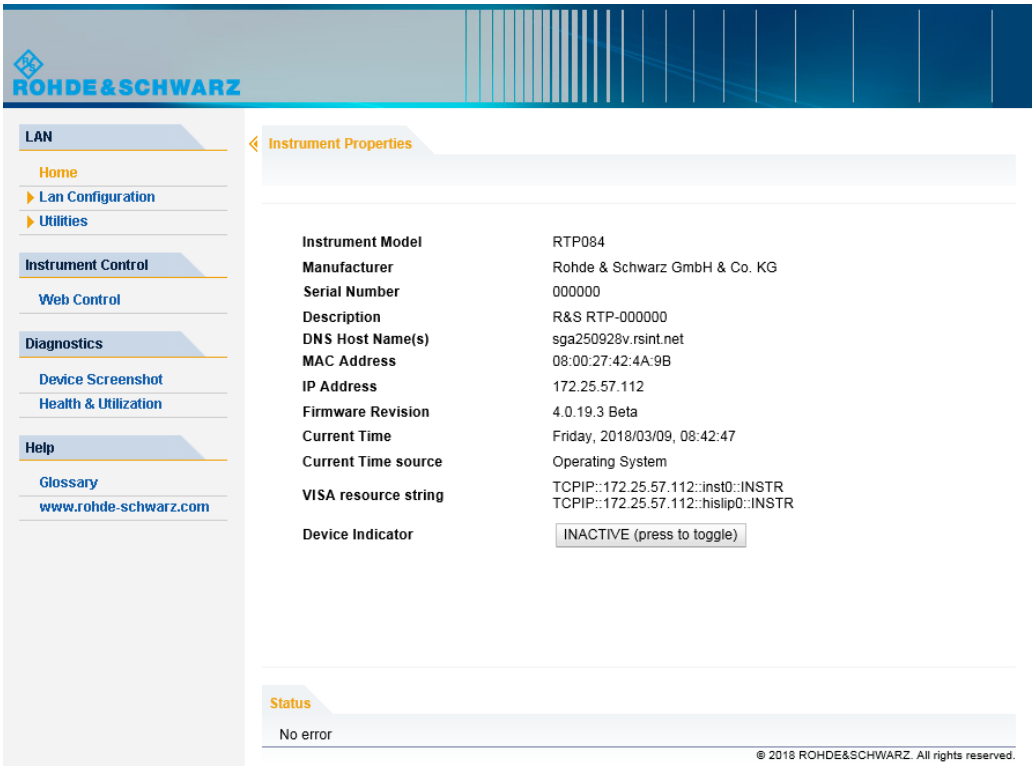
Only users with administrator rights can change LAN settings.

### 23.3.2 Web browser

The instrument's web interface works with all W3C compliant browsers. Only users with administrator rights on the instrument can use the web interface.

1. Open a web browser on the computer.
2. Type the instrument's host name or IP address in the address field of the browser on your PC, e.g. "http://10.113.10.203".

   The instrument home page opens.




The instrument home page displays the device information required by the LAN standard including the VISA resource string in read-only format.

► To update the "Host Name", press the "Device Indicator" button. The "Device Indicator" is not password-protected.

The most important items in the navigation menu of the browser interface are the following:

- "LAN Configuration" opens the menu with configuration pages.
- "Utilities > Logging" shows log messages and provides buttons to download or clear the logfile.
- "Web Control" emulates the front panel and shows the instrument display. You see a live image of the instrument, and you can operate the instrument remotely. You can use the keys, the knobs and the menus in the same way as directly on the instrument.
- "Device Screenshot": creates a screenshot of the display.
- "Health & Utilization": not yet supported
- "Help" provides a glossary of terms related to the standard, and a link to the Rohde & Schwarz Internet site.

#### 23.3.2.1 LAN configuration

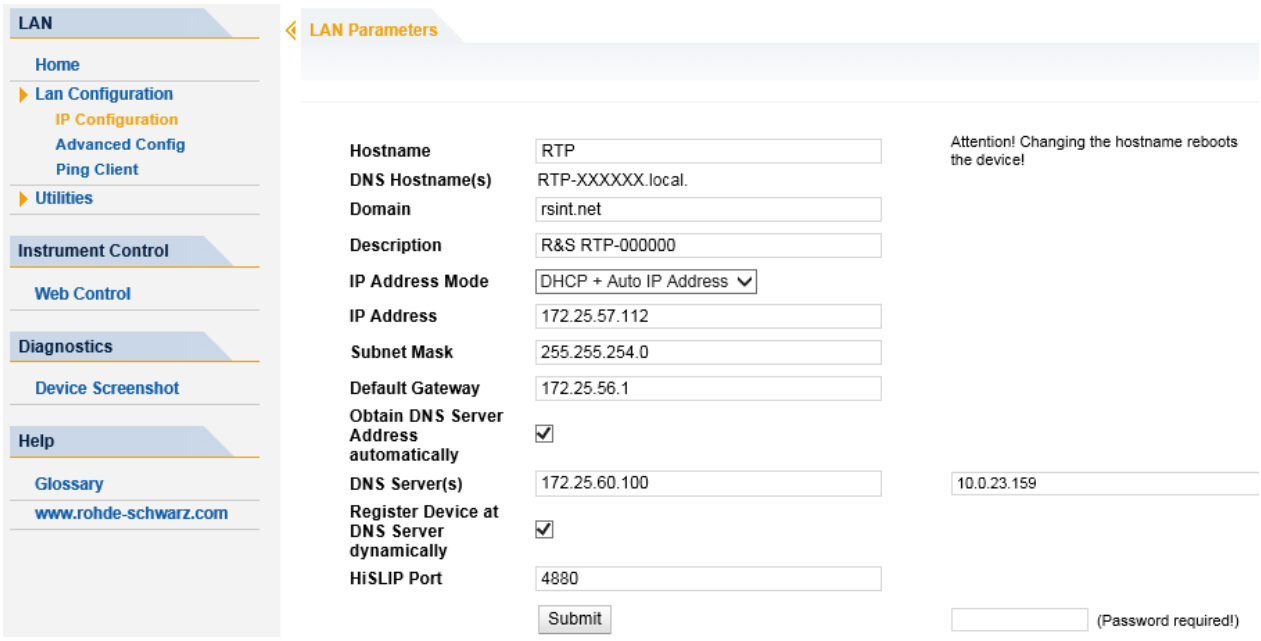
The LAN configuration consists of three parts:

- "IP configuration" provides all mandatory LAN parameters.

- "Advanced LAN Configuration" provides LAN settings that are not declared mandatory by the standard.
- "Ping Client" provides the ping utility to verify the connection between the instrument and other devices.

### IP configuration

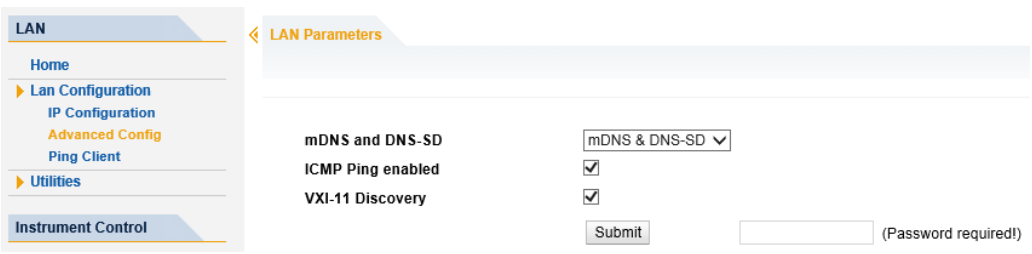
The "LAN Configuration > IP configuration" web page displays all mandatory LAN parameters and allows their modification.




The "TCP/IP Mode" configuration field controls how the IP address for the instrument gets assigned (see also Chapter 23.2.2, "Assigning the IP address", on page 1138). For the manual configuration mode, the static IP address, subnet mask, and default gateway are used to configure the LAN. The automatic configuration mode uses DHCP server or Dynamic Link Local Addressing (automatic IP) to obtain the instrument IP address.

Changing the LAN configuration is password protected. The password is *LxiWebIfc* (notice upper and lower case characters). This password cannot be changed in the current firmware version.

### Advanced config



The "LAN Configuration > Advanced Config" parameters are used as follows:

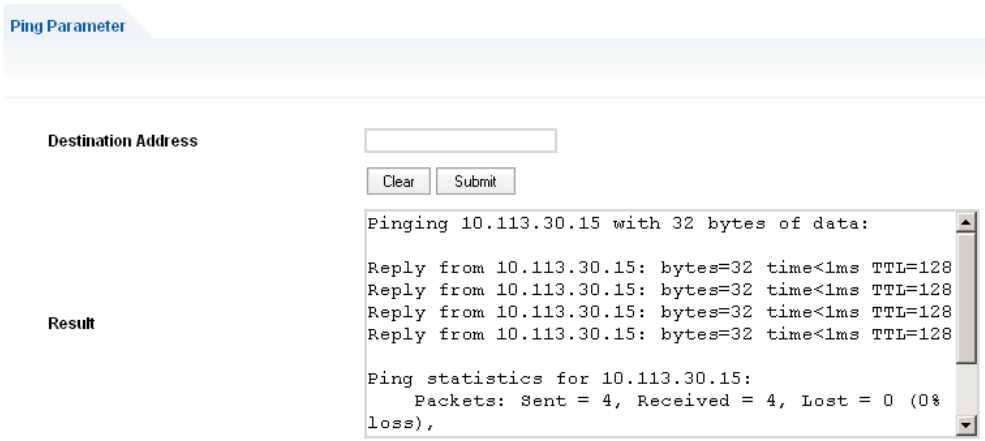
- mDNS and DNS-SD are two additional protocols: Multicast DNS and DNS Service Discovery. They are used for device communication in zero configuration networks working without DNS and DHCP
- "ICMP Ping" must be enabled to use the ping utility.
- "VXI-11" is the protocol that is used to detect the instrument in the LAN.

### Ping client

Ping is a utility that verifies the connection between the instrument and another device. The ping command uses the `ICMP` echo request and echo reply packets to determine whether the LAN connection is functional. Ping is useful for diagnosing IP network or router failures. The ping utility is not password-protected.

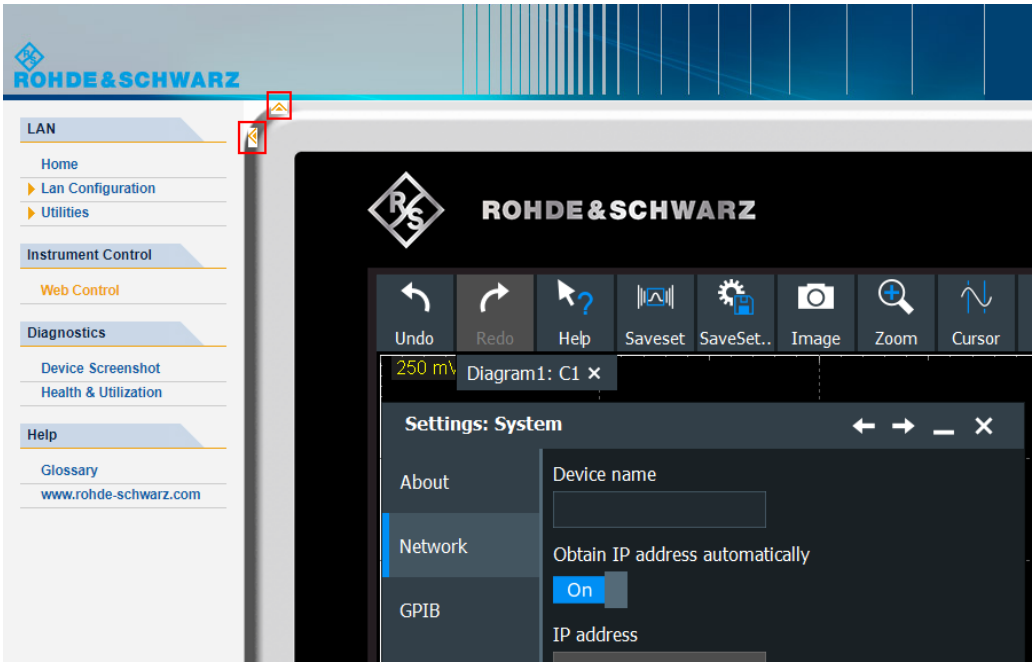
To initiate a ping between the compliant instrument and a second connected device:

1. Enable "ICMP Ping" on the "Advanced Config" page (enabled by default).
2. On the "Ping Client" page, enter the IP address of the second device **without the ping command and without any further parameters** into the "Destination Address" field (e.g. *10.113.10.203*).
3. Click "Submit".




## 23.3.2.2 Web control

"Web Control" emulates the front panel and shows the instrument display. You see a live image of the instrument, and you can operate the instrument remotely. You can use the keys, the knobs and the menus in the same way as directly on the instrument. The Web control replaces VNC as control tool for remote operation.

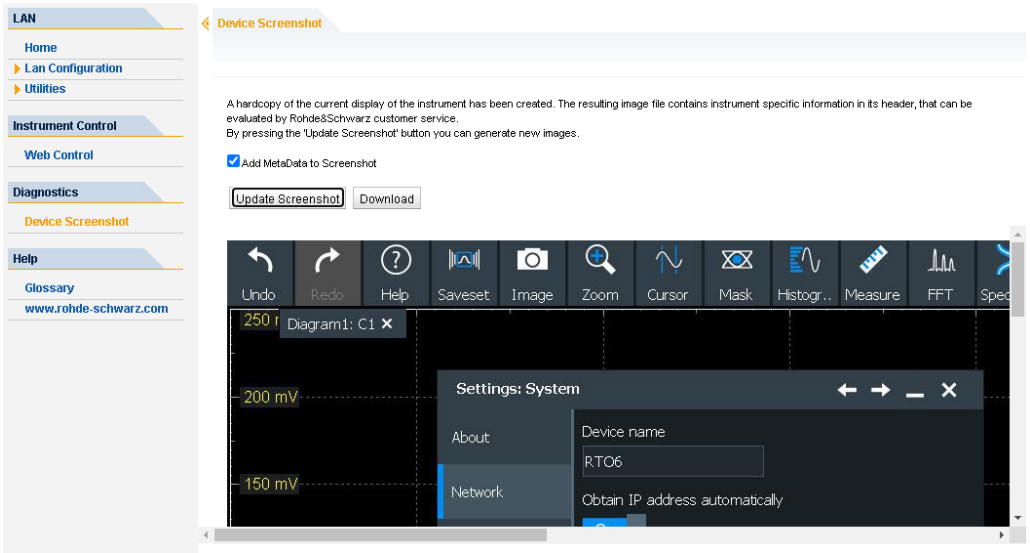



In the upper left corner of the emulated display, you see two arrows:

- The up/down arrow hides or shows the front panel.
- The left/right arrow hides or shows the menu.

### 23.3.2.3 Device screenshot

If you click "Device Screenshot", a screenshot of the current instrument display is shown on the computer.



**Add MetaData to Screenshot**
Adds the instrument information to PNG and JPG files. Meta information is saved as EXIF information and can be read, for example, using the ExifTool, see "Meta information in screenshots" on page 507.

**Update Screenshot**
Updates the display.

**Download**
Saves the screenshot to the download directory of the computer. By default, JPG format is saved.

To save the screenshot in PNG format, select "Save as", select "All files" as type, and enter the filename with extension *.png*.

## 23.4 Remote Desktop connection

Remote Desktop is a Windows application, which can be used to access and control the instrument from a remote computer through a LAN connection. While the instrument is in operation, the contents of the instrument screen are displayed on the remote computer. Remote Desktop provides access to all applications, files, and network resources of the instrument. Thus, remote operation of the instrument is possible.

**NOTICE**

**Risk of unauthorized access**

If you enable the Windows Remote Desktop application on the instrument, anyone using the network who knows the computer name and login data can access it. To prevent unauthorized access, make sure that the Remote Desktop application on the instrument is disabled: "Start" > "Settings" > "Control Panel" > " System"

**To set up a Remote Desktop connection**

1. Enable remote desktop control on the instrument. See "Enabling remote desktop control on the instrument" on page 1147 for details.
2. Connect the instrument and the remote computer to a LAN. See Chapter 23.2.1, "Connecting the instrument to the network", on page 1137 for details.
3. Set up the Remote Desktop connection between the remote computer and the instrument.

Remote Desktop Connection is part of the operating system and can be accessed via "Start > Windows Accessories > Remote Desktop Connection.".

**Enabling remote desktop control on the instrument**

1. Press the [Setup] key and select the "System" tab.

2. Tap "System".
3. Select "Remote settings".

   The "Remote" tab of the "System Properties" is shown.
4. Under "Remote Desktop", activate "Allow remote connections to this computer".

   **Note:** Remote Desktop access and firewall settings.
   When you enable or disable the Windows Remote Desktop option (in the "System Properties"), the associated firewall settings are adapted automatically.
5. If necessary, click "Select Users" and select users who are allowed to access the R&S RTP via Remote Desktop.

   The user account under which configuration is carried out is automatically enabled for Remote Desktop.

**Setting up the Remote Desktop connection on the remote computer**

1. On the remote computer, select "Start > Windows Accessories > Remote Desktop Connection".
2. Enter the instrument's name or IP address in the dialog box.
   See also Chapter 23.2.2, "Assigning the IP address", on page 1138.
3. Enter the user ID and password for the instrument. See Chapter 23.1, "Operating system", on page 1132 for details.
4. Click "Connect".

   When the connection has been set up, the instrument's screen appears on the remote computer.

For detailed information about Remote Desktop and the connection, refer to the Windows Help.

**Helpful settings for Remote Desktop**

The following settings for the Remote Desktop connection can make working on the remote PC more convenient.

1. When setting up the connection to the instrument, you can configure the connection settings in the "Remote Desktop Connection" dialog box.
   Click the "Show Options".

   The dialog box is expanded to display the configuration data.
2. Customize the settings:
   a) On the "Experience" tab, select the appropriate connection to optimize the connection speed.

b) On the "Local Resources" tab:
   - To use printers connected to the remote PC while accessing them from the instrument, activate "Printers" under "Local devices and resources".
   - If you want access drives of the remote PC from the instrument, e.g. to store settings or to copy files from the PC to the instrument:
     – Select "More" under "Local devices and resources".
     – Select the "Drives" that are needed.
     Windows maps the selected drives of the remote PC to corresponding network drives. When a connection is established, a warning on the PC indicates that the drives are enabled for access from the instrument.

c) On the "Display" tab:
   - Use the slider to set the size of the R&S RTP window on the remote PC desktop.
   - Activate "Display the connection bar when in full screen mode".
     A bar with the network address of the instrument is displayed on the screen, which you can use to reduce, minimize or close the window.

d) On the "General" tab, you can save the connection settings for later use: click "Save As".

**Terminating Remote Desktop Control**

A Remote Desktop connection can be terminated either on the R&S RTP or on the remote PC. The connection can be established again any time as long as remote control is enabled on the instrument. Consider the notice "Risk of unauthorized access" on page 1147.

► To terminate the connection on the remote PC, close the "Remote Desktop" window, or select "Start > Disconnect".

## 23.5 Remote control

Remote control automates the operation of the instrument using SCPI commands, scripts and programs.

For general information on remote control of Rohde & Schwarz products via SCPI, refer to www.rohde-schwarz.com/rc-via-scpi.

The following sections describe the instrument-specific basics of remote control. SCPI commands are listed described in chapter "Remote Control Commands".

- Remote control interfaces and protocols ..... 1150
- Remote settings ..... 1151
- Starting and stopping remote control ..... 1153
- Remote control - status reporting system ..... 1154

### 23.5.1 Remote control interfaces and protocols

The instrument supports different interfaces for remote control. The following table gives an overview.

***Table 23-1: Remote control interfaces and protocols***

| Interface | Protocols, VISA address string | Remarks |
|---|---|---|
| Local Area Network (LAN) | **Protocol HiSLIP**<br>VISA address string:<br>`TCPIP::<host address>::hislip0[,<port>][::INSTR]`<br>**Protocol VXI-11**<br>VISA address string:<br>`TCPIP::<host address>[::inst0]::[INSTR]` | The LAN connector is located on rear panel of the instrument.<br>The interface is based on TCP/IP and supports various protocols. |
| GPIB (IEC/IEEE Bus Interface) | VISA address string:<br>GPIB::primary address[::INSTR]<br>(no secondary address) | The GPIB bus interface according to standard IEC 625.1/IEEE 488.1 is located on the rear panel of the instrument. |

Within this interface description, the term GPIB is used as a synonym for the IEC/IEEE bus interface.

#### 23.5.1.1 LAN interface

To be integrated in a LAN, the instrument is equipped with a LAN interface, consisting of a connector, a network interface card and protocols. The interface details are given in the specifications.

For remote control via a network, the PC and the instrument must be connected via the LAN interface to a common network with TCP/IP network protocol. They are connected using a commercial RJ45 cable (shielded or unshielded twisted pair category 5). The TCP/IP network protocol and the associated network services are preconfigured on the instrument. Software for instrument control and the VISA program library must be installed on the controller.

**IP address**

Only the IP address or a valid DNS host name is required to set up the connection. The host address is part of the "VISA resource string" used by programs to identify and control the instrument. The VISA resource string has the form:

`TCPIP::<host address>::hislip0[,<port>][::INSTR]` for HiSLIP protocol

`TCPIP::<host address>[::inst0][::INSTR]` for VXI-11 protocol

Where:

- `host address` identifies the instrument in the network, usually the IP address. If the LAN is supported by a DNS server, the host name can be used instead of the IP address. The DNS server (Domain Name System server) translates the host name to the IP address.
- `hislip0` indicates the HiSLIP protocol

- `inst0` is the default LAN device name. VISA supports several devices running on the instrument. On R&S RTP, only one device is configured, so the LAN device name can be omitted.
- `INSTR` specifies a VISA resource of the type INSTR. By default, the VISA resource name control is set to the INSTR class.

**Example: HiSLIP**

IP address is *192.1.2.3*: the valid resource string is: `TCPIP::192.1.2.3::hislip0`

Instrument name is *RSRT1*: the valid resource string is: `TCPIP::RSRT1::hislip0`.

DNS host name is *RTP-123456*: the valid resource string is: `TCPIP::RTP-123456::hislip0`.

**Example: VXI-11**

IP address is *192.1.2.3*: the valid resource string is: `TCPIP::192.1.2.3`

Instrument name is *RSRT1*: the valid resource string is: `TCPIP::RSRT1`.

DNS host name is *RTP-123456*: the valid resource string is: `TCPIP::RTP-123456`.

See also:

- Find IP address: "Menu" > "Settings" > "System" tab, see "IP Address" on page 72
- Chapter 23.2.2, "Assigning the IP address", on page 1138

#### 23.5.1.2 GPIB interface (IEC/IEEE bus interface)

To control the instrument remotely via GPIB, the instrument and the controlle must be connected by a GPIB bus cable. The total cable length is restricted to a maximum of 15 m; the cable length between two instruments should not exceed 2m.

**GPIB instrument address**

The GPIB address is set on the instrument, see Chapter 23.5.2.1, "GPIB settings", on page 1151.

### 23.5.2 Remote settings

The settings on this tab are required for remote control of the instrument via a connected computer.

#### 23.5.2.1 GPIB settings

Access: "Menu" > "Settings" > "System" > "Remote"

Displays the GPIB settings of the instrument.

**Address**
Indicates the GPIB address of the instrument.

The remote control address is factory-set to 20, but it can be changed if it does not fit in the network environment. For remote control, addresses 0 through 30 are allowed. The GPIB address is maintained after a reset of the instrument settings. Be aware that changing the address has major effects on the communication to the remote computer.

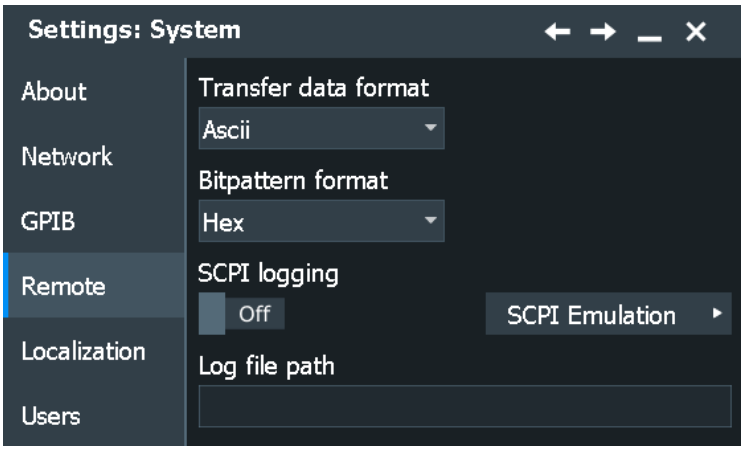
Remote command:
`GPIB:ADDRess` on page 1204

**Terminator**
Specifies the symbol that is used as a terminator in GPIB communication.

Remote command:
`GPIB:TERMinator` on page 1204

### 23.5.2.2 Remote settings

Access: "Menu" > "Settings" > "System" > "Remote"




The following settings are required for remote control of the instrument via a connected computer.

**Transfer data format**
Selects the data format that is used for transmission of waveform data from the instrument to the controlling computer.

Waveform data can be retrieved using the following commands:
- `CHANnel<m>[:WAVeform<n>]:DATA[:VALues]?`
- `CALCulate:MATH<m>:DATA[:VALues]?`
- `REFCurve<m>:DATA[:VALues]?`
- `DIGital<m>:DATA[:VALues]?`

The content of the data stream can be defined with "Menu" > "Save/Recall" > "Save > Waveforms > Interleaved X/Y" (or `EXPort:WAVeform:INCXvalues`).

| | |
|---|---|
| "Ascii" | Data values are returned in ASCII format as a list of comma separated values in floating point format. |
| "FLOAT" | Binary format. The data is stored as binary data (Definite Length Block Data according to IEEE 488.2). |

"INT8" Signed integer data with length 8 bit.

"INT16" Signed integer data with length 16 bit. The Byte order can be set using .
For details on the formats, refer to the description of the remote command.

Remote command:
`FORMat[:DATA]` on page 1201

**Byte order**
Sets the endianess for INT16 data:
- LSB first: little endian, least significant byte first
- MSB first: big endian, most significant byte first

Remote command:
`FORMat:BORDer` on page 1203

**Bit pattern format**
Sets the format for all bit pattern queries.

Remote command:
`FORMat:BPATtern` on page 1203

**SCPI logging**
If enabled, all received remote commands are written into a text file. Enter the path and filename with extension (log, txt, or csv) in "Log file path".

## 23.5.3 Starting and stopping remote control

### 23.5.3.1 Starting a remote control session

When you switch on the instrument, it is always in manual operation state ("local" state). It can be operated via the front panel, the touch screen and external keyboard and/or mouse.

► To start remote control:
- Send a command from the controller.
- VXI-11 protocol (LAN or USB interface): Use `&GTR` interface message.

While remote control is active, the instrument settings are optimized for maximum measurement speed; the display is switched off. Operation via the front panel is disabled.

On the touch screen, two buttons appear in the upper left corner: "Local" and "View".

#### 23.5.3.2 Using the display during remote control

You can observe the screen while a remote control script is executed. This is helpful for program test purposes but tends to slow down the measurement. Therefore it is recommended that you switch off the display in real measurement applications where a tested program script is to be executed repeatedly.

► To switch on the display, do one of the following:

- Tap the "View" button in the upper left corner of the touch screen.
- Use the `SYSTem:DISPlay:UPDate ON` command.

► To switch off the display, do one of the following:

- Tap the "View" button again.
- Use the `SYSTem:DISPlay:UPDate OFF` command.

#### 23.5.3.3 Returning to manual operation

The instrument switches back to manual operation when the remote connection is closed. Besides, you can return to manual operation manually or via remote control.

► To return to manual operation:

- Tap the "Local" button in the upper left corner of the touch screen.
- VXI-11 protocol: Use `&GTL` interface message.

### 23.5.4 Remote control - status reporting system

The status reporting system stores all information on the current operating state of the instrument, and on errors which have occurred. This information is stored in the status registers and in the error queue. Both can be queried via `STATus...` commands.

#### 23.5.4.1 Hierarchy of status registers

As shown in the following figure, the status information is of hierarchical structure.

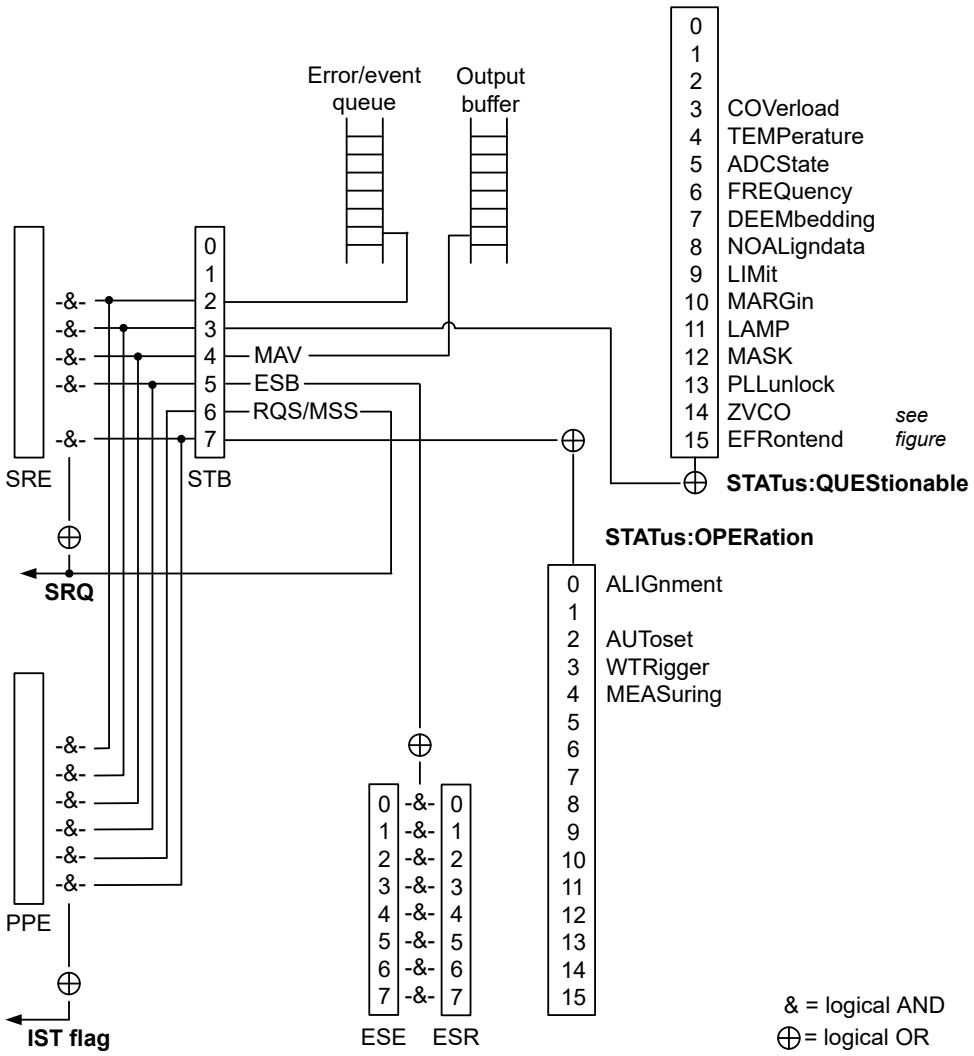



*Figure 23-1: Overview of the status registers hierarchy*

All status registers have the same internal structure.

### 23.5.4.2 Contents of the status registers

#### STATus:OPERation register

In the `CONDition` part, this register contains information on which actions the instrument is executing. In the `EVENt` part, it contains information on which actions the instrument has executed since the last reading.

It can be read using the commands `STATus:OPERation:CONDition?` or `STATus:OPERation[:EVENt]?`, see also Chapter 24.27.1, "STATus:OPERation register", on page 2463.

*Table 23-2: Bits in the STATus:OPERation register*

| Bit No. | Meaning |
|---|---|
| 0 | `ALIGnment`<br>This bit is set as long as the instrument is performing a self-alignment. |
| 1 | Not used |
| 2 | `AUToset`<br>This bit is set while the instrument is performing an auto setup. |
| 3 | `WTRIgger`<br>The wait for trigger status bit indicates that the instrument is ready to trigger, and the pre-trigger time is expired. The bit is set if the instrument did not trigger for more than 10 ms.<br>The bit is only valid if the trigger event is initiated by the user, for example, using an external generator. If the instrument triggers on signals or if it is in auto trigger mode, the bit status is undefined. |
| 4 | `MEASuring`<br>The bit is set as long as an acquisition - sampling and postprocessing - is running. In run continuous mode, the bit is always set. |
| 5 - 15 | Not used |

### STATus:QUEStionable register

This register contains information about indefinite states which can occur if the unit is operated without meeting the specifications. It can be read using the commands `STATus:QUEStionable:CONDition?` and `STATus:QUEStionable[:EVENt]?`.

The remote commands for the STATus:QUEStionable register are described in Chapter 24.27.2, "STATus:QUEStionable registers", on page 2464.

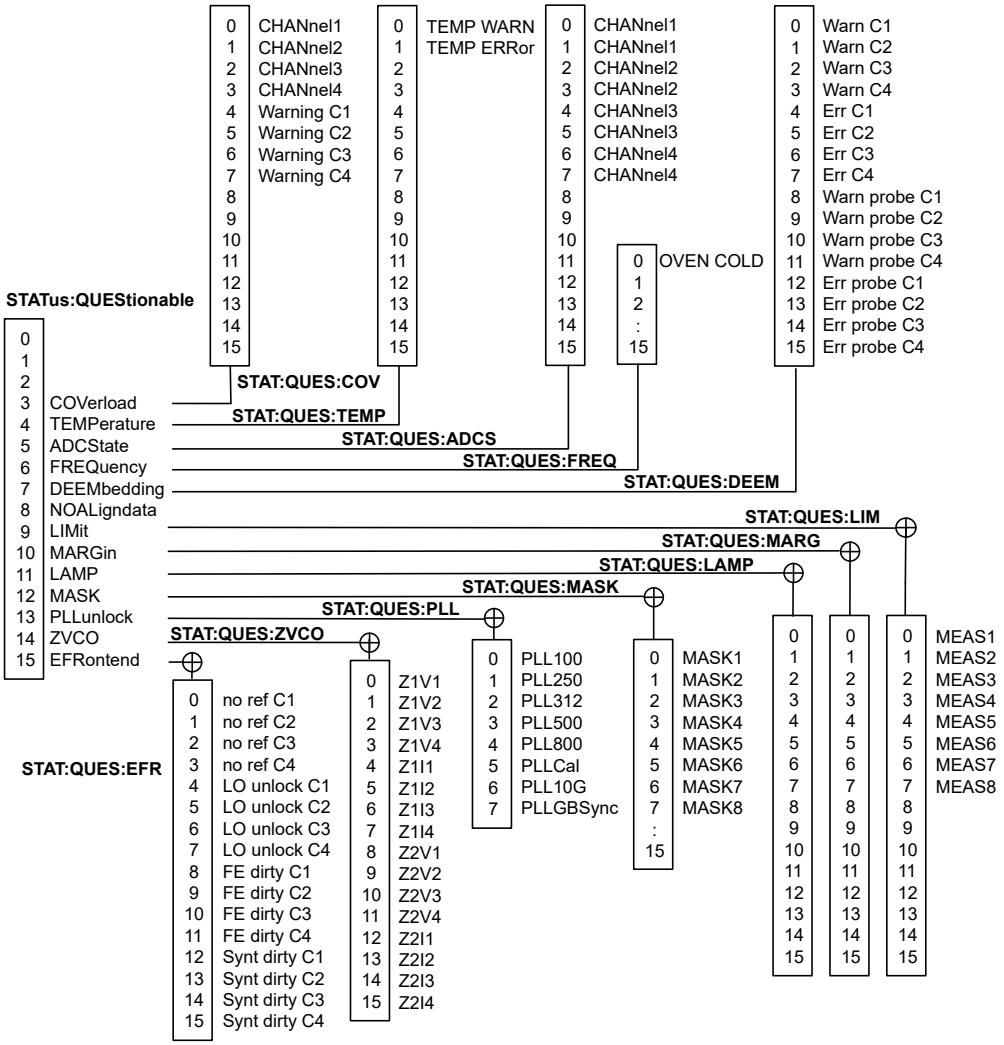



*Figure 23-2: Overview of the STATus:QUEStionable register*

*Table 23-3: Bits in the STATus:QUEStionable register*

| Bit No. | Meaning |
|---|---|
| 0 to 2 | not used |
| 3 | COVerload<br>This bit is set if a questionable channel overload occurs (see "STATus:QUEStionable:COVerload register" on page 1158). |
| 4 | TEMPerature<br>This bit is set if a questionable temperature occurs (see "STATus:QUEStionable:TEMPerature register" on page 1159). |
| 5 | ADCState<br>The bit is set if the signal is clipped on the upper or lower edge of the screen, if an overflow of the ADC occurs (see "STATus:QUEStionable:ADCState register" on page 1159). |

| Bit No. | Meaning |
|---|---|
| 6 | FREQuency<br>This bit is set if there is anything wrong with the frequency of the local oscillator or the reference frequency (see "STATus:QUEStionable:FREQuency register" on page 1159). |
| 7 | DEEMbedding<br>The bit is set if a warning or an error occurs during deembedding (see "STATus:QUEStionable:DEEMbedding register" on page 1160). |
| 8 | NOALigndata<br>This bit is set if no alignment data is available - the instrument is uncalibrated. |
| 9 | LIMit<br>This bit is set if a limit value is violated (see "STATus:QUEStionable:LIMit, STATus:QUEStionable:MARGin registers" on page 1160). |
| 10 | MARGin<br>This bit is set if a margin value is violated, and the limit value is not violated (see "STATus:QUEStionable:LIMit, STATus:QUEStionable:MARGin registers" on page 1160). |
| 11 | LAMP (Low AMPlitude)<br>This bit is set if the magnitude of the signal is too low to get reliable measurement results. See "STATus:QUEStionable:LAMP register" on page 1161. |
| 12 | MASK<br>This bit is set if a mask value is violated (see "STATus:QUEStionable:MASK register" on page 1161 |
| 13 | PLLunlock<br>See "STATus:QUEStionable:PLLunlock register" on page 1161. |
| 14 | ZVCO: This bit is set if a channel overload occurs at one or more input channels of the R&S RT-ZVC multi-channel power probe. See "STATus:QUEStionable:ZVCO register" on page 1162. |
| 15 | EFRontend: this bit is set if problems occur with the connection of external frontends (requires option R&S RTP-K553). See "STATus:QUEStionable:EFRontend register" on page 1163. |

### STATus:QUEStionable:COVerload register

This register contains all information about overload of the channels. The bit is set if the assigned channel is overloaded, or if an overload risk occurred (overload warning).

***Table 23-4: Bits in the STATus:QUEStionable:COVerload register***

| Bit No. | Meaning |
|---|---|
| 0 | Overload on CHANnel1 |
| 1 | Overload on CHANnel2 |
| 2 | Overload on CHANnel3 |
| 3 | Overload on CHANnel4 |
| 4 | Overload warning for CHANnel1 |
| 5 | Overload warning for CHANnel2 |
| 6 | Overload warning for CHANnel3 |

| Bit No. | Meaning |
|---|---|
| 7 | Overload warning for CHANnel4 |
| 8 | Overload on external trigger input |
| 9 | Overload warning for external trigger input |

### STATus:QUEStionable:TEMPerature register

This register contains information about the instrument's temperature.

*Table 23-5: Bits in the STATus:QUEStionable:TEMPerature register*

| Bit No. | Meaning |
|---|---|
| 0 | TEMP WARN<br>This bit is set if a temperature warning on channel 1, 2, 3 or 4 occurred. |
| 1 | TEMP ERRor<br>This bit is set if a temperature error on channel 1, 2, 3 or 4 occurred. |

### STATus:QUEStionable:ADCState register

This register contains all information about overflow of the ADC.

The bit is set if the assigned channel signal is clipped on the upper or lower edge of the screen. In this case, the signal does not fit in the range of the ADC and overflow occurs.

*Table 23-6: Bits in the STATus:QUEStionable:ADCState register*

| Bit No. | Meaning |
|---|---|
| 0 | CHANnel1, clipping on the upper limit |
| 1 | CHANnel1, clipping on the lower limit |
| 2 | CHANnel2, clipping on the upper limit |
| 3 | CHANnel2, clipping on the lower limit |
| 4 | CHANnel3, clipping on the upper limit |
| 5 | CHANnel3, clipping on the lower limit |
| 6 | CHANnel4, clipping on the upper limit |
| 7 | CHANnel4, clipping on the lower limit |

### STATus:QUEStionable:FREQuency register

*Table 23-7: Bits in the STATus:QUEStionable:FREQuency register*

| Bit No. | Meaning |
|---|---|
| 0 | Oven cold. This bit is set if the reference oscillator has not yet attained its operating temperature. |

### STATus:QUEStionable:DEEMbedding register

*Table 23-8: Bits in the STATus:QUEStionable:DEEMbedding register*

| Bit No. | Meaning |
|---|---|
| 0<br>1<br>2<br>3 | Warning for channel 1, 2, 3, 4 accordingly. The bits are set if there are problems in the deembedding configuration of the assigned channel. |
| 4<br>5<br>6<br>7 | Error for channel 1, 2, 3, 4 accordingly. The bits are set if the deembedding configuration of the assigned channel causes an error. |
| 8<br>9<br>10<br>11 | Warning for channel 1, 2, 3, 4 accordingly. The bits are set if there are problems in the probe deembedding of the assigned channel. |
| 12<br>13<br>14<br>15 | Error for channel 1, 2, 3, 4 accordingly. The bits are set if the probe deembedding of the assigned channel causes an error. |

### STATus:QUEStionable:LIMit, STATus:QUEStionable:MARGin registers

These registers contain information about the observance of the limits or margins of measurements.

The LIMit bit is set if the limit of the assigned measurement is violated. The MARGin bit is set if the margin of the assigned measurement is violated but not the limit (the limit bit is not set).

The status bits are set if the limits or margins of one or more assigned measurements are violated.

If multiple measurements are active, all measurement results affect the status bits. Statistical results do not change the status bits.

*Table 23-9: Bits in the STATus:QUEStionable:LIMit and STATus:QUEStionable.MARGin registers*

| Bit No. | Meaning |
|---|---|
| 0 | MEAS1 |
| 1 | MEAS2 |
| 2 | MEAS3 |
| 3 | MEAS4 |
| 4 | MEAS5 |
| 5 | MEAS6 |

| Bit No. | Meaning |
|---|---|
| 6 | MEAS7 |
| 7 | MEAS8 |

### STATus:QUEStionable:LAMP register

The LAMP (Low AMPlitude) bit is set if the magnitude of the signal is too low to get reliable measurement results.

*Table 23-10: Bits in the STATus:QUEStionable:LAMP register*

| Bit No. | Meaning |
|---|---|
| 0 | MEAS1 |
| 1 | MEAS2 |
| 2 | MEAS3 |
| 3 | MEAS4 |
| 4 | MEAS5 |
| 5 | MEAS6 |
| 6 | MEAS7 |
| 7 | MEAS8 |

### STATus:QUEStionable:MASK register

This register contains information about the violation of masks. This bit is set if the assigned mask is violated.

*Table 23-11: Bits in the STATus:QUEStionable:MASK register*

| Bit No. | Meaning |
|---|---|
| 0 | MASK1 |
| 1 | MASK2 |
| 2 | MASK3 |
| 3 | MASK4 |
| 4 | MASK5 |
| 5 | MASK6 |
| 6 | MASK7 |
| 7 | MASK8 |

### STATus:QUEStionable:PLLunlock register

This register contains information on the phase-locked loop process status. An unlocked status sets the bit true.

*Table 23-12: Bits in the STATus:QUEStionable:PLLunlock register*

| Bit No. | Meaning |
|---|---|
| 0 | PLL100: indicates if the frequency reference is locked. The frequency reference is usually the OCXO. It can also be an external reference if it is connected and activated. |
| 1 | not used |
| 2 | not used |
| 3 | PLL500: internal use |
| 4 | PLL800: internal use |
| 5 | PLLCal: internal use |
| 6 | not used |
| 7 | PLLGBSync: internal use |

### STATus:QUEStionable:ZVCO register

This register contains all information about overload of the R&S RT-ZVC input channels. The bit is set if the assigned channel is overloaded.

*Table 23-13: Bits in the STATus:QUEStionable:ZVCO register*

| Bit No. | Meaning |
|---|---|
| 0 | Overload on Z1V1 |
| 1 | Overload on Z1V2 |
| 2 | Overload on Z1V3 |
| 3 | Overload on Z1V4 |
| 4 | Overload on Z1I1 |
| 5 | Overload on Z1I2 |
| 6 | Overload on Z1I3 |
| 7 | Overload on Z1I4 |
| 8 | Overload on Z2V1 |
| 9 | Overload on Z2V2 |
| 10 | Overload on Z2V3 |
| 11 | Overload on Z2V4 |
| 12 | Overload on Z2I1 |
| 13 | Overload on Z2I2 |
| 14 | Overload on Z2I3 |
| 15 | Overload on Z2I4 |

### STATus:QUEStionable:EFRontend register

This register contains information on connected frontends (requires option R&S RTP-K553). The bit is set if connection errors occur.

*Table 23-14: Bits in the STATus:QUEStionable:EFR register*

| Bit No. | Meaning |
|---|---|
| 0 | No reference signal on C1 |
| 1 | No reference signal on C2 |
| 2 | No reference signal on C3 |
| 3 | No reference signal on C4 |
| 4 | LO unlocked on C1 |
| 5 | LO unlocked on C2 |
| 6 | LO unlocked on C3 |
| 7 | LO unlocked on C4 |
| 8 | Frontend dirty on C1 |
| 9 | Frontend dirty on C2 |
| 10 | Frontend dirty on C3 |
| 11 | Frontend dirty on C4 |
| 12 | Synthesizer dirty on C1 |
| 13 | Synthesizer dirty on C2 |
| 14 | Synthesizer dirty on C3 |
| 15 | Synthesizer dirty on C4 |

## 23.5.4.3 Reset values of the status reporting system

The following table contains the different commands and events causing the status reporting system to be reset. None of the commands, except *RST and SYSTem:PRESet, influence the functional instrument settings. In particular, DCL does not change the instrument settings.

*Table 23-15: Resets of the status reporting system*

| Event | Switching on supply voltage<br>Power-On-Status-Clear | | DCL, SDC<br>(Device Clear, Selected Device Clear) | *RST or SYSTem:PRESet | STATus:PRESet | *CLS |
|---|---|---|---|---|---|---|
| Effect | 0 | 1 | | | | |
| Clear STB, ESR | - | yes | - | - | - | yes |
| Clear SRE, ESE | - | yes | - | - | - | - |
| Clear PPE | - | yes | - | - | - | - |

| Event | Switching on supply voltage<br>Power-On-Status-Clear | | DCL, SDC<br>(Device Clear, Selected Device Clear) | *RST or SYS-Tem:PRE Set | STA-Tus:PRE-Set | *CLS |
|---|---|---|---|---|---|---|
| **Effect** | **0** | **1** | | | | |
| Clear EVENt parts of the registers | - | yes | - | - | - | yes |
| Clear ENABle parts of all OPERation and QUEStionable registers;<br>Fill ENABle parts of all other registers with "1". | - | yes | - | - | yes | - |
| Fill PTRansition parts with "1";<br>Clear NTRansition parts | - | yes | - | - | yes | - |
| Clear error queue | yes | yes | - | - | - | yes |
| Clear output buffer | yes | yes | yes | 1) | 1) | 1) |
| Clear command processing and input buffer | yes | yes | yes | - | - | - |
| 1) The first command in a command line that immediately follows a <PROGRAM MESSAGE TERMINATOR> clears the output buffer. | | | | | | |

# 24 Remote control commands

This chapter describes all remote commands available for R&S RTP and provides examples and information how to use the commands.

For further information on remote control, see Chapter 23.5, "Remote control", on page 1149.

**24.1 Conventions used in remote command description....1171**
**24.2 Finding the appropriate command....1172**
**24.3 Programming examples....1172**
24.3.1 Display....1173
24.3.2 Automatic measurements....1175
24.3.3 Mask testing....1178
24.3.4 Search....1179
24.3.5 Data management....1179
24.3.6 Deembedding....1186
24.3.7 Protocol analysis....1187
**24.4 Frequently used parameters and suffixes....1188**
24.4.1 Waveform suffix....1188
24.4.2 Waveform parameter....1193
24.4.3 Slope parameter....1194
24.4.4 Polarity parameter....1194
24.4.5 Event parameter....1195
24.4.6 Bit pattern parameter....1195
**24.5 Common commands....1196**
**24.6 General remote settings....1201**
**24.7 Instrument setup....1205**
24.7.1 System....1205
24.7.2 Appearance....1207
24.7.3 Display....1229
24.7.4 User-defined preset....1231
24.7.5 Save / recall....1232
24.7.6 Maintenance....1232
**24.8 Acquisition and setup....1234**
24.8.1 Starting and stopping acquisition....1235

24.8.2 Time base....1236
24.8.3 Acquisition....1242
24.8.4 Fast segmentation....1247
24.8.5 Vertical....1248
24.8.6 Waveform data....1252
24.8.7 Probes....1254
24.8.8 R&S RT-ZVC probe....1280
24.8.9 Differential signals....1294
24.8.10 Digital filter....1298
24.8.11 Skew....1299
24.8.12 AUX OUT....1301
24.8.13 High definition mode....1302
24.8.14 Reference clock....1303
**24.9 Trigger....1304**
24.9.1 Basic trigger settings....1304
24.9.2 Edge trigger....1308
24.9.3 Glitch trigger....1308
24.9.4 Width trigger....1309
24.9.5 Runt trigger....1311
24.9.6 Window trigger....1313
24.9.7 Timeout trigger....1316
24.9.8 Interval trigger....1317
24.9.9 Slew rate trigger....1319
24.9.10 Setup & Hold....1321
24.9.11 State trigger....1324
24.9.12 Pattern trigger....1324
24.9.13 High-speed serial pattern trigger (option R&S RTP-K140/K141)....1327
24.9.14 CDR trigger....1348
24.9.15 TV/Video trigger....1351
24.9.16 Line trigger....1356
24.9.17 Trigger qualification....1357
24.9.18 Holdoff....1360
24.9.19 Conditioning....1364

24.9.20 Trigger sequence ... 1367
24.9.21 Trigger control ... 1370
24.9.22 Actions on trigger ... 1372
24.9.23 Zone trigger (option R&S RTP-K19) ... 1374
24.9.24 External trigger input ... 1375
24.9.25 Acquisition info ... 1378
**24.10 Waveform analysis ... 1378**
24.10.1 Zoom ... 1378
24.10.2 Reference waveforms ... 1388
24.10.3 Mathematics ... 1400
24.10.4 History ... 1405
24.10.5 XY-Diagram ... 1410
**24.11 Cursor measurements ... 1413**
24.11.1 Cursor setup ... 1413
24.11.2 Cursor results ... 1423
24.11.3 Peak search using cursors ... 1424
**24.12 Automatic measurements ... 1426**
24.12.1 General settings ... 1427
24.12.2 Results ... 1432
24.12.3 Amplitude/time measurements ... 1436
24.12.4 Arithmetic for amplitude/time and spectrum measurements ... 1447
24.12.5 Eye diagram measurements ... 1456
24.12.6 Spectrum ... 1466
24.12.7 Optical measurements ... 1473
24.12.8 Histograms ... 1476
24.12.9 Display ... 1487
24.12.10 Statistics and long-term measurements ... 1488
24.12.11 Protocol ... 1495
24.12.12 Track ... 1502
24.12.13 Gating ... 1503
24.12.14 Limit check and event actions ... 1506
24.12.15 Reference levels ... 1509
**24.13 Spectrum analysis ... 1524**

24.13.1 Basic FFT....1524
24.13.2 Waveform data....1538
24.13.3 Spectrum analysis (option R&S RTP-K37)....1539
**24.14 Mask testing....1542**
24.14.1 Mask test definition....1542
24.14.2 Mask definition: user mask....1546
24.14.3 Mask definition: waveform mask....1553
24.14.4 Event actions....1555
24.14.5 Results....1558
24.14.6 Eye mask testing....1560
**24.15 Search....1565**
24.15.1 General search settings....1566
24.15.2 Basic trigger search conditions....1568
24.15.3 Edge search conditions....1570
24.15.4 Glitch search conditions....1570
24.15.5 Interval search conditions....1572
24.15.6 Runt search conditions....1573
24.15.7 Slew rate search conditions....1576
24.15.8 Timeout search conditions....1578
24.15.9 Width search conditions....1579
24.15.10 Window search conditions....1581
24.15.11 Data2clock search conditions....1584
24.15.12 Pattern search conditions....1586
24.15.13 State search conditions....1591
24.15.14 Search on spectrum....1594
24.15.15 Search gate settings....1595
24.15.16 Search results....1598
**24.16 Data management....1604**
24.16.1 Instrument settings....1604
24.16.2 Savesets and one file....1612
24.16.3 Waveform data transmission....1613
24.16.4 Waveform data export to file....1613
24.16.5 Waveform histogram export to file....1622

24.16.6 Eye....1623
24.16.7 Long term measurement results and measurement histogram export to file....1625
24.16.8 Screenshots....1627
24.16.9 Reports....1631
24.16.10 Autonaming....1633
24.16.11 CSV export....1634
24.16.12 External application....1635
**24.17 Protocols....1636**
24.17.1 Configuration settings for all serial protocols....1636
24.17.2 Trigger Settings for all serial protocols....1641
24.17.3 I²C (option R&S RTP-K1)....1643
24.17.4 SPI (option R&S RTP-K1)....1677
24.17.5 UART/RS-232/RS-422/RS-485 (option R&S RTP-K2)....1697
24.17.6 CAN (option R&S RTP-K3/K9)....1708
24.17.7 LIN (option R&S RTP-K3)....1752
24.17.8 MIL-1553 (option R&S RTP-K6)....1783
24.17.9 ARINC 429 (option R&S RTP-K7)....1810
24.17.10 Ethernet 10BASE-T and 100BASE-TX (option R&S RTP-K8)....1827
24.17.11 Ethernet 100BASE-T1 (option R&S RTP-57)....1852
24.17.12 Ethernet 1000BASE-T1 (option R&S RTP-K58)....1883
24.17.13 RFFE (option R&S RTP-K40)....1916
24.17.14 D-PHY (option R&S RTP-K42)....1949
24.17.15 M-PHY (option R&S RTP-K44)....1982
24.17.16 Custom: manchester / NRZ (option R&S RTP-K50)....2043
24.17.17 8B/10B (option R&S RTP-K52)....2088
24.17.18 MDIO (option R&S RTP-K55)....2104
24.17.19 USB (option R&S RTP-K60)....2120
24.17.20 USB 3.1 (option R&S RTP-K61/K62)....2162
24.17.21 USBPD (option R&S RTP-K63)....2200
24.17.22 Space wire (option R&S RTP-K65)....2225
24.17.23 PCIe (option R&S RTPK72/K73)....2244
24.17.24 DDR (option R&S RTP-K91/K93)....2293
**24.18 Mixed signal option (MSO, R&S RTP-B1)....2309**

24.18.1 Digital channels....2310
24.18.2 Parallel bus configuration....2313
24.18.3 Digital resolution....2320
24.18.4 Trigger settings for digital signals and parallel buses....2321
24.18.5 MSO data....2332
**24.19 Waveform generator (option R&S RTP-B6)....2334**
24.19.1 Waveform generator setup....2334
24.19.2 Pattern generator setup....2348
24.19.3 Output setup....2350
24.19.4 Noise setup....2352
24.19.5 DC offset alignment....2354
24.19.6 Coupling & sync....2355
**24.20 Pulse source and TDR/TDT analysis (options R&S RTP-B7/K130)....2357**
24.20.1 Pulse source (option R&S RTP-B7)....2357
24.20.2 TDR/TDT analysis....2359
**24.21 I/Q software interface (option R&S RTP-K11)....2366**
24.21.1 I/Q sampling settings....2367
24.21.2 I/Q data output....2372
**24.22 Jitter analysis and clock data recovery....2373**
24.22.1 Jitter measurements (option R&S RTP-K12)....2374
24.22.2 Advanced jitter and noise (option R&S RTP-K133/K134)....2379
24.22.3 Clock data recovery (software-based, option R&S RTP-K12)....2406
24.22.4 Eye mask testing....2412
**24.23 Eye diagram analysis....2412**
24.23.1 Basic setup....2413
24.23.2 Display....2419
24.23.3 Qualify: gate settings....2421
24.23.4 Qualify: signal settings....2424
24.23.5 Filter....2425
24.23.6 Results of eye....2428
24.23.7 Results of mask tests on eye diagrams....2429
**24.24 Deembedding (option R&S RTP-K121/-K122)....2432**
24.24.1 Measurement setup....2432

24.24.2 Save and load the deembedding setup....2435
24.24.3 Components....2436
24.24.4 Responses....2443
**24.25 External frontend control (option R&S RTP-K553)....2444**
24.25.1 Setup and connection....2444
24.25.2 Frontend settings and parameters....2447
24.25.3 Accessories....2460
**24.26 Maintenance....2460**
**24.27 Status reporting....2463**
24.27.1 STATus:OPERation register....2463
24.27.2 STATus:QUEStionable registers....2464
**24.28 Remote trace....2469**
24.28.1 Standard commands....2470
24.28.2 Diagnostic remote trace commands....2472
**24.29 Deprecated commands....2473**
24.29.1 Base instrument....2473
24.29.2 Options....2474

## 24.1 Conventions used in remote command description

The following conventions are used in the remote command descriptions:

- **Command usage**
  If not specified otherwise, commands can be used both for setting and for querying parameters.
  If a command can be used for setting or querying only, or if it initiates an event, the usage is stated explicitly.
- **Parameter usage**
  If not specified otherwise, a parameter can be used to set a value and it is the result of a query.
  Parameters required only for setting are indicated as **Setting parameters**.
  Parameters required only to refine a query are indicated as **Query parameters**.
  Parameters that are only returned as the result of a query are indicated as **Return values**.
- **Conformity**
  Commands that are taken from the SCPI standard are indicated as **SCPI confirmed**. All commands used by the R&S RTP follow the SCPI syntax rules.
- **Asynchronous commands**

A command which does not automatically finish executing before the next command starts executing (overlapping command) is indicated as an **Asynchronous command**.

- **Reset values (*RST)**
  Default parameter values that are used directly after resetting the instrument (`*RST` command) are indicated as ***RST** values, if available.
- **Default unit**
  The default unit is used for numeric values if no other unit is provided with the parameter.

## 24.2 Finding the appropriate command

In the following chapters, the commands are sorted according to the menu and dialog structure of the instrument.

A list of all commands in alphabetical order ist given in the "List of Commands" at the end of this documentation.

To find the appropriate command for a setting easily, you can use the context help:

1. Enable the "Help" icon on the toolbar.

2. Tap the parameter for which you need information.

   The tooltip opens.

3. Tap the "Show Help" button in the lower right corner of the tooltip.

   The "Help" window opens and displays the comprehensive description and the corresponding remote command.

4. Tap the remote command link to open the command description.

## 24.3 Programming examples

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"

- Chapter 24.3, "Programming examples", on page 1172

24.3.1 Display....1173
24.3.1.1 Creating diagrams (SmartGrid)....1173
24.3.1.2 Creating zoom diagrams....1174
24.3.2 Automatic measurements....1175
24.3.2.1 Performing amplitude/time measurements....1175
24.3.2.2 Setting reference levels....1175
24.3.2.3 Waveform histograms....1176
24.3.2.4 Long-term measurements....1178
24.3.3 Mask testing....1178
24.3.3.1 Creating a user mask....1178
24.3.4 Search....1179
24.3.4.1 Searching for a pulse of specified width....1179
24.3.5 Data management....1179
24.3.5.1 Saving a screenshot to file....1180
24.3.5.2 Exporting waveform data to file....1180
24.3.5.3 Exporting measurement results to file....1186
24.3.6 Deembedding....1186
24.3.6.1 Adding cables to the measurement setup....1186
24.3.7 Protocol analysis....1187
24.3.7.1 RFFE (option R&S RTP-K40)....1187

## 24.3.1 Display

- Creating diagrams (SmartGrid)....1173
- Creating zoom diagrams....1174

### 24.3.1.1 Creating diagrams (SmartGrid)

The example creates 3 diagrams at different positions on the screen, and a zoom diagram. It also minimizes a wavform and adds it to another diagram

Command description in: Chapter 24.7.2.4, "Smartgrid", on page 1211

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

```
// Turn on display update
:SYSTem:DISPlay:UPDate ON
// Turn on Channel 2
```

```
:CHANnel2:STATe 1
// Create new diagram 'MyDiag2' below existing 'Diagram1' with waveform C2W1 in it
:LAYout:ADD 'Diagram1',VERTical,OFF,C2W1,'MyDiag2'
// Turn on Channel 3
:CHANnel3:STATe 1
// Create new diagram 'MyDiag3' right of existing 'MyDiag2'
:LAYout:ADD 'MyDiag2',HORizontal,OFF,C3W1,'MyDiag3'
// Turn on Channel 4
:CHANnel4:STATe 1
// Create new diagram 'MyDiag4' tabbed to existing 'Diagram1'
:LAYout:ADD 'Diagram1',TAB,OFF,C4W1,'MyDiag4'
// Select 'Diagram1'
:LAYout:SHOW 'Diagram1'
// Create zoom window of 'Diagram1'
:LAYout:ZOOM:ADD 'Diagram1',VERT,OFF,-10e-9,10e-9,-0.05,0.04,'MyZoom1'
// Minimize Channel 2 (Wfm1)
:LAYout:SIGNal:UNAssign C2W1
// Add Channel 2 (Wfm1) to Diagram1
:LAYout:SIGNal:ASSign 'Diagram1', C2W1

*OPC?
```

#### 24.3.1.2 Creating zoom diagrams

The example creates a zoom diagram, sets the relative size of the zoom area, and removes the zoom diagram.

Command description in: Chapter 24.10.1, "Zoom", on page 1378.

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

```
// Create an new zoom diagram for Diagram1
LAYout:ZOOM:ADD 'Diagram1', VERT, OFF, -10e-9, 20e-9, -0.1, 0.05, 'MyZoom1'
// Query the horizontal zoom mode, return value: ABS
LAYout:ZOOM:HORZ:MODE? 'Diagram1', 'MyZoom1'
<--ABS
// Set horizontal zoom mode to relative
LAYout:ZOOM:HORZ:MODE 'Diagram1', 'MyZoom1', REL
// Set horizontal zoom span in percent
LAYout:ZOOM:HORZ:REL:SPAN 'Diagram1', 'MyZoom1', 10
// Set horizontal zoom position in percent
LAYout:ZOOM:HORZ:REL:POS 'Diagram1', 'MyZoom1', 15
*OPC?
// Remove zoom diagram
LAYout:ZOOM:REM 'Diagram1', 'MyZoom1'
*OPC?
```

## 24.3.2 Automatic measurements


- Performing amplitude/time measurements....1175
- Setting reference levels....1175
- Waveform histograms....1176
- Long-term measurements....1178


### 24.3.2.1 Performing amplitude/time measurements

Command description is given in Chapter 24.12.1, "General settings", on page 1427 and Chapter 24.12.2, "Results", on page 1432

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

#### Simple frequency and amplitude measurement

```
SING;*OPC?
MEAS1:SOUR C1W1        // Configure frequency measurement
MEAS1:MAIN FREQ
MEAS1 ON
*OPC?
MEAS2:SOUR C1W1        // Configure amplitude measurement
MEAS2:MAIN AMPL
MEAS2 ON
*OPC?
MEAS1:RES:ACT?        // Get frequency result
MEAS2:RES:ACT?        // Get amplitude result
*OPC?
```

### 24.3.2.2 Setting reference levels

Command description in Chapter 24.12.15, "Reference levels", on page 1509

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

#### Manual reference level definition using relative values

Reference levels are set to 15%, 50%, and 85% of the high signal level for waveform Ch2Wfm1 (= suffix 5).

```
REFLevel5:LDETection MANual
REFLevel5:LMODe REL
REFLevel5:RELative:MODE USER
REFLevel5:RELative:LOWer 15
REFLevel5:RELative:MIDDle 50
REFLevel5:RELative:UPPer 85
*OPC?
```

#### Manual reference level definition using absolute values

Set reference levels manually for waveform C1W1 (= suffix 2), defining high and low signal levels and the distances between signal and reference levels.

```
REFLevel2:LDETection MANual
REFLevel2:LMODe ABS
REFLevel2:ABSolute:HIGH 0.12
REFLevel2:ABSolute:TDIStance 0.03
REFLevel2:ABSolute:Low -0.12
REFLevel2:ABSolute:BDIStance 0.04
*OPC?
REFLevel2:ABSolute:MLEVel?
<-- 0
```

Set reference levels manually for waveform C1W1 (= suffix 2) by defining upper and lower reference levels and the distances between signal and reference levels.

```
REFLevel2:LDETection MANual
REFLevel2:LMODe ABS
REFLevel2:USRLevel UREF
REFLevel2:ABSolute:LLEVel 0
REFLevel2:ABSolute:BDIStance 0.02
REFLevel2:ABSolute:ULEVel 0.2
REFLevel2:ABSolute:TDIStance 0.03
*OPC?
REFLevel2:ABSolute:MLEVel?
<-- 0.1
```

#### Automatic level detection, peak probability

Reference levels are set to the signal levels with the highest probability values for waveform C3W1 (= suffix 8).

```
REFLevel8:LDETection Auto
REFLevel8:AUTO:MODE PPRobability
*OPC?
```

### 24.3.2.3 Waveform histograms

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

- Creating and reading histograms ..... 1176
- Exporting histogram data to file ..... 1177
- Transferring histogram data ..... 1177

#### Creating and reading histograms

The example creates a histogram, activates two measurements (mean and standard deviation measurements of Histogram1), and queries the results of both measurements.

Command description in:

- Chapter 24.12.1, "General settings", on page 1427
- Chapter 24.12.8.2, "Histogram measurement", on page 1482
- Chapter 24.12.2, "Results", on page 1432

```
LAY:HIST:ADD 'Histogram1', C1W1, -2.5E-007, 2.5E-007, -1.32, 5.35, OFF, VERT *OPC?

MEAS1 ON
MEAS1:HIST:SEL 'Histogram1'
MEAS1:CAT HIST
MEAS1:MAIN HME
*OPC?

MEAS2 ON
MEAS2:HIST:SEL 'Histogram1'
MEAS2:CAT HIST
MEAS2:MAIN HSTD
*OPC?

MEAS1:RES:ACT?
MEAS2:RES:ACT?
```

### Exporting histogram data to file

The example writes the absolute data values of Histogram1 to `C:\Histograms\Hist1.xml` in XML format.

Command description in Chapter 24.16.5, "Waveform histogram export to file", on page 1622.

```
EXPort:HISTogram:SELect 'Histogram1'
EXPort:HISTogram:INCidence ABS
EXPort:HISTogram:NAME 'C:\Histograms\Hist1.xml'
EXPort:HISTogram:SAVE
*OPC?
```

### Transferring histogram data

The example transfers the absolute values of Histogram1 to a controlling computer in ASCII format.

Command description in Chapter 24.16.5, "Waveform histogram export to file", on page 1622.

```
EXP:HIST:SEL 'Histogram1'
EXP:HIST:INC ABS
*OPC?
FORM ASC
EXP:HIST:DATA?
<--0,0,0,0,0,2037,5754804,4683496,3100169,2874565,...
```

#### 24.3.2.4 Long-term measurements

In the following example `*OPC` prevents overlapping execution of asynchronous commands.


- Exporting long-term measurement data to file ..... 1178
- Transferring long-term measurement data ..... 1178


##### Exporting long-term measurement data to file

The example writes the long-term data of Meas1 to `C:\Measurements\Meas1.csv` in CSV format.

Command description in Chapter 24.16.7, "Long term measurement results and measurement histogram export to file", on page 1625.

```
EXPort:MEASurement:SEL MEAS1
EXPort:MEASurement:TYPE LONGTERM
EXPort:MEASurement:NAME 'C:\Measurements\Meas1.csv'
EXPort:MEASurement:SAVE
*OPC?
```

##### Transferring long-term measurement data

The example transfers the long-term data of Meas1 to a controlling computer in ASCII format.

Command description in Chapter 24.16.7, "Long term measurement results and measurement histogram export to file", on page 1625.

```
MEASurement:LTM ON
MEASurement:STAT ON
EXPort:MEASurement:SELect MEAS1
EXPort:MEASurement:TYPE LONGTERM
*OPC?
FORM ASC
EXPort:MEASurement:DATA?
<--50,0.24901185771,0.24731225296,0.24703557312,0.00069270717936,0,50,....
```

### 24.3.3 Mask testing


- Creating a user mask ..... 1178


#### 24.3.3.1 Creating a user mask

Creates new user mask "MyMask" with one inner segment, and turns on the mask test.

Command description in: Chapter 24.14, "Mask testing", on page 1542.

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

```
MTEST:ADD 'MyMask'
MTEST:SEGM:ADD 'MyMask'
MTEST:SEGM:POIN:ADD 'MyMask', 0
MTEST:SEGM:POIN:X 'MyMask', 0, 0, -20e-9
MTEST:SEGM:POIN:Y 'MyMask', 0, 0, -0.1
MTEST:SEGM:POIN:ADD 'MyMask', 0
MTEST:SEGM:POIN:X 'MyMask', 0, 1, -20e-9
MTEST:SEGM:POIN:Y 'MyMask', 0, 1, 0.1
MTEST:SEGM:POIN:ADD 'MyMask', 0
MTEST:SEGM:POIN:X 'MyMask', 0, 2, 20e-9
MTEST:SEGM:POIN:Y 'MyMask', 0, 2, 0.1
MTEST:SEGM:POIN:ADD 'MyMask', 0
MTEST:SEGM:POIN:X 'MyMask', 0, 3, 20e-9
MTEST:SEGM:POIN:Y 'MyMask', 0, 3, -0.1
MTEST:SEGM:REG 'MyMask', 0, INNER
MTEST:STAT 'MyMask', ON
*OPC?
```

### 24.3.4 Search

- Searching for a pulse of specified width ..... 1179

#### 24.3.4.1 Searching for a pulse of specified width

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

Search for positive pulses with pulse width 12 ± 10 µs (2 µs to 22 µs).

Command description in: Chapter 24.15, "Search", on page 1565.

The usage of asynchronous commands is described in www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization".

```
SEAR:ADD 'MySearch'                  // Create a new search
SEAR:TRIG:WIDT:STAT 'MySearch',1     // Configure search type
SEAR:SOUR 'MySearch',M1              // Configure search source - here Math1
SEAR:TRIG:WIDT:RANG 'MySearch',WITH  // Configure search parameters
SEAR:TRIG:WIDT:WIDT 'MySearch',5e-4  // Configure search parameters
SEAR:TRIG:WIDT:DELT 'MySearch',5e-6  // Configure search parameters
SEAR:RES:LIM 'MySearch',1            // Set number of result lines in table to 1
*OPC?
SEAR:ALL 'MySearch';                // Initiate search for all events, asynchr. command
```

### 24.3.5 Data management

- Saving a screenshot to file ..... 1180
- Exporting waveform data to file ..... 1180
- Exporting measurement results to file ..... 1186

#### 24.3.5.1 Saving a screenshot to file

Saves three display images in png format to the files `Print.png`, `Print_001.png`, and `Print_002.png` in the directory `C:\Temp`. To get a correct screenshot, turn on the display first.

Command description in: Chapter 24.16.8, "Screenshots", on page 1627.

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

```
SYST:DISP:UPD ON
HCOP:DEST 'MMEM'
HCOP:DEV:LANG PNG
*OPC?
MMEM:NAME 'C:\Temp\Print.png'
HCOP:IMMediate; *OPC?
HCOP:IMM:NEXT; *OPC?
HCOP:IMM:NEXT; *OPC?
```

#### 24.3.5.2 Exporting waveform data to file

Command description in:

- Chapter 24.16.4, "Waveform data export to file", on page 1613
- Chapter 24.16.1, "Instrument settings", on page 1604
- Chapter 24.10.4, "History", on page 1405

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

- Exporting a single waveform to XML file ..... 1180
- Exporting raw data of a single waveform to BIN file ..... 1181
- Exporting raw data of a measurement gate to BIN file ..... 1181
- Exporting interleaved x/y data of a single waveform to CSV file ..... 1182
- Exporting interleaved x/y data of a zoom to CSV file ..... 1182
- Exporting multiple running acquisitions of a single waveform to XML file ..... 1183
- Exporting a single acquisition of the history to BIN file ..... 1183
- Exporting multiple acquisition of the history to XML file ..... 1184
- Exporting and reconstructing multiple raw acquisitions of the history file ..... 1184

##### Exporting a single waveform to XML file

Saves a single analog waveform completely to an XML file. Data logging is off.

The usage of asynchronous commands is described in www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization".

```
STOP;*OPC?
EXPort:WAVeform:FASTexport ON
CHANnel1:WAVeform1:STATe 1
*OPC?
RUNSingle;*OPC?
```

```
EXPort:WAVeform:SOURce C1W1
EXPort:WAVeform:SCOPe WFM
EXPort:WAVeform:NAME 'C:\Data\DataExportWfm_analog.xml'
EXPort:WAVeform:RAW OFF
EXPort:WAVeform:INCXvalues OFF
EXPort:WAVeform:DLOGging OFF
MMEM:DEL 'C:\Data\DataExportWfm_analog.*'
EXPort:WAVeform:SAVE
*OPC?
MMEM:DATA? 'C:\Data\DataExportWfm_analog.xml'
MMEM:DATA? 'C:\Data\DataExportWfm_analog.wfm.xml'
```

### Exporting raw data of a single waveform to BIN file

Saves the data of a single analog waveform in integer 8-bit format (raw data) to a BIN file. Data logging is off.

Data conversion is described in "Raw (ADC direct)" on page 493.

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

```
STOP;*OPC?
EXPort:WAVeform:FASTexport ON
CHANnel1:WAVeform1:STATe 1
RUNSingle; *OPC?
EXPort:WAVeform:SOURce C1W1
EXPort:WAVeform:SCOPe WFM
EXPort:WAVeform:NAME 'C:\Data\DataExportWfm_analog.bin'
EXPort:WAVeform:RAW ON
EXPort:WAVeform:INCXvalues OFF
EXPort:WAVeform:DLOGging OFF
MMEM:DEL 'C:\Data\DataExportWfm_analog.*'
EXPort:WAVeform:SAVE
*OPC?
MMEM:DATA? 'C:\Data\DataExportWfm_analog.bin'
MMEM:DATA? 'C:\Data\DataExportWfm_analog.wfm.bin'
```

### Exporting raw data of a measurement gate to BIN file

Saves the data of a measurement gate in integer 8-bit format (raw data) to a BIN file. Data logging is off.

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

```
STOP;*OPC?
EXPort:WAVeform:FASTexport ON
CHANnel1:WAVeform1:STATe 1
MEASurement2:CATegory AMPT
MEASurement2:MAIN MEAN
MEASurement2:ENABle 1
MEASurement2:SOURce C1W1
```

```
MEASurement2:GATE:MODE ABS
MEASurement2:GATE:ABS:STARt -0.00012
MEASurement2:GATE:ABS:STOP -5e-06
MEASurement2:GATE:STATe On
EXPort:WAVeform:SOURce C1W1
EXPort:WAVeform:SCOPe GATE
EXPort:WAVeform:MEAS Meas2
*OPC?
RUNSingle;*OPC?
EXPort:WAVeform:NAME 'C:\Data\DataExportWfm_analog.bin'
EXPort:WAVeform:RAW ON
EXPort:WAVeform:INCXvalues OFF
EXPort:WAVeform:DLOGging OFF
MMEM:DEL 'C:\Data\DataExportWfm_analog.*'
EXPort:WAVeform:SAVE
*OPC?
MMEM:DATA? 'C:\Data\DataExportWfm_analog.bin'
MMEM:DATA? 'C:\Data\DataExportWfm_analog.wfm.bin'
```

### Exporting interleaved x/y data of a single waveform to CSV file

Saves the x- and y- values of a single analog waveform to a CSV file. Data logging is off.

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

```
STOP;*OPC?
EXPort:WAVeform:FASTexport ON
CHANnel1:WAVeform1:STATe 1
*OPC?
RUNSingle;*OPC?
EXPort:WAVeform:SOURce C1W1
EXPort:WAVeform:SCOPe WFM
EXPort:WAVeform:NAME 'C:\Data\DataExportWfm_analog.csv'
EXPort:WAVeform:RAW OFF
EXPort:WAVeform:INCXvalues ON
EXPort:WAVeform:DLOGging OFF
MMEM:DEL 'C:\Data\DataExportWfm_analog.*'
EXPort:WAVeform:SAVE
*OPC?
MMEM:DATA? 'C:\Data\DataExportWfm_analog.csv'
MMEM:DATA? 'C:\Data\DataExportWfm_analog.wfm.csv'
```

### Exporting interleaved x/y data of a zoom to CSV file

Saves the x- and y- values that is displayed in a zoom diagram to a CSV file. Data logging is off.

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

```
STOP;*OPC?
EXPort:WAVeform:FASTexport ON
CHANnel1:WAVeform1:STATe 1
LAYout:ZOOM:ADD 'Diagram1',HORIZONTAL,OFF,-0.00012,-5e-06,0.308,-0.092,'ExportAreaZoom'
*OPC?
RUNSingle;*OPC?
EXPort:WAVeform:SOURce C1W1
EXPort:WAVeform:SCOPe ZOOM
EXPort:WAVeform:ZOOM 'Diagram1', 'ExportAreaZoom'
EXPort:WAVeform:NAME 'C:\Data\DataExportWfm_analog.csv'
EXPort:WAVeform:RAW OFF
EXPort:WAVeform:INCXvalues ON
EXPort:WAVeform:DLOGging OFF
MMEM:DEL 'C:\Data\DataExportWfm_analog.*'
EXPort:WAVeform:SAVE
*OPC?
MMEM:DATA? 'C:\Data\DataExportWfm_analog.csv'
MMEM:DATA? 'C:\Data\DataExportWfm_analog.wfm.csv'
```

### Exporting multiple running acquisitions of a single waveform to XML file

Saves the data of 5 subsequent acquisitions of a single analog waveform to an XML file. Data logging is on.

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

```
STOP;*OPC?
EXPort:WAVeform:FASTexport ON
CHANnel1:WAVeform1:STATe 1
EXPort:WAVeform:SOURce C1W1
EXPort:WAVeform:SCOPe WFM
EXPort:WAVeform:NAME 'C:\Data\DataExportWfm_analog.xml'
EXPort:WAVeform:RAW OFF
EXPort:WAVeform:INCXvalues OFF
*OPC?
ACQuire:COUNt 5
EXPort:WAVeform:DLOGging ON
MMEM:DEL 'C:\Data\DataExportWfm_analog.*'
RUNSingle;*OPC?
MMEM:DATA? 'C:\Data\DataExportWfm_analog.xml'
MMEM:DATA? 'C:\Data\DataExportWfm_analog.wfm.xml'
```

### Exporting a single acquisition of the history to BIN file

Saves the oldest acquisition of the history to a BIN file. Data logging is off.

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

```
STOP;*OPC?
EXPort:WAVeform:FASTexport ON
CHANnel1:WAVeform1:STATe 1
```

```
EXPort:WAVeform:SOURce C1W1
EXPort:WAVeform:SCOPe WFM
EXPort:WAVeform:NAME 'C:\Data\DataExportWfm_analog.bin'
EXPort:WAVeform:RAW OFF
EXPort:WAVeform:INCXvalues OFF
EXPort:WAVeform:DLOGging OFF
*OPC?
MMEM:DEL 'C:\Data\DataExportWfm_analog.*'
ACQuire:COUNt 5             //Acquire 5 waveforms
RUNSingle;*OPC?
CHANnel1:WAV1:HISTory:STATe ON
CHANnel1:WAV1:HISTory:CURRent -4;*OPC?  //Oldest waveform of 5 has index -4
EXPort:WAVeform:SAVE
*OPC?
MMEM:DATA? 'C:\Data\DataExportWfm_analog.bin'
MMEM:DATA? 'C:\Data\DataExportWfm_analog.wfm.bin'
```

### Exporting multiple acquisition of the history to XML file

Saves the data of 5 subsequent acquisitions of the history to an XML file. Data logging is on.

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

```
STOP;*OPC?
EXPort:WAVeform:FASTexport ON
CHANnel1:WAVeform1:STATe 1
EXPort:WAVeform:SOURce C1W1
EXPort:WAVeform:SCOPe WFM
EXPort:WAVeform:NAME 'C:\Data\DataExportWfm_analog.xml'
EXPort:WAVeform:RAW OFF
EXPort:WAVeform:INCXvalues OFF
EXPort:WAVeform:DLOGging ON
*OPC?
MMEM:DEL 'C:\Data\DataExportWfm_analog.*'
ACQuire:COUNt 5               //Acquire 5 waveforms
RUNSingle;*OPC?
CHANnel1:WAV1:HISTory:STATe ON
CHANnel1:WAV1:HISTory:STARt -4
CHANnel1:WAV1:HISTory:STOP 0
CHANnel1:WAV1:HISTory:REPLay OFF
CHANnel1:WAV1:HISTory:PLAY
*OPC?
MMEM:DATA? 'C:\Data\DataExportWfm_analog.xml'
MMEM:DATA? 'C:\Data\DataExportWfm_analog.wfm.xml'
```

### Exporting and reconstructing multiple raw acquisitions of the history file

This example captures and exports 100 waveforms acquired in fast segmentation mode (minimize blind time). The data is transferred in blocks using SCPI commands.

In the following example *OPC prevents overlapping execution of asynchronous commands.

```
// --- Set data format to signed integers, 1 byte
FORMat:DATA INT,8

// --- Create history data (assuming on channel 1) ---
CHANnel1:WAVeform1:STATe 1           //turn on channel 1
ACQuire:COUNt 100                    //acquire 100 waveforms
ACQuire:SEGMented:STATe ON; *OPC?
//turn on fast segmentation, acquisition is started and data is stored in the memory


// --- Set Export variables and export data ---
EXPort:WAVeform:SOURce C1W1         //specify source for data export
EXPort:WAVeform:SCOPe WFM           //specify range: complete acq. time
MMEM:DEL "C:\Users\Public\Documents\Rohde-Schwarz\RTx\Temp\DataExportWfm_analog.*"
        //delete data from previous runs of this script
EXPort:WAVeform:NAME "C:\Users\Public\Documents\Rohde-Schwarz\RTx\Temp\DataExpWfm.bin"
        //store data in this path and filename
EXPort:WAVeform:RAW ON
//export as raw ADC integer values (saves memory) --> data needs to be converted later
EXPort:WAVeform:INCXvalues OFF
//disable time values in data file. Time can be constructed from the header file
CHANnel1:WAV1:HISTory:STATe ON      //switch to history view
EXPort:WAVeform:DLOGging ON         //enable data logging & history
EXPort:WAVeform:TIMestamps ON       //enable relative time stamp for each acq.
CHANnel1:WAV1:HISTory:STARt -99     //oldest waveform of n acq. has index =(-1)*(n-1)
CHANnel1:WAV1:HISTory:STOP 0        //newest waveforms has index 0
CHANnel1:WAV1:HISTory:REPLay OFF
CHANnel1:WAV1:HISTory:PLAY; *OPC?   //exports waveforms to defined file location.
//2 files are created: header file *.bin and waveform data file *.Wfm.bin


// --- Put files into output buffer of scope and collect ---
// The following code lines are mostly Pseudo Code.
//Sorting out the binary waveform data is more complex and require additional coding
binaryFormat = '1 byte'
//Pseudo Code, tell your language how to interpret the binary data,
//e.g. 'int8' for MATLAB
MMEM:DATA? "C:\Users\Public\Documents\Rohde-Schwarz\RTx\Temp\DataExpWfm.bin"
//Put header file into output buffer
header = readSCPIBinary(visaInstrument, binaryFormat);
//Pseudo Code, use appropriate command from your programming language,
//e.g. binblockread in MATLAB
MMEM:DATA? "C:\Users\Public\Documents\Rohde-Schwarz\RTx\Temp\DataExpWfm.Wfm.bin"
//Put data file into output buffer
//(your input buffer of the VISA resource might need to be increased)
wfmRaw = readSCPIBinary(visaInstrument, binaryFormat)
//Pseudo Code, use appropriate command from your programming language,
```

```
//e.g. binblockread in MATLAB.
//Note: Sort different acquisitions into an array separately after file transfer.


// --- Convert raw ADC values into voltage floating point values ---
// header is assumed to be a struct and the members are accessed via "." syntax.
vertOffsetByPosition = header.VerticalScale * header.VerticalPosition
conversionFactor = (1/header.NofQuantisationLevels) *
  header.VerticalScale * header.VerticalDivisionCount
for(i = 0; i<100; i++){
    wfmVolt(i) = wfmRaw(i) * conversionFactor +
      header.VerticalOffset - vertOffsetByPosition
}
// Note: Depending on settings, the waveform can contain more samples than the
//record length. Remove leading and trailing samples from the waveform.
```

#### 24.3.5.3 Exporting measurement results to file

See:

- "Exporting histogram data to file" on page 1177
- "Transferring histogram data" on page 1177
- "Exporting long-term measurement data to file" on page 1178
- "Transferring long-term measurement data" on page 1178

### 24.3.6 Deembedding

#### 24.3.6.1 Adding cables to the measurement setup

The example adds a cable to each channel and assigns an S-parameter file for each cable. Commands are explained in Chapter 24.24, "Deembedding (option R&S RTP-K121/-K122)", on page 2432.

```
// Preset de-embedding, because *RST does not affect de-embedding
:DEEMbedding1:PRESet    // Ch1
:DEEMbedding2:PRESet    // Ch2
:DEEMbedding3:PRESet    // Ch3
:DEEMbedding4:PRESet    // Ch4

// Add cables to each channel, cable is added next to DUT at index 2 (DUT = index 1)
:DEEMbedding1:ADD 2,cabl    // Ch1
:DEEMbedding2:ADD 2,cabl    // Ch2
:DEEMbedding3:ADD 2,cabl    // Ch3
:DEEMbedding4:ADD 2,cabl    // Ch4

// Assign corresponding S2P file to channel
:DEEMbedding1:COMPonent2:SPONE 'C:\Rtx\Deembedding\RT-ZA17-1806671-0005-A.s2p'
```

```
:DEEMbedding2:COMPonent2:SPONE 'C:\Rtx\Deembedding\RT-ZA17-1806671-0005-B.s2p'
:DEEMbedding3:COMPonent2:SPONE 'C:\Rtx\Deembedding\RT-ZA17-1806671-0006-A.s2p'
:DEEMbedding4:COMPonent2:SPONE 'C:\Rtx\Deembedding\RT-ZA17-1806671-0006-B.s2p'
```

## 24.3.7 Protocol analysis

### 24.3.7.1 RFFE (option R&S RTP-K40)

#### Configuring RFFE bus

Example 1: 1.8 V bus

In the following example `*OPC` prevents overlapping execution of asynchronous commands.

```
//Configure source and thresholds for 1.8 V bus
STOP;*OPC?
BUS1:TYPE RFFE
BUS1:LABel "RFFE Test 2"
BUS1:RFFE:CLOCk:SOUR C2W1
BUS1:RFFE:DATA:SOUR C1W1
BUS:RFFE:PRESet V18

//Load a label list and switch on
BUS1:NEWList 'C:\Protocols\RFFE.csv'
BUS1:SYMBols ON

//Initiate single sweep
CHANnel1:STATe ON
CHANnel2:STATe ON
BUS1 ON
RUNSingle;*OPC?
```

Example 2: User-defined thresholds with threshold coupling

```
//Configure source and user-defined thresholds
STOP;*OPC?
BUS1:TYPE RFFE
BUS1:LABel "RFFE Test 1"
BUS1:RFFE:CLOCk:SOUR C2W1
BUS1:RFFE:DATA:SOUR C1W1
BUS1:RFFE:COUPling ON
BUS1:RFFE:CLOCk:THReshold:HYSTeresis 0.2
BUS1:RFFE:CLOCk:THReshold:HIGH 0.72
BUS1:RFFE:CLOCk:THReshold:LOW  0.48
```

### Triggering on RFFE bus

Trigger on sequence start of sequences that have the slave address 0010, 0011, or 0100.

```
//Set trigger source to serial bus
TRIGger1:SOURce SBUS
TRIGger:MODE NORMal

//Trigger on sequence start of all sequences with slave address range
TRIGger1:RFFE:TYPE STARt
TRIG1:RFFE:SADD:CONDition INR
TRIG1:RFFE:SADD:MIN #H2  //SAD = 0010
TRIG1:RFFE:SADD:MAX #H4  //SAD = 0100
```

### Searching RFFE data

Search for sequence ends of sequences if last bit of slave address is 1.

```
STOP;*OPC?
SEARch:ADD 'RFFESearch1'
SEARch:SOURce 'RFFESearch1',SBUS1
SEARch:TRIGger:RFFE:TYPE 'RFFESearch1',STOP
SEARch:TRIGger:RFFE:SADD:COND 'RFFESearch1', EQU
SEARch:TRIGger:RFFE:SADD:MIN 'RFFESearch1', 'XXX1'
RUNSingle;*OPC?              //asynchronous command
SEARch:ALL 'RFFESearch1';*OPC?
SEARch:RESult:RFFE:FCOunt? 'RFFESearchCh1'

SEARch:RESult:RFFE:SEQuence1:STATe?  'RFFESearchCh1'
<--OK    //First sequence is valid
SEARch:RESult:RFFE:SEQuence1:TYPE?  'RFFESearchCh1'
<--ERRL  //Extended Register Read Long sequence
```

## 24.4 Frequently used parameters and suffixes

This chapter describes in general those parameters and suffixes that are used in several subsystems.

### 24.4.1 Waveform suffix

The numeric waveform suffix is used in some commands, for example, to indicate the source waveform number from which the reference level is taken, and to assign color tables to waveforms.

Depending on the command, not all suffix values are supported. For example, in REFLevel commands, only suffixes 2 to 21 are allowed. The range of supported suffix numbers is indicated in the description of the individual commands.

**NOTICE**

**Suffix 1**

Suffix 1 means that no waveform is assigned. The first waveform C1W1 corresponds to suffix number 2.

| Waveform number | Description |
|---|---|
| 1 | None |
| 2 | C1W1 (channel 1, waveform 1) |
| 3 | C1W2 (channel 1, waveform 2) |
| 4 | C1W3 (channel 1, waveform 3) |
| 5 | C2W1 (channel 2, waveform 1) |
| 6 | C2W2 (channel 2, waveform 2) |
| 7 | C2W3 (channel 2, waveform 3) |
| 8 | C3W1 (channel 3, waveform 1) |
| 9 | C3W2 (channel 3, waveform 2) |
| 10 | C3W3 (channel 3, waveform 3) |
| 11 | C4W1 (channel 4, waveform 1) |
| 12 | C4W2 (channel 4, waveform 2) |
| 13 | C4W3 (channel 4, waveform 3) |
| 14 to 17 | Math waveforms: M1, M2, M3, M4<br>M5 to M8: suffixes 119 to 122 |
| 18 to 21 | Reference waveforms: R1, R2, R3, R4 |
| 22 to 25 | XY-waveforms: XY1, XY2, XY3, XY4 |
| 26 to 33 | Measurement results: MRESult1, MRESult2, MRESult3, MRESult4, MRESult5, MRESult6, MRESult7, MRESult8<br>MRESult1 are the results of measurement group 1 and so on. |
| 34 to 35 | not used |
| 36 to 39 | Serial buses: SBUS1, SBUS2, SBUS3, SBUS4 |
| 40 to 55 | Digital channels: D0 to D15 (option R&S RTP-B1) |
| 56 to 59 | Digital buses: MSO1, MSO2, MSO3, MSO4 (option R&S RTP-B1) |
| 60 | not used |
| 61 to 68 | Track waveforms: TRK1, TRK2, TRK3, TRK4, TRK5, TRK6, TRK7, TRK8 |

| Waveform number | Description |
|---|---|
| 69 to 71 | not used |
| 72 to 75 | Spectrograms: SG1, SG2, SG3, SG4. Available for option R&S RTP-K37.<br>SG5 to SG8: suffixes 119 to 122 . Available for option R&S RTP-K133. |
| 76 to 83 | Timeline spectrums: SG1TL1, SG1TL2, SG2TL1, SG2TL2, SG3TL1, SG3TL2, SG4TL1, SG4TL2. Available for option R&S RTP-K37<br>TL1 is timeline 1, TL2 is timeline 2<br>SG5TL1 to SG8TL2: suffixes 123 to 130. Available for option R&S RTP-K133. |
| 84 to 87 | Eye diagrams: EYE1, EYE2, EYE3, EYE4<br>Available for eye diagram analysis with option R&S RTP-K91 |
| 88 to 91 | Voltage input channels of multi-channel probe R&S RT-ZVC (probe 1)<br>R&S RT-ZVC04: Z1V1 \| Z1V2 \| Z1V3 \| Z1V4<br>R&S RT-ZVC02: Z1V1 \| Z1V2. Suffixes 90 to 91 are not available. |
| 92 to 95 | Current input channels of multi-channel probe R&S RT-ZVC (probe 1)<br>R&S RT-ZVC04: Z1I1 \| Z1I2 \| Z1I3 \| Z1I4<br>R&S RT-ZVC02: Z1I1 \| Z1I2. Suffixes 94 to 95 are not available. |
| 96 to 99 | Voltage input channels of multi-channel probe R&S RT-ZVC (probe 2)<br>R&S RT-ZVC04: Z2V1 \| Z2V2 \| Z2V3 \| Z2V4<br>R&S RT-ZVC02: Z2V1 \| Z2V2. Suffixes 98 to 99 are not available. |
| 100 to 103 | Current input channels of multi-channel probe R&S RT-ZVC (probe 2)<br>R&S RT-ZVC04: Z2I1 \| Z2I2 \| Z2I3 \| Z2I4<br>R&S RT-ZVC02: Z2I1 \| Z2I2. Suffixes 102 to 103 are not available. |
| 104 to 107 | not used |
| 108 to 111 | Resulting waveforms of differential deembedding: DIFF1 \| DIFF2 \| COMMON1 \| COMMON2 |
| 112 to 114 | Resulting waveforms of TDR/TDT analysis (option R&S RTP-K130): TDRZ \| TDRH \| TDRR |
| 115 to 118 | Math waveforms: M5, M6, M7, M8<br>M1 to M4: suffixes 14 to 17 |
| 119 to 122 | Spectrograms: SG5, SG6, SG7, SG8. Available for option R&S RTP-K133.<br>SG1 to SG4: suffixes 72 to 75. Available for option R&S RTP-K37. |
| 123 to 130 | Timeline spectrums: SG5TL1, SG5TL2, SG6TL1, SG6TL2, SG7TL1, SG7TL2, SG8TL1, SG8TL2. Available for option R&S RTP-K133.<br>TL1 is timeline 1, TL2 is timeline 2<br>SG1TL1 to SG4TL2: suffixes 76 to 83. Available for option R&S RTP-K37. |
| 131 to 140 | HP1 \| HP2 \| HP3 \| HP4 \| HP5 \| HP6 \| HP7 \| HP8 \| HP9 \| HP10<br>Available for option R&S RTP-K133. |
| 141 | not used |
| 142 to 144 | BATHtub1 \| BMEas1 \| SRESponse1<br>Bathtub diagram, bathtub measurement, and step response of 1st advanced jitter measurement, option R&S RTP-K133/K134. |

| Waveform number | Description |
|---|---|
| 145 to 151 | SRHist1 \| SDHist1 \| TJHist1 \| DJHist1 \| DDJHist1 \| PJHist1 \| RJOHist1<br>Histograms of 1st advanced jitter measurement, option R&S RTP-K133/K134.<br>SR: symbol rate, SD: symbol duration, TJ: total jitter, DDJ: data-dependent jitter, PJ: periodic jitter, RJO: random jitter. |
| 152, 154- 158 | SRTRack1 \| TJTRack1 \| DJTRack1 \| DDJTrack1 \| PJTRack1 \| RJOTRack1<br>Tracks of 1st advanced jitter measurement, option R&S RTP-K133/K134. |
| 153 | not used |
| 159 to 163 | TJSPectrum1 \| DDJSpectrum1 \| PJOSPectrum1 \| RJOSPectrum1 \| SEYE1<br>Spectrograms and synthetic eye diagram of 1st advanced jitter measurement, option R&S RTP-K133/K134. |
| 164 | not used |
| 165 to 167 | BATHtub2 \| BMEas2 \| SRESponse2<br>Bathtub diagram, bathtub measurement, and step response of 2nd advanced jitter measurement, option R&S RTP-K133/K134. |
| 168 to 174 | SRHist2 \| SDHist2 \| TJHist2 \| DJHist2 \| DDJHist2 \| PJOHist2 \| RJOHist2<br>Histograms of 2nd advanced jitter measurement, option R&S RTP-K133/K134. |
| 175-177, 179-181 | SRTRack2 \| SDTRack2 \| TJTRack2 \| DDJTrack2 \| PJOTrack2 \| RJOTrack2<br>Tracks of 1st advanced jitter measurement, option R&S RTP-K133/K134. |
| 178 | not used |
| 182 to 186 | TJSPectrum2 \| DDJSpectrum2 \| PJOSPectrum2 \| RJOSPectrum2 \| SEYE2<br>Spectrograms and synthetic eye diagram of 2nd advanced jitter measurement, option R&S RTP-K133/K134. |
| 187 to 188 | NBAThtub1 \| NBMeas1<br>Noise bathtub diagram and noise bathtub measurement of 1st noise measurement, option R&S RTP-K134. |
| 189 to 193 | TNHist1 \| DNHist1 \| DDNHist1 \| PNHist1 \| RNOHist1 \|<br>Histograms of 1st noise measurement, option R&S RTP-K134.<br>TN: total noise, DN: deterministic noise, DDN: data-dependent noise, PJ: periodic noise, RNO: random noise. |
| 194, 196 to 198 | TNTRack1 \| DDNTrack1 \| PNTRack1 \| RNOTRack1<br>Tracks of 1st noise measurement, option R&S RTP-K134. |
| 195 | not used |
| 199 to 202 | TNSPectrum1 \| DDNSpectrum1 \| PNSPectrum1 \| RNOSPectrum1<br>Spectrograms of 1st noise measurement, option R&S RTP-K134. |
| 203 to 204 | NBAThtub2 \| NBMeas2<br>Noise bathtub diagram and noise bathtub measurement of 2nd noise measurement, option R&S RTP-K134. |
| 205 to 209 | TNHist2 \| DNHist2 \| DDNHist2 \| PNHist2 \| RNOHist2<br>Histograms of 2nd noise measurement, option R&S RTP-K134. |

| Waveform number | Description |
|---|---|
| 210, 212 to 214 | TNTRack2 \| DDNTrack2 \| PNTRack2 \| RNOTRack2<br>Tracks of 2nd noise measurement, option R&S RTP-K134. |
| 211 | not used |
| 215 to 218 | TNSPectrum2 \| DDNSpectrum2 \| PNSPectrum2 \| RNOSpectrum2<br>Spectrograms eye diagram of 2nd noise measurement, option R&S RTP-K134. |
| 219 to 221 | PJDHist1 \| BUJHist1 \| RJHist1<br>Histograms of 1st jitter measurement, option R&S RTP-K134.<br>BUJ: (Other) bounded uncorrelated jitter. |
| 222 | PJDTrack1<br>Tracks of 1st data dependent and periodic jitter noise measurement, option R&S RTP-K134. |
| 223 | PJDSpectrum1<br>Spectrograms of 1st data dependent and periodic jitter noise measurement, option R&S RTP-K134. |
| 224 to 226 | PJDHist2 \| BUJHist2 \| RJHist2<br>Histograms of 2nd noise measurement, option R&S RTP-K134. |
| 227 | PJDTrack2<br>Tracks of 2nd data dependent and periodic jitter noise measurement, option R&S RTP-K134. |
| 228 | PJDSpectrum2<br>Spectrograms of 2nd data dependent and periodic jitter measurement, option R&S RTP-K134. |
| 229 to 231, 234 to 236 | PNDHist1 \| BUNHist1 \| RNHist1 \| PNDHist2 \| BUNHist2 \| RNHist2<br>Histograms of noise measurement, option R&S RTP-K134.<br>BUN: (Other) bounded uncorrelated noise. |
| 232 | PNDTrack1<br>Tracks of 1st data dependent and periodic noise measurement, option R&S RTP-K134. |
| 233 | PNDSpectrum1<br>Spectrograms of 1st data dependent and periodic noise measurement, option R&S RTP-K134. |
| 237 | PNDTrack2<br>Tracks of 2nd data dependent and periodic noise measurement, option R&S RTP-K134. |
| 238 | PNDSpectrum2<br>Spectrograms of 2nd data dependent and periodic noise measurement, option R&S RTP-K134. |

### 24.4.2 Waveform parameter

Many commands requires one of the waveforms to be specified as source. The following table lists all waveforms. For each command using a waveform parameter, the available waveforms are specified in the command description.

| Waveform | Description |
|---|---|
| C1W1 \| C1W2 \| C1W3 | Channel 1 waveforms |
| C2W1 \| C2W2 \| C2W3 | Channel 2 waveforms |
| C3W1 \| C3W2 \| C3W3 | Channel 3 waveforms |
| C4W1 \| C4W2 \| C4W3 | Channel 4 waveforms |
| M1 \| M2 \| M3 \| M4 \| M5 \| M6 \| M7 \| M8 | Math waveforms |
| R1 \| R2 \| R3 \| R4 | Reference waveforms |
| XY1 \| XY2 \| XY3 \| XY4 | XY-waveforms |
| MRESult1 \| MRESult2 \| MRESult3 \| MRESult4 \| MRESult5 \| MRESult6 \| MRESult7 \| MRESult8 | Measurement results |
| SBUS1 \| SBUS2 \| SBUS3 \| SBUS4 | Serial buses |
| D0 \| D1 \| D2 \| D3 \| D4 \| D5 \| D6 \| D7 \| D8 \| D9 \| D10 \| D11 \| D12 \| D13 \| D14 \| D15 | Digital channels (option R&S RTP-B1) |
| MSOB1 \| MSOB2 \| MSOB3 \| MSOB4 | Digital buses (option R&S RTP-B1 |
| TRK1 \| TRK2 \| TRK3 \| TRK4 \| TRK5 \| TRK6 \| TRK7 \| TRK8 | Track waveforms |
| CDRSW1 = CDRSw1, CDRSW2 = CDRSw2 | Generated clock signals |
| SG1 \| SG2 \| SG3 \| SG4 | Spectrograms (option R&S RTP-K37) |
| SG5 \| SG6 \| SG7 \| SG8 | Advanced jitter analysis (option R&S RTP-K133) |
| SG1TL1, SG1TL2, SG2TL1, SG2TL2, SG3TL1, SG3TL2, SG4TL1, SG4TL2 | Spectrograms (option R&S RTP-K37) |
| SG5TL1, SG5TL2, SG6TL1, SG6TL2, SG7TL1, SG7TL2, SG8TL1, SG8TL2 | Advanced jitter analysis (option R&S RTP-K133) |
| EYE1 \| EYE2 \| EYE3 \| EYE4 | Available for eye diagram analysis with option R&S RTP-K91 |
| Z1V1 \| Z1V2 \| Z1V3 \| Z1V4 \| Z1I1 \| Z1I2 \| Z1I3 \| Z1I4 \| Z2V1 \| Z2V2 \| Z2V3 \| Z2V4 \| Z2I1 \| Z2I2 \| Z2I3 \| Z2I4<br>Z1V1 \| Z1V2 \| Z1I1 \| Z1I2 \| Z2V1 \| Z2V2 \| Z2I1 \| Z2I2 | Input channels of multi-channel probe R&S RT-ZVC04<br>Input channels of multi-channel probe R&S RT-ZVC02 |
| DIFF1 \| DIFF2 \| COMMON1 \| COMMON2 | Resulting waveforms of differential deembedding |
| TDRZ \| TDRH \| TDRR | Resulting waveforms of TDR/TDT analysis (option R&S RTP-K130) |
| SEYE1 \| SEYE2 | Available for synthetic eye diagram analysis with option R&S RTP- |

| Waveform | Description |
|---|---|
| TJSPectrum1 \| TJSPectrum2 \| DDJSpectrum1 \| DDJSpectrum2 \| PJSPectrum1 \| PJSPectrum2 \| RJOSPectrum \| RJOSPectrum1 | Spectrograms of the advanced jitter analysis components (option R&S RTP-K133/K134)<br>TJ: total jitter<br>DDJ: data-dependent jitter<br>PJ: periodic jitter<br>RJO: random jitter |
| BATHtub1 \| BATHtub2 | Bathtub diagrams of the advanced jitter analysis components (option R&S RTP-K133/K134) |
| BMEas1 \| BMEas2 | Bathtub measurements of the advanced jitter analysis components (option R&S RTP-K133/K134) |
| SRTRack1 \| SRTRack2 \| SDTRack1 \| SDTRack2 \| TJTRack1 \| TJTRack2 \| DJTRack1 \| DJTRack2 \| DDJTrack1 \| DDJTrack2\| PJOTRack1 \| PJOTRack2 \| RJOTRack1\| RJOTRack2 | Tracks of the advanced jitter analysis components (option R&S RTP-K133/K134)<br>SR: Symbol rate<br>SD: Symbol duration |
| SRESponse1 \| SRESponse2 | Step response of the advanced jitter analysis (option R&S RTP-K133/K134) |
| TNSPectrum1 \| TNSPectrum2 \| DDNSpectrum1 \| DDNSpectrum2 \| PNSPectrum1 \| PNSPectrum2 \| RNOSpectrum1 \| RNOSpectrum2 | Spectrograms of the advanced jitter analysis components (option R&S RTP-K134)<br>TN: total noise<br>DDN: data-dependent noise<br>PN: periodic noise<br>RNO: random noise |
| NBAThtub1 \| NBAThtub2 | Bathtub diagrams of the noise analysis components (option R&S RTP-K134) |
| NBMeas1 \| NBMeas2 | Bathtub measurements of the noise analysis components (option R&S RTP-K134) |
| TNTRack1 \| TNTRack2 \| DNTRack1 \| \| DNTRack2 \| DDNTrack1 \| DDNTrack2 \| PNTRack1 \| PNTRack2 \| RNOTRack1 \| RNOTRack2 | Tracks of the noise analysis components (option R&S RTP-K134) |

### 24.4.3 Slope parameter

The slope parameter is used with several trigger and search condition commands.

| Slope | Description |
|---|---|
| POSitive | Rising edge, that is a positive voltage change. |
| NEGative | Falling edge, that is a negative voltage change |
| EITHer | rising as well as the falling edge. |

### 24.4.4 Polarity parameter

The polarity parameter is used with several trigger and search condition commands.

| Polarity | Description |
|---|---|
| POSitive | Positive going pulses. |
| NEGative | Negative going pulses. |
| EITHer | Both positive and negative going pulses. |

### 24.4.5 Event parameter

The event parameter is used with commands defining an action for mask testing, limit checks and margin checks.

| Event | Description |
|---|---|
| NOACtion | The action is not initiated. |
| SUCCess | The action is initiated if the operation finished successfully:<br>• Limits or margins were not exceeded during the entire measurement<br>• Mask test passed |
| VIOLation | The action is initiated if the operation finished with error:<br>• Limits or margins were violated during the measurement<br>• Mask test failed |

### 24.4.6 Bit pattern parameter

Bit pattern parameter are required with commands triggering on address, identifier, or data pattern.

To set the pattern value, you can use either a numeric parameter as defined in the SCPI standard, or a string parameter.

#### Bit pattern in numeric parameter

In a numeric parameter, the values are listed byte-by-byte, with bytes separated by commas and MSB first. The default numeral format is decimal, other formats can be indicated by a format identifier (#B = binary, #H = hexadecimal, #Q = octal). Currently, no format for signed values is available.

**Example:** Parameter with three bytes, decimal byte values are 10, 20, 30. The examples are given for CAN, the bit pattern in other commands is defined in the same way.

- `TRIGger:CAN:DMIN 10,20,30`
- `TRIGger:CAN:DMIN #B00001010,#B00010100,#B00011110`
- `TRIGger:CAN:DMIN #H0A,#H14,#H1E`
- `TRIGger:CAN:DMIN #Q012,#Q024,#Q036`

#### Bit pattern in string parameter

In a string, the complete binary pattern is written without separation of bytes, for example:

```
TRIGger:CAN:DMIN '000010100001010000011110'
```

Unlike a numeric parameter, the string parameter accepts wildcards for single bits (X = don't care). Whether wildcards can be used or not depends on the remote command. Usually, address and identifier parameter require unique patterns while data parameters may contain wildcards.

Mostly the length of the bit pattern is defined, for example, by the I²C address type, the CAN identifier type, or the data length code. In theses cases, it is recommended that you enter the complete bit pattern. If you enter a shorter pattern, the instrument fills up the pattern with X bits to the right of the defined pattern.

**Example**: You want to trigger on an 11 bit CAN address and enter the bit pattern '11100011' (8 bits only). The instrument uses the pattern '11100011XXX' for triggering.

**Query for a pattern**

The pattern format for the return value of a pattern is defined by the `FORMat:BPATtern` command.

## 24.5 Common commands

Common commands are described in the IEEE 488.2 (IEC 625-2) standard. These commands have the same effect and are employed in the same way on different devices. The headers of these commands consist of "*" followed by three letters. Many common commands are related to the Status Reporting System.

Available common commands:

*CAL? ..... 1197
*CLS ..... 1197
*ESE ..... 1197
*ESR? ..... 1197
*IDN? ..... 1197
*IST? ..... 1198
*OPC ..... 1198
*OPT? ..... 1198
*PCB ..... 1199
*PRE ..... 1199
*PSC ..... 1199
*RCL ..... 1199
*RST ..... 1199
*SAV ..... 1200
*SRE ..... 1200
*STB? ..... 1200
*TRG ..... 1200
*TST? ..... 1200
*WAI ..... 1201

***CAL?**

Starts a self-alignment of the instrument, and then queries a status response. Return values ≠ 0 indicate an error.

**Return values:**

<State> 0: no error
1: alignment failed
2: not aligned, e.g. init
3: device needs longer warmup time before selfalignment can start
4: input signal connected during selfalignment

**Usage:** Query only

**Manual operation:** See "Start Alignment" on page 104

***CLS**

Clear status

Sets the status byte (STB), the standard event register (ESR) and the `EVENt` part of the `QUEStionable` and the `OPERation` registers to zero. The command does not alter the mask and transition parts of the registers. It clears the output buffer.

**Usage:** Setting only

***ESE** <Value>

Event status enable

Sets the event status enable register to the specified value. The query returns the contents of the event status enable register in decimal form.

**Parameters:**

<Value> Range: 0 to 255

***ESR?**

Event status read

Returns the contents of the event status register in decimal form and then sets the register to zero.

**Return values:**

<Contents> Range: 0 to 255

**Usage:** Query only

***IDN?**

Identification

Returns the instrument identification.

**Return values:**

<ID> "Rohde&Schwarz,<device type>,<part number>/<serial number>,<firmware version>"

**Example:** `Rohde&Schwarz,RTP,1803.7000K04/200153,5.2.1.2`
Model assignment:
RTP044B - 1803.7000K04
RTP064B - 1803.7000K06
RTP084B - 1803.7000K08
RTP134B - 1803.7000K13
RTP164B - 1803.7000K16

**Usage:** Query only

---

***IST?**

Individual status query

Returns the contents of the IST flag in decimal form. The IST flag is the status bit which is sent during a parallel poll.

**Return values:**

<ISTflag> 0 | 1

**Usage:** Query only

---

***OPC**

Operation complete

Sets bit 0 in the event status register when all preceding commands have been executed. This bit can be used to initiate a service request. The query writes a "1" into the output buffer when all preceding commands have been executed, which is useful for command synchronization.

---

***OPT?**

Option identification query

Queries the options included in the instrument. For a list of all available options and their description, refer to the data sheet.

**Return values:**

<Options> The query returns a list of options. The options are returned at fixed positions in a comma-separated string. A zero is returned for options that are not installed.

**Usage:** Query only

***PCB** <Address>

Pass control back

Indicates the controller address to which remote control is returned after termination of the triggered action.

**Setting parameters:**
<Address> Range: 0 to 30

**Usage:** Setting only

***PRE** <Value>

Parallel poll register enable

Sets parallel poll enable register to the indicated value. The query returns the contents of the parallel poll enable register in decimal form.

**Parameters:**
<Value> Range: 0 to 255

***PSC** <Action>

Power on status clear

Determines whether the contents of the `ENABle` registers are preserved or reset when the instrument is switched on. Thus a service request can be triggered when the instrument is switched on, if the status registers ESE and SRE are suitably configured. The query reads out the contents of the "power-on-status-clear" flag.

**Parameters:**
<Action> 0 | 1

**0**
The contents of the status registers are preserved.

**1**
Resets the status registers.

***RCL** <Number>

Recall

Loads the instrument settings from an intermediate memory identified by the specified number. The instrument settings can be stored to this memory using the command `*SAV` with the associated number.

It also activates the instrument settings which are stored in a file and loaded using `MMEMory:LOAD:STATe`.

***RST**

Reset

Sets the instrument to a defined default status. The default settings are indicated in the description of commands.

**Usage:** Setting only

---

***SAV** <Number>

Save

Stores the current instrument settings under the specified number in an intermediate memory. The settings can be recalled using the command `*RCL` with the associated number.

To transfer the stored instrument settings to a file, use `MMEMory:STORe:STATe`.

---

***SRE** <Contents>

Service request enable

Sets the service request enable register to the indicated value. This command determines under which conditions a service request is triggered.

**Parameters:**

<Contents> Contents of the service request enable register in decimal form. Bit 6 (MSS mask bit) is always 0.

Range: 0 to 255

---

***STB?**

Status byte query

Reads the contents of the status byte in decimal form.

**Usage:** Query only

---

***TRG**

Trigger

Triggers all actions waiting for a trigger event. In particular, *TRG generates a manual trigger signal. This common command complements the commands of the `TRIGger` subsystem.

`*TRG` corresponds to the `INITiate:IMMediate` command.

**Usage:** Event

---

***TST?**

Self-test query

Initiates self-tests of the instrument and returns an error code.

**Return values:**

<ErrorCode> **integer > 0 (in decimal format)**
An error occurred.
(For details, see the Service Manual supplied with the instrument).

**0**
No errors occurred.

**Usage:** Query only

**Manual operation:** See "Selftest" on page 106

---

***WAI**

Wait to continue

Prevents servicing of the subsequent commands until all preceding commands have been executed and all signals have settled (see also command synchronization and `*OPC`).

**Usage:** Event

## 24.6 General remote settings

This chapter describes commands that have effect on many other remote commands in different applications of the instrument.

FORMat[:DATA] ..... 1201
FORMat:BORDer ..... 1203
FORMat:BPATtern ..... 1203
SYSTem:DISPlay:UPDate ..... 1203
SYSTem:KLOCk ..... 1203
GPIB:ADDRess ..... 1204
GPIB:TERMinator ..... 1204
SYSTem:DISPlay:MESSage:STATe ..... 1204
SYSTem:DISPlay:MESSage[:TEXT] ..... 1204
SYSTem:LANGuage ..... 1205

---

**FORMat[:DATA]** <Format>, [<Length>]

Selects the data type that is used for transmission of data from analog channels, math and reference waveforms, and some measurement results from the instrument to the controlling computer.

The command sets the data format for the following query commands:

- `CHANnel<m>[:WAVeform<n>]:DATA[:VALues]?`
- `CALCulate:MATH<m>:DATA[:VALues]?`
- `REFCurve<m>:DATA[:VALues]?`

- EXPort:HISTogram:DATA? on page 1623
- EXPort:MEASurement:DATA? on page 1626
- MEASurement<m>:TRACk:DATA[:VALues]? on page 1502

**Parameters:**

<Format>,[<Length>] ASCii | REAL,32 | INT,8 | INT,16

**ASCii**
Data values are returned in ASCII format as a list of comma-separated values in floating point format. The length can be omitted. It is 0 which means that the instrument selects the number of digits to be returned. The query returns both values (ASC, 0).

**REAL,32**
The data is stored as binary data (Definite Length Block Data according to IEEE 488.2). Each waveform value is formatted in 32-Bit IEEE 754 Floating Point Format.
The schema of the result string is as follows:
#41024<value1><value2>…<value n> with:
#4 = number of digits (= 4 in the example) of the following number
1024 = number of following data bytes (= 1024 in the example)
<value> = 4-byte floating point values
For large data (≥ 1 GB), the result string starts with "#(data length)". The number inside the parentheses indicates the real data length in bytes.

**INT,8 | INT,16**
Signed integer data with length 8 bit or 16 bit. It defines that CHANnel<m>[:WAVeform<n>]:DATA[:VALues]? returns the raw sample data of the ADC as integers. If format of the waveform data differs from the defined export format, the instrument converts the data to the required length.
The result string has the same schema as the REAL format.
Fot INT,16 you can set the byte order using the FORMat:BORDer command.
Data conversion is described in "Raw (ADC direct)" on page 493.
For digital channel data, math and histogram data, INT formats are not available.
EXPort:WAVeform:INCXvalues must be set OFF.

*RST: ASCii

**Example:**

```
FORMat:DATA REAL,32
FORMat:DATA?
REAL,32
```

**Usage:** SCPI confirmed

**Manual operation:** See "Transfer data format" on page 1152

**FORMat:BORDer** <ByteOrder>

Sets the endianess.

The command is only relevant for raw data export in high definition mode (16 bit word length).

**Parameters:**

<ByteOrder> LSBFirst | MSBFirst

LSB first: little endian, least significant byte first
MSB first: big endian, most significant byte first

*RST: LSBFirst

**Usage:** Asynchronous command

**Manual operation:** See "Byte order" on page 1153

**FORMat:BPATtern** <BitPatternFormat>

Sets the number format for remote bit pattern queries on serial protocols.

**Parameters:**

<BitPatternFormat> DEC | HEX | OCT | BIN | ASCII | ASCii | STRG

ASCII = ASCii

*RST: HEX

**Usage:** Asynchronous command

**Manual operation:** See "Bit pattern format" on page 1153

**SYSTem:DISPlay:UPDate** <Enable>

Defines whether the display is updated while the instrument is in the remote state. If the display is switched off, the normal GUI is replaced by a static image while the instrument is in the remote state. Switching off the display can speed up the measurement. This is the recommended state.

See also: Chapter 23.5.3.2, "Using the display during remote control", on page 1154

**Parameters:**

<Enable> **ON | 1:** Display is shown and updated during remote control
**OFF | 0:** Display shows static image during remote control

**Example:** `SYSTem:DISPlay:UPDate 1`
Switch on the display update.

**SYSTem:KLOCk** <Enable>

Locks or unlocks the local controls of the instrument. This includes the front panel keys, the keyboard, or other local interfaces. except for the "View" button on the display.

**Parameters:**
<Enable> **ON | 1:** Locks the local keys
**OFF | 0:** Keys are unlocked

**Usage:** SCPI confirmed

---

**GPIB:ADDRess** <Address>

Sets the GPIB address of the instrument. Changing the address has major effects on the communication to the remote computer.

**Parameters:**
<Address> Range: 0 to 30
Increment: 1
*RST: 20

**Usage:** Asynchronous command

**Manual operation:** See "Address" on page 1151

---

**GPIB:TERMinator** <Terminator>

Specifies the symbol that is used as a terminator in GPIB communication.

**Parameters:**
<Terminator> LFEoi | EOI
*RST: EOI

**Usage:** Asynchronous command

**Manual operation:** See "Terminator" on page 1152

---

**SYSTem:DISPlay:MESSage:STATe** <DispMessState>

Enables and disables the display of an additonal text in remote control.

To define the text, use `SYSTem:DISPlay:MESSage[:TEXT]`.

**Parameters:**
<DispMessState> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Firmware/software:** Version 2.70

---

**SYSTem:DISPlay:MESSage[:TEXT]** <DisplayMessage>

Defines an additional text that is displayed during remote control operation.

To enable the text display, use `SYSTem:DISPlay:MESSage:STATe` on page 1204.

**Parameters:**
<DisplayMessage> String that contains the text.

**Usage:** Asynchronous command

**Firmware/software:** Version 2.70

---

**SYSTem:LANGuage** <Language>

Defines the remote control behavior of the instrument and sets the remote control command set.

**Parameters:**

<Language> String value. Available values:
'SCPI': R&S RTP remote command set is used.
"DPO7000" or 'TDS540': Compatible remote command set of Tektronix oscilloscopes DPO7000 or TDS540 is used.
If one of these emulation modes is used, you can define alternative responses to the `IDN*?` and `OPT*?` commands on the "Menu" > "Settings" > "System" > "Remote" > "SCPI Emulation" tab.

**Usage:** Asynchronous command

**Options:** R&S RTP--K301

## 24.7 Instrument setup

### 24.7.1 System

SYSTem:PRESet ..... 1205
SYSTem:DATE ..... 1205
SYSTem:TIME? ..... 1206
SYSTem:DEVice:ID? ..... 1206
DIAGnostic:SERVice:COMPutername ..... 1206

---

**SYSTem:PRESet**

Resets the instrument to the factory default settings, to the initial state. Factory settings comprise all instrument settings, including display, intensity and transparency settings. After loading factory defaults, perform a self-alignment to synchronize the signal data.

**Usage:** Event

---

**SYSTem:DATE** <Year>, <Month>, <Day>

Sets the date of the internal calendar.

**Parameters:**

<Year> Year, to be entered as a four-digit number (including the century and millennium information)

<Month> Month, 1 (January) to 12 (December)

<Day> Day, 1 to the maximum number of days in the specified month
*RST: does not affect the date settings

**Example:** `SYSTem:DATE?`
Returned value: `2011,09,13`

**Usage:** SCPI confirmed

**Manual operation:** See "Time settings" on page 74

---

**SYSTem:TIME?** <Hour>, <Minute>, <Second>

Returns the UTC (Universal Time Coordinated) of the internal clock. To define the current local time, use the time zone setting of the operating system

**Example:** `SYSTem:TIME?`
Returned value: `15,09,20`. UTC is 15:09:20.

**Usage:** Query only
SCPI confirmed

**Manual operation:** See "Time settings" on page 74

---

**SYSTem:DEVice:ID?**

Returns the instrument ID - that is the part number and the serial number.

**Return values:**
<ID> String containing the part number and the serial number

**Example:** `1316.1000K24-001122-jT`

**Usage:** Query only

---

**DIAGnostic:SERVice:COMPutername** <ComputerName>

The query returns the computer name that is currently defined. The computer name is required when configuring a network.

The setting command changes the computer name. The change takes effect after the next reboot of the computer.

**Parameters:**
<ComputerName> Name string

**Usage:** Asynchronous command

**Manual operation:** See "Device name" on page 72

## 24.7.2 Appearance

### 24.7.2.1 Colors appearance

DISPlay:COLor:SIGNal<m>:COLor....1207
DISPlay:COLor:SIGNal<m>:ASSign....1207
DISPlay:COLor:SIGNal<m>:USE....1207

**DISPlay:COLor:SIGNal<m>:COLor** <Value>

**Suffix:**
<m> 1...218
Waveform number, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value> Decimal value of the ARGB color. Use the color dialog box on the instrument to get the hex value of the color, and convert the hex value to a decimal value.
0 is fully transparent black.
4278190080 (dec) = FF000000 (hex) is opaque black.
4294967295 (dec) = FFFFFFFF (hex) is opaque white.

**Manual operation:** See "Color" on page 78

**DISPlay:COLor:SIGNal<m>:ASSign** <ColorTable>

Assigns the color table to the specified signal.

**Suffix:**
<m> 1...218
Waveform number, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<ColorTable> Color table name to be assigned to the signal.

**Manual operation:** See "Assigned color table" on page 78

**DISPlay:COLor:SIGNal<m>:USE** <State>

If enabled, the selected waveform is displayed according to its assigned color table.

If this option is disabled, the default color table is used, i.e. the intensity of the specific signal color varies according to the cumulative occurance of the values.

**Suffix:**
<m> 1...218
Waveform number, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<State> ON | OFF

**Manual operation:** See "Use color table" on page 78

### 24.7.2.2 Color tables

DISPlay:COLor:PALette:ADD ..... 1208
DISPlay:COLor:PALette:REMove ..... 1208
DISPlay:COLor:PALette:COUNt? ..... 1208
DISPlay:COLor:PALette:POINt:ADD ..... 1208
DISPlay:COLor:PALette:POINt:INSert ..... 1208
DISPlay:COLor:PALette:POINt:REMove ..... 1209
DISPlay:COLor:PALette:POINt[:VALue] ..... 1209
DISPlay:COLor:PALette:POINt:COUNt? ..... 1209

---

**DISPlay:COLor:PALette:ADD** <Name>

Adds a new color table with the specified name.

**Setting parameters:**
<Name> color table

**Usage:** Setting only

---

**DISPlay:COLor:PALette:REMove** <Name>

Removes the specified color table.

**Setting parameters:**
<Name> color table

**Usage:** Setting only

---

**DISPlay:COLor:PALette:COUNt?**

Queries the number of configured color maps.

**Usage:** Query only

---

**DISPlay:COLor:PALette:POINt:ADD** <PaletteName>

Appends a new row at the end of the color table.

**Setting parameters:**
<PaletteName> color table

**Usage:** Setting only

---

**DISPlay:COLor:PALette:POINt:INSert** <PaletteName>, <PointIndex>

Inserts the entry at the specified index in the color table.

**Setting parameters:**
<PaletteName> color table

<PointIndex> row number in the color table

**Usage:** Setting only

---

**DISPlay:COLor:PALette:POINt:REMove** <PaletteName>, <PointIndex>

Removes the entry with the specified index from the color table.

**Setting parameters:**
<PaletteName> color table

<PointIndex> row number in the color table

**Usage:** Setting only

---

**DISPlay:COLor:PALette:POINt[:VALue]** <ColorTableName>, <Index>,<Position>, <Color>
**DISPlay:COLor:PALette:POINt[:VALue]?** <ColorTableName>, <Index>

Inserts a new entry or queries the specified entry in the specified color table.

**Parameters:**
<Position> Cumulative occurance value

Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

<Color> ARGB value of the color to be used for the table entry.
ARGB=<Opacity(alpha) value><red value><green value><blue value>, in hexadecimal or decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 0

**Parameters for setting and query:**
<ColorTableName> Color table to be edited

<Index> Index (row number) of the new entry in the color table

**Usage:** Asynchronous command

---

**DISPlay:COLor:PALette:POINt:COUNt?** <PaletteName>

Queries the number of entries in the color table.

**Query parameters:**
<PaletteName> color table

**Usage:** Query only

### 24.7.2.3 Grid appearance

DISPlay:DIAGram:GRID ..... 1210
DISPlay:DIAGram:LABels ..... 1210
DISPlay:DIAGram:FINegrid ..... 1210
DISPlay:DIAGram:CROSshair ..... 1210
DISPlay:DIAGram:YFIXed ..... 1211
DISPlay:DIAGram:XFIXed ..... 1211

---

**DISPlay:DIAGram:GRID** <Show>

If enabled, a grid is displayed in the diagram area.

**Parameters:**
<Show> ON | OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show grid" on page 81

---

**DISPlay:DIAGram:LABels** <ShowLabels>

If enabled, labels mark values on the x- and y-axes in specified intervals in the diagram.

**Parameters:**
<ShowLabels> ON | OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show labels" on page 81

---

**DISPlay:DIAGram:FINegrid** <ShowFineScale>

If ON, the crosshair is displayed as a ruler with scale markers. If OFF, the crosshair is shown as dashed lines.

**Parameters:**
<ShowFineScale> ON | OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show fine grid scale" on page 81

---

**DISPlay:DIAGram:CROSshair** <Crosshair>

If selected, a crosshair is displayed in the diagram area. A crosshair allows you to select a specific data point by its co-ordinates.

**Parameters:**
<Crosshair> ON | OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show crosshair" on page 82

---

**DISPlay:DIAGram:YFIXed** <YGridFixed>

If enabled, the horizontal grid lines remain in their position when the position of the curve is changed. Only the values at the grid lines are adapted. This reflects the behavior of traditional oscilloscopes.

**Parameters:**
<YGridFixed> ON | OFF

**Usage:** Asynchronous command

**Manual operation:** See "Keep Y-grid fixed" on page 82

---

**DISPlay:DIAGram:XFIXed** <XGridFixed>

If enabled, the vertical grid lines remain in their position when the horizontal position is changed. Only the values at the grid lines are adapted.

**Parameters:**
<XGridFixed> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Keep X-grid fixed" on page 82

### 24.7.2.4 Smartgrid

The following `LAYout` commands configure the SmartGrid: add and remove diagrams, and assign signals to the diagrams. In manual operation, you configure the SmartGrid by drag and drop.

LAYout:ADD ..... 1211
LAYout:REMove ..... 1213
LAYout:SHOW ..... 1213
LAYout:SIGNal:ASSign ..... 1213
LAYout:SIGNal:UNASsign ..... 1215

---

**LAYout:ADD** <NodeName>, <ParentType>, <InsertBefore>, <FirstSource>, <DiagramName>

Adds a new diagram with a waveform on the screen, in relation to an existing diagram.

**Setting parameters:**
<NodeName> String with the name of the existing diagram

<ParentType> HORizontal | VERTical | TAB
Position of the new diagram in relation to the existing one.
**HORizontal**
Besides the existing diagram

**VERTical**
Above or below the existing diagram

**TAB**
In a new tab in the existing diagram

<InsertBefore> ON | OFF

If on, the new diagram is inserted to the left (for HORizontal), above (for VERTical) or in a tab in front the existing diagram.
HOR, ON = left to the existing diagram, defined in <NodeName>
HOR, OFF = right to the existing diagram
VERT, ON = above the existing diagram
VERT, OFF = below the existing diagram

<FirstSource> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | XY1 | XY2 | XY3 | XY4 | MRESult1 | MRESult2 | MRESult3 | MRESult4 | MRESult5 | MRESult6 | MRESult7 | MRESult8 | SBUS1 | SBUS2 | SBUS3 | SBUS4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | MSOB1 | MSOB2 | MSOB3 | MSOB4 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | BATHtub1 | BMEas1 | SRESponse1 | SRHist1 | SDHist1 | TJHist1 | DJHist1 | DDJHist1 | PJHist1 | RJOHist1 | SRTRack1 | SDTRack1 | TJTRack1 | DJTRack1 | DDJTrack1 | PJTRack1 | RJOTrack1 | TJSPectrum1 | DDJSpectrum1 | PJSPectrum1 | RJOSpectrum1 | SEYE1 | BATHtub2 | BMEas2 | SRESponse2 | SRHist2 | SDHist2 | TJHist2 | DJHist2 | DDJHist2 | PJHist2 | RJOHist2 | SRTRack2 | SDTRack2 | TJTRack2 | DJTRack2 | DDJTrack2 | PJTRack2 | RJOTrack2 | TJSPectrum2 | DDJSpectrum2 | PJSPectrum2 | RJOSpectrum2 | SEYE2 | NBAThtub1 | NBMeas1 | TNHist1 | DNHist1 | DDNHist1 | PNHist1 | RNOHist1 | TNTRack1 | DNTRack1 | DDNTrack1 | PNTRack1 | RNOTrack1 | TNSPectrum1 | DDNSpectrum1 | PNSPectrum1 | RNOSpectrum1 | NBAThtub2 | NBMeas2 | TNHist2 | DNHist2 | DDNHist2 | PNHist2 | RNOHist2 | TNTRack2 | DNTRack2 | DDNTrack2 | PNTRack2 | RNOTrack2 | TNSPectrum2 | DDNSpectrum2 | PNSPectrum2 | RNOSpectrum2 | PJDHist1 | BUJHist1 | RJOHist1 | PJDTrack1 | PJDSpectrum1 | PJDHist2 | BUJHist2 | RJOHist2 | PJDTrack2 | PJDSpectrum2 | PNDHist1 | BUNHist1 | RNOHist1 | PNDTrack1 | PNDSpectrum1 | PNDHist2 | BUNHist2 | RNOHist2 | PNDTrack2 | PNDSpectrum2

Waveform to be displayed in the new diagram, see Chapter 24.4.2, "Waveform parameter", on page 1193.
Spectrum analysis, option R&S RTP-K37: Spectrograms and timeline spectrums are automatically displayed in their own diagrams.

Eye diagram analysis with option R&S RTP-K91: Eye diagrams are automatically displayed in their own diagrams.
TDR/TDT analysis with option R&S RTP-K130: Measurement waveforms are automatically displayed in their own diagrams.
Advanced jitter analysis with option R&S RTP-K133: Jitter analysis waveforms are automatically displayed in their own diagrams.
Advanced noise analysis with option R&S RTP-K134 : Noise analysis waveforms are automatically displayed in their own diagrams.

<DiagramName> String with the name of the new diagram.

**Example:** `LAYout:ADD 'Diagram2',TAB,ON,C4W1,'MyDiagram3'`
Creates a new diagram 'MyDiagram3' with waveform C4W1 in a new tab that in front of 'Diagram2'.

**Example:** See Chapter 24.3.1.1, "Creating diagrams (SmartGrid)", on page 1173.

**Usage:** Setting only
Asynchronous command

---

**LAYout:REMove** <DiagramName>

Closes the specified diagram. The waveforms are displayed as minimized waveforms in their signal icons on the signal bar.

**Setting parameters:**
<DiagramName> String with the name of the diagram

**Usage:** Setting only
Asynchronous command

---

**LAYout:SHOW** <DiagramName>

Selects the specified diagram.

**Setting parameters:**
<DiagramName> String with the name of the diagram

**Example:** See Chapter 24.3.1.1, "Creating diagrams (SmartGrid)", on page 1173.

**Usage:** Setting only
Asynchronous command

---

**LAYout:SIGNal:ASSign** <DiagramName>, <Source>

Shows the specified waveform in the selected diagram.

**Setting parameters:**

<DiagramName> String with the diagram name

<Source> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | XY1 | XY2 | XY3 | XY4 | MRESult1 | MRESult2 | MRESult3 | MRESult4 | MRESult5 | MRESult6 | MRESult7 | MRESult8 | SBUS1 | SBUS2 | SBUS3 | SBUS4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | MSOB1 | MSOB2 | MSOB3 | MSOB4 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | BATHtub1 | BMEas1 | SRESponse1 | SRHist1 | SDHist1 | TJHist1 | DJHist1 | DDJHist1 | PJHist1 | RJOHist1 | SRTRack1 | SDTRack1 | TJTRack1 | DJTRack1 | DDJTrack1 | PJTRack1 | RJOTrack1 | TJSPectrum1 | DDJSpectrum1 | PJSPectrum1 | RJOSpectrum1 | BATHtub2 | BMEas2 | SRESponse2 | SRHist2 | SDHist2 | TJHist2 | DJHist2 | DDJHist2 | PJHist2 | RJOHist2 | SRTRack2 | SDTRack2 | TJTRack2 | DJTRack2 | DDJTrack2 | PJTRack2 | RJOTrack2 | TJSPectrum2 | DDJSpectrum2 | PJSPectrum2 | RJOSpectrum2 | NBAThtub1 | NBMeas1 | TNHist1 | DNHist1 | DDNHist1 | PNHist1 | RNOHist1 | TNTRack1 | DNTRack1 | DDNTrack1 | PNTRack1 | RNOTrack1 | TNSPectrum1 | DDNSpectrum1 | PNSPectrum1 | RNOSpectrum1 | NBAThtub2 | NBMeas2 | TNHist2 | DNHist2 | DDNHist2 | PNHist2 | RNOHist2 | TNTRack2 | DNTRack2 | DDNTrack2 | PNTRack2 | RNOTrack2 | TNSPectrum2 | DDNSpectrum2 | PNSPectrum2 | RNOSpectrum2 | PJDHist1 | BUJHist1 | RJHist1 | PJDTrack1 | PJDSpectrum1 | PJDHist2 | BUJHist2 | RJHist2 | PJDTrack2 | PJDSpectrum2 | PNDHist1 | BUNHist1 | RNHist1 | PNDTrack1 | PNDSpectrum1 | PNDHist2 | BUNHist2 | RNHist2 | PNDTrack2 | PNDSpectrum2

Waveform to be assigned, see Chapter 24.4.2, "Waveform parameter", on page 1193
Spectrum analysis, option R&S RTP-K37: Spectrograms and timeline spectrums are automatically displayed in their own diagrams.
Eye diagram analysis with option R&S RTP-K91: Eye diagrams are automatically displayed in their own diagrams.
TDR/TDT analysis with option R&S RTP-K130: Measurement waveforms are automatically displayed in their own diagrams.
Advanced jitter and noise analysis with option R&S RTP-K133/K134

**Example:** See Chapter 24.3.1.1, "Creating diagrams (SmartGrid)", on page 1173.

**Usage:** Setting only
Asynchronous command

**LAYout:SIGNal:UNASsign** <Source>

Minimizes the specified waveform, i.e. removes it from the diagram and shows it in the signal icon.

**Setting parameters:**

<Source> See `LAYout:SIGNal:ASSign` on page 1213 and Chapter 24.4.2, "Waveform parameter", on page 1193

**Example:** See Chapter 24.3.1.1, "Creating diagrams (SmartGrid)", on page 1173.

**Usage:** Setting only
Asynchronous command

### 24.7.2.5 Diagram appearance

DISPlay:DIAGram:TITLe ..... 1215
DISPlay:SIGBar[:STATe] ..... 1215
DISPlay:GATE:TRANsparency ..... 1215
LAYout:ZOOM:ONEDiagram ..... 1216

**DISPlay:DIAGram:TITLe** <DiagTitleSt>

If enabled, the tab titles of all diagrams are displayed: "Diagram1", "Diagram2" ...

If disabled, the tab titles are not shown except for those in a tabbed diagram. In tabbed diagrams, the tab titles are required to change the tabs.

**Parameters:**

<DiagTitleSt> ON | OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show tabs always" on page 83

**DISPlay:SIGBar[:STATe]** <State>

If enabled, the signal bar is displayed.

**Parameters:**

<State> ON | OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show signal bar" on page 83

**DISPlay:GATE:TRANsparency** <Transparency>

Sets the transparency of the area that is defined as measurement or search gate.

**Parameters:**

<Transparency> Range: 0 to 100
Increment: 1
*RST: 43
Default unit: %

**Usage:** Asynchronous command

**Firmware/software:** FW 3.20

**Manual operation:** See "Gate transparency" on page 84

---

**LAYout:ZOOM:ONEDiagram** <ShowInOne>

Shows all zooms of a diagram in one zoom window. The zoomed areas are overlaid for better comparison of the zoomed waveforms.

The command takes effect on all zoom diagrams.

**Parameters:**

<ShowInOne> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Zoom overlay" on page 84

### 24.7.2.6 Annotations

To create an annotation (or waveform label), use `DISPlay:SIGNal:LABel:ADD`. Using the other `DISP:SIGN:LAB:...` commands, you can query the text and position of a label, and modify the initial settings. The <LabelID> and <Source> parameters identify each label uniquely. You can query the <LabelID> with `DISPlay:SIGNal:LABel:IDSList`.

DISPlay:SIGNal:LABel:POSMode....1217
DISPlay:SIGNal:LABel:FONTsize....1217
DISPlay:SIGNal:LABel:ADD....1217
DISPlay:SIGNal:LABel:REMove....1219
DISPlay:SIGNal:LABel:IDSList....1219
DISPlay:SIGNal:LABel:SHOW....1219
DISPlay:SIGNal:LABel:STATe....1220
DISPlay:SIGNal:LABel:TEXT....1221
DISPlay:SIGNal:LABel:HORizontal:ABSolute:POSition....1221
DISPlay:SIGNal:LABel:VERTical:ABSolute:POSition....1221
DISPlay:SIGNal:LABel:HORizontal:RELative:POSition....1222
DISPlay:SIGNal:LABel:VERTical:RELative:POSition....1222

**DISPlay:SIGNal:LABel:POSMode** <Source>, <PositionMode>
**DISPlay:SIGNal:LABel:POSMode?** <Source>

Modifies or queries the position mode: either relative to the diagram or with absolute values according to the units of the waveform. The position mode applies to all labels of the selected source. For different sources, different position modes can be selected.

**Parameters:**

<PositionMode> ABS | REL

**ABS**
Position in time and voltage values, or in other units depending on the waveform character. Absolute positions move with the waveform display when the scales, the vertical position or offset, or the reference point are changed.
Use `DISPlay:SIGNal:LABel:HORizontal:ABSolute:POSition` and `DISPlay:SIGNal:LABel:HORizontal:RELative:POSition` to set the position.

**REL**
Fixed label position in percent of the screen counting from the upper left corner.
Use `DISPlay:SIGNal:LABel:HORizontal:RELative:POSition` and `DISPlay:SIGNal:LABel:VERTical:RELative:POSition` to set the position.

**Parameters for setting and query:**

<Source> All waveforms that can be displayed, see `DISPlay:SIGNal:LABel:ADD`

**Usage:** Asynchronous command

**Manual operation:** See "Position mode" on page 85

**DISPlay:SIGNal:LABel:FONTsize** <Fontsize>

Sets the font size of waveform labels.

**Parameters:**

<Fontsize>
Range: 7 to 30
Increment: 1
*RST: 15

**Usage:** Asynchronous command

**Manual operation:** See "Font size" on page 86

**DISPlay:SIGNal:LABel:ADD** <LabelID>, <Source>, <LabelText>, <PositionMode>, <XPositon>, <YPositon>

Creates a new waveform label for the specified source waveform.

**Setting parameters:**

<LabelID> String with the label identifier. The <LabelID> and <Source> parameters identify each label uniquely, so the label ID must be unique for the given waveform. Note the <LabelID> because it is not possible to query it, or to read it on the user interface.

<Source> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | XY1 | XY2 | XY3 | XY4 | MRESult1 | MRESult2 | MRESult3 | MRESult4 | MRESult5 | MRESult6 | MRESult7 | MRESult8 | QUICk | QUICK | SBUS1 | SBUS2 | SBUS3 | SBUS4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | MSOB1 | MSOB2 | MSOB3 | MSOB4 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | SG1 | SG2 | SG3 | SG4 | SG5 | SG6 | SG7 | SG8 | SG1TL1 | SG1TL2 | SG2TL1 | SG2TL2 | SG3TL1 | SG3TL2 | SG4TL1 | SG4TL2 | SG5TL1 | SG5TL2 | SG6TL1 | SG6TL2 | SG7TL1 | SG7TL2 | SG8TL1 | SG8TL2 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | EYE1 | EYE2 | EYE3 | EYE4 | TDRZ | TDRH | TDRR | SEYE1 | SEYE2

Waveform to that the label belongs, see Chapter 24.4.1, "Waveform suffix", on page 1188.

<LabelText> String with the label text that is shown on the display

<PositionMode> ABS | REL

**ABS**
Position in time and voltage values, or in other units depending on the waveform character. Absolute positions move with the waveform display when the scales, the vertical position or offset, or the reference point are changed.

**REL**
Fixed label position in percent of the screen counting from the upper left corner.

<XPositon> Horizontal position of the label text. Values, range and unit depend on the position mode, the waveform position and scaling. For relative position mode, the range is 0 to 100 %.

<YPositon> Vertical position of the label text. Values, range and unit depend on the position mode, the waveform position and scaling. For relative position mode, the range is 0 to 100 %.

**Example:**

```
DISPlay:SIGNal:LABel:ADD 'Label1', C1W1, 'Label on C1', REL, 20, 20
```

Adds the label text 'Label on C1W1' to waveform1 of channel1 at relative position 20% from the upper left corner of the screen. The label ID is 'Label1'.

**Example:** `DISPlay:SIGNal:LABel:ADD 'Label1', C2W1, 'Label on C21', ABS, 10e-09, 0.1`
Adds the label text 'Label on C2W1' to waveform1 of channel2 at absolute position 10 ns and 0.1 V. The label ID is 'Label1'.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Labels" on page 86

---

**DISPlay:SIGNal:LABel:REMove** <LabelID>, <Source>

Deletes the specifies waveform label.

**Setting parameters:**
<LabelID> String with the label identifier.

<Source> All waveforms that can be displayed, see `DISPlay:SIGNal:LABel:ADD`

**Example:** `DISPlay:SIGNal:LABel:REMove 'Label1', C1W1`

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Labels" on page 86

---

**DISPlay:SIGNal:LABel:IDSList** <Source>

Returns a list of label IDs for the specified source. Labels that are defined in the user interface get an automatic label "DiagramText<No>".

**Parameters:**
<Source> See `DISPlay:SIGNal:LABel:ADD`.

**Usage:** Asynchronous command

**Manual operation:** See "Labels" on page 86

---

**DISPlay:SIGNal:LABel:SHOW** <Source>, <State>
**DISPlay:SIGNal:LABel:SHOW?** <Source>

Selects the source for the label display and displays all labels for that source that have state = ON in `DISPlay:SIGNal:LABel:STATe` on page 1220.

**Parameters:**
<State> OFF | ON

**Parameters for setting and query:**

<Source> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | XY1 | XY2 | XY3 | XY4 | MRESult1 | MRESult2 | MRESult3 | MRESult4 | MRESult5 | MRESult6 | MRESult7 | MRESult8 | SBUS1 | SBUS2 | SBUS3 | SBUS4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | MSOB1 | MSOB2 | MSOB3 | MSOB4 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | BATHtub1 | BMEas1 | SRESponse1 | SRHist1 | SDHist1 | TJHist1 | DJHist1 | DDJHist1 | PJHist1 | RJOHist1 | SRTRack1 | SDTRack1 | TJTRack1 | DJTRack1 | DDJTrack1 | PJTRack1 | RJOTrack1 | TJSPectrum1 | DDJSpectrum1 | PJSPectrum1 | RJOSpectrum1 | SEYE1 | BATHtub2 | BMEas2 | SRESponse2 | SRHist2 | SDHist2 | TJHist2 | DJHist2 | DDJHist2 | PJHist2 | RJOHist2 | SRTRack2 | SDTRack2 | TJTRack2 | DJTRack2 | DDJTrack2 | PJTRack2 | RJOTrack2 | TJSPectrum2 | DDJSpectrum2 | PJSPectrum2 | RJOSpectrum2 | SEYE2 | NBAThtub1 | NBMeas1 | TNHist1 | DNHist1 | DDNHist1 | PNHist1 | RNOHist1 | TNTRack1 | DNTRack1 | DDNTrack1 | PNTRack1 | RNOTrack1 | TNSPectrum1 | DDNSpectrum1 | PNSPectrum1 | RNOSpectrum1 | NBAThtub2 | NBMeas2 | TNHist2 | DNHist2 | DDNHist2 | PNHist2 | RNOHist2 | TNTRack2 | DNTRack2 | DDNTrack2 | PNTRack2 | RNOTrack2 | TNSPectrum2 | DDNSpectrum2 | PNSPectrum2 | RNOSpectrum2 | PJDHist1 | BUJHist1 | RJOHist1 | PJDTrack1 | PJDSpectrum1 | PJDHist2 | BUJHist2 | RJOHist2 | PJDTrack2 | PJDSpectrum2 | PNDHist1 | BUNHist1 | RNOHist1 | PNDTrack1 | PNDSpectrum1 | PNDHist2 | BUNHist2 | RNOHist2 | PNDTrack2 | PNDSpectrum2

**Usage:** Asynchronous command

**Manual operation:** See "Show / Source" on page 85

---

**DISPlay:SIGNal:LABel:STATe** <LabelID>, <Source>, <State>
**DISPlay:SIGNal:LABel:STATe?** <LabelID>, <Source>

Displays or hides a single label that is identified by the signal source and label ID.

**Parameters:**

<State> OFF | ON

ON: shows the label on the screen.
`DISPlay:SIGNal:LABel:SHOW` must be set to ON for the source.

**Parameters for setting and query:**

<LabelID> String with the label identifier.

<Source> See DISPlay:SIGNal:LABel:ADD.

**Usage:** Asynchronous command

**Manual operation:** See "Labels" on page 86

---

**DISPlay:SIGNal:LABel:TEXT** <LabelID>, <Source>, <LabelText>
**DISPlay:SIGNal:LABel:TEXT?** <LabelID>, <Source>

Modifies or queries the text of the specified label.

**Parameters:**
<LabelText> String with the label text that is shown

**Parameters for setting and query:**
<LabelID> String with the label identifier.

<Source> All waveforms that can be displayed, see DISPlay:SIGNal:LABel:ADD

**Usage:** Asynchronous command

**Manual operation:** See "Labels" on page 86

---

**DISPlay:SIGNal:LABel:HORizontal:ABSolute:POSition** <LabelID>, <Source>, <Position>
**DISPlay:SIGNal:LABel:HORizontal:ABSolute:POSition?** <LabelID>, <Source>
**DISPlay:SIGNal:LABel:VERTical:ABSolute:POSition** <LabelID>, <Source>, <Position>
**DISPlay:SIGNal:LABel:VERTical:ABSolute:POSition?** <LabelID>, <Source>

Modifies or queries the absolute horizontal and vertical positiions of the specified label if DISPlay:SIGNal:LABel:POSMode is set to ABS.

**Parameters:**
<Position> Range: Depends on waveform position and scaling
Default unit: s and V, or in other units depending on the waveform character

**Parameters for setting and query:**
<LabelID> String with the label identifier.

<Source> All waveforms that can be displayed, see DISPlay:SIGNal:LABel:ADD

**Example:**
```
:DISPlay:SIGNal:LABel:POSMode C2W1, ABS
DISPlay:SIGNal:LABel:HORizontal:ABSolute:POSition 'Label1', C2W1, 5e-09
DISPlay:SIGNal:LABel:VERTical:ABSolute:POSition 'Label1', C2W1, -0.1
```

Move the label to 5 ns and -0.1 V.

**Manual operation:** See "Labels" on page 86

**DISPlay:SIGNal:LABel:HORizontal:RELative:POSition** <LabelID>, <Source>, <Position>
**DISPlay:SIGNal:LABel:HORizontal:RELative:POSition?** <LabelID>, <Source>
**DISPlay:SIGNal:LABel:VERTical:RELative:POSition** <LabelID>, <Source>, <Position>
**DISPlay:SIGNal:LABel:VERTical:RELative:POSition?** <LabelID>, <Source>

Modifies or queries the relative horizontal and vertical positiions of the specified label if `DISPlay:SIGNal:LABel:POSMode` is set to `REL`.

**Parameters:**

<Position> Position in percent of the screen counting from the upper left corner.

Range: 0 to 100
Default unit: %

**Parameters for setting and query:**

<LabelID> String with the label identifier.

<Source> All waveforms that can be displayed, see `DISPlay:SIGNal:LABel:ADD`

**Example:**

```
:DISPlay:SIGNal:LABel:POSMode C2W1, REL
DISPlay:SIGNal:LABel:HORizontal:RELative:POSition 'Label1', C2W1, 30
DISPlay:SIGNal:LABel:VERTical:RELative:POSition 'Label1', C2W1, 70
```

Move the label to new relative position: horizontal at 30 % and vertical at 70 % of the screen.

**Manual operation:** See "Labels" on page 86

### 24.7.2.7 Dialog appearance

**DISPlay:EXTended:PORDialogs** <PopOutResDialogs>

If an external monitor is connected to the instrument, enables the display of result boxes on the external monitor.

**Parameters:**

<PopOutResDialogs> ON | OFF

*RST: OFF

**Manual operation:** See "Extended desktop placement" on page 87

**DISPlay:EXTended:POSDialogs** <PopOutSettDialogs>

If an external monitor is connected to the instrument, enables the display of dialog boxes on the external monitor.

**Parameters:**

<PopOutSettDialogs> ON | OFF

*RST: OFF

**Manual operation:** See "Extended desktop placement" on page 87

### 24.7.2.8 Cursor appearance

**CURSor<m>:STYLe** <Style>

Defines how the cursor is displayed in the diagram.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<Style> LINes | LRHombus | VLRHombus | RHOMbus

**LINes**
The cursors are displayed as lines.

**LRHombus**
The cursors are displayed as lines. The intersections of the cursors with the waveforms are displayed by rhombus-shaped points.

**VLRHombus**
The cursors are displayed only as vertical lines. The intersections of the cursors with the waveforms are displayed by rhombus-shaped points.

**RHOMbus**
The intersections of the cursors with the waveforms are displayed by rhombus-shaped points.

*RST: LINes

**Usage:** Asynchronous command

**Manual operation:** See "Cursor style" on page 88

---

**DISPlay:RESultboxes:CUPosition** <Position>

Defines the position of the cursor measurement results on the screen.

**Parameters:**
<Position> PREV | FLOA | DOCK

**PREV**
Preview: result icon on the signal bar.

**FLOA**
Floating result box in front of the diagrams.

**DOCK**
Docked: fixed tab below the diagrams.

*RST: DOCK

**Usage:** Asynchronous command

**Manual operation:** See "Result position" on page 88

### 24.7.2.9 Measurement appearance

**MEASurement<m>:DISPlay:GROuping** <GroupResDialogs>

If ON, all measurement results are shown in one result box. If disabled, the results of each measurement group are shown in a separate result box. The default position is ignored.

**Suffix:**
<m> 1..10
The suffix is irrelevant. The setting affects all measurements except for the peak list.

**Parameters:**
<GroupResDialogs> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Group results" on page 89

**DISPlay:RESultboxes:MEPosition** <Position>

Defines the position of the measurement results on the display.

**Parameters:**
<Position> PREV | FLOA | DOCK
**PREV**
Preview: result icon on the signal bar
**FLOA**
Floating: floating result box in front of the diagrams
**DOCK**
Docked: fixed tab below the diagrams
*RST: DOCK

**Usage:** Asynchronous command

**Manual operation:** See "Result position" on page 89

### 24.7.2.10 Peak list appearance

**MEASurement<m>:RESult:LABorder** <MeasType>, <FrameType>
**MEASurement<m>:RESult:LABorder?** <MeasType>

Defines the layout of the labels.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FrameType> NOBorder | UNDERLINE | FULL

**NOBorder**
Label without border or lines.

**UNDERLINE**
The label is underlined.

**FULL**
The label is surrounded by a frame.

**Parameters for setting and query:**
<MeasType> PLISt

Only for peaklist measurements. For other measurements, this command returns an error.

**Manual operation:** See "Frame type" on page 90

---

**MEASurement<m>:RESult:INVerse** <MeasType>, <State>
**MEASurement<m>:RESult:INVerse?** <MeasType>

Displays labels with black font on white background using the "Full frame" label type (if `MEASurement<m>:RESult:LABorder=ON`). ).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<State> ON | OFF

**Parameters for setting and query:**
<MeasType> PLISt

Only for peaklist measurements. For other measurement, this command returns an error.

**Example:** Display labels.
```
MEAS:RES:SHL PLIS,ON
```
Select inverted labels.
```
MEAS:RES:INV PLIS,ON
```
Query the type of labels for peak lists.
```
MEAS:RES:INV? PLIS
//Result: ON
```

**Manual operation:** See "Invert" on page 90

### 24.7.2.11 Mask appearance

MTESt:LABel....1226
MTESt:REName....1226
MTESt:HIGHlight:INFinite....1226
MTESt:HIGHlight:STATe....1227
MTESt:HIGHlight:TIME....1227
MTESt:COLor:UNMatch....1227
MTESt:COLor:MATCh....1227
MTESt:COLor:BORDer....1228
MTESt:COLor:INTerior....1228

---

**MTESt:LABel** <LabelState>

Switches the display of the mask test name on or off.

To change the name of the mask test, use `MTESt:REName` on page 1226.

**Parameters:**
<LabelState> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Show labels" on page 91

---

**MTESt:REName** <MaskTestName>, <NewName>
**MTESt:REName?** <MaskTestName>

Changes the name of the mask test.

**Parameters:**
<NewName> String with the new mask test name.

**Parameters for setting and query:**
<MaskTestName> String with the existing mask test name.

**Manual operation:** See "Show labels" on page 91

---

**MTESt:HIGHlight:INFinite** <State>

If ON, the mask hits are highlighted for an unlimited period of time.

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show hits forever" on page 91

**MTESt:HIGHlight:STATe** <HighlightHits>

If ON, the mask hits are highlighted on the screen. You can define the color and the time of the hit display.

**Parameters:**

<HighlightHits> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Show hits" on page 91

**MTESt:HIGHlight:TIME** <HighlightTime>

Sets the time how long the mask hits are highlighted.

**Parameters:**

<HighlightTime> Range: 0.05 to 50
Increment: 0.1
*RST: 1
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Show for" on page 91

**MTESt:COLor:UNMatch** <ClrUnmatchedMsk>

Sets the color of masks segments that were not hit.

**Parameters:**

<ClrUnmatchedMsk> Decimal value corresponding to the ARGB hex value that is shown in the "Color" dialog box. ARGB is a 32-bit unsigned integer with alpha, red, green, blue word order. Alpha defines the transparency.
To get the value, covert the hex value from the dialog to decimal value.

Range: 0 (black) to 4294967295 (white)
Increment: 1
*RST: 2578098858

**Usage:** Asynchronous command

**Manual operation:** See "Mask without violation" on page 92

**MTESt:COLor:MATCh** <ClrMatchedPixel>

Sets the color of samples that violated the mask.

**Parameters:**

<ClrMatchedPixel> Decimal value corresponding to the ARGB hex value that is shown in the "Color" dialog box. ARGB is a 32-bit unsigned integer with alpha, red, green, blue word order. Alpha defines the transparency.
To get the value, covert the hex value from the dialog to decimal value.

Range: 0 (black) to 4294967295 (white)
Increment: 1
*RST: 4294901760

**Usage:** Asynchronous command

**Manual operation:** See "Color" on page 92

---

**MTESt:COLor:BORDer** <ClrMatchedMskBorder>

Sets the color of masks segments that were touched at the border. In this case, the resolution is not sufficient to detect if the mask was really hit or not. Zoom into the concerned area to see the actual result.

**Parameters:**

<ClrMatchedMskBorder> Decimal value corresponding to the ARGB hex value that is shown in the "Color" dialog box. ARGB is a 32-bit unsigned integer with alpha, red, green, blue word order. Alpha defines the transparency.
To get the value, covert the hex value from the dialog to decimal value.

Range: 0 (black) to 4294967295 (white)
Increment: 1
*RST: 2580309606

**Usage:** Asynchronous command

**Manual operation:** See "Mask with contact" on page 92

---

**MTESt:COLor:INTerior** <ClrMatchedMskInterior>

Sets the color of mask segments the signal has entered into.

**Parameters:**

<ClrMatchedMskInterior> Decimal value corresponding to the ARGB hex value that is shown in the "Color" dialog box. ARGB is a 32-bit unsigned integer with alpha, red, green, blue word order. Alpha defines the transparency.
To get the value, covert the hex value from the dialog to decimal value.

Range: 0 (black) to 4294967295 (white)
Increment: 1
*RST: 2578055168

**Usage:** Asynchronous command

**Manual operation:** See "Mask with violation" on page 92

## 24.7.3 Display

### 24.7.3.1 Persistence

DISPlay:PERSistence[:STATe] ..... 1229
DISPlay:PERSistence:INFinite ..... 1229
DISPlay:PERSistence:TIME ..... 1229
DISPlay:PERSistence:RESet ..... 1230

**DISPlay:PERSistence[:STATe]** <State>

If enabled, each new data point in the diagram area remains on the screen for the duration defined using `DISPlay:PERSistence:TIME`, or as long as `DISPlay:PERSistence:INFinite` is enabled.

If disabled, the signal value is only displayed as long as it actually occurs.

**Parameters:**

<State> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 94

**DISPlay:PERSistence:INFinite** <State>

If persistence is enabled (`DISPlay:PERSistence[:STATe]`), each new data point in the diagram area remains on the screen infinitely until this command is set to "OFF".

**Parameters:**

<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Infinite persistence" on page 94

**DISPlay:PERSistence:TIME** <Time>

If persistence is enabled (`DISPlay:PERSistence[:STATe]`), each new data point in the diagram area remains on the screen for the duration defined here.

**Parameters:**

<Time> Range: 0.05 to 50
Increment: 0.05
*RST: 0.05
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Time" on page 94

---

**DISPlay:PERSistence:RESet**

Resets the display, removing persistent values.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Reset" on page 94

### 24.7.3.2 Signal

---

**DISPlay:INTensity** <Intensity>

This value determines the strength of the waveform line in the diagram. Enter a percentage between 0 (not visible) and 100% (very strong).

The exact mapping of the cumulative value occurences according to the assigned color table is guaranteed only if the intensity is set to 50% (default). All other intensity values falsify the mapping but may improve the visibility of the signal.

**Parameters:**

<Intensity>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Intensity" on page 95

---

**DISPlay:DIAGram:STYLe** <Style>

Select the style in which the waveform is displayed.

**Parameters:**

<Style>
VECTors | DOTS

**VECTors**
The individual data points are connected by a line.

**DOTS**
Only the individual data points are displayed.

*RST: VECTors

**Usage:** Asynchronous command

**Manual operation:** See "Waveform style" on page 95

#### 24.7.3.3 Clear results

**DISPlay:CLR**

Deletes all measurement results including long term measurement and statistics, all waveforms, and the history.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Clear" on page 58

### 24.7.4 User-defined preset

**USRDefined:RST:NAME** <Name>

Sets the path and file name of a user-defined preset file.

**Parameters:**
<Name> String parameter

**Manual operation:** See "Saveset to be loaded on preset" on page 101

**USRDefined:RST:OPEN**

opens and loads a user-defined preset file. Name and path of the preset file are defined with `USRDefined:RST:NAME`.

**Usage:** Event

**Manual operation:** See "Saveset to be loaded on preset" on page 101

**USRDefined:RST:SAVE**

Saves the current instrument settings to a user-defined preset file. Name and path of the preset file are defined with `USRDefined:RST:NAME`.

**Usage:** Event

**Manual operation:** See "Saveset to be loaded on preset" on page 101

**USRDefined:RST[:ENABle]** <Name>

If ON, the settings from a previously defined preset file are restored when the [Preset] key is pressed. The preset file is defined with `USRDefined:RST:NAME`.

If OFF, [Preset] sets the instrument to the factory defaults.

**Parameters:**
<Name> ON | OFF
*RST: OFF

**Manual operation:** See "Enable user defined preset" on page 102

## 24.7.5 Save / recall

See the following chapters:

- Chapter 24.16.10, "Autonaming", on page 1633
- Chapter 24.16.11, "CSV export", on page 1634
- Chapter 24.16.12, "External application", on page 1635

## 24.7.6 Maintenance

### 24.7.6.1 Self-alignment

CALibration:DATE? 1232
CALibration:TIME? 1232
CALibration:RESult? 1232

**CALibration:DATE?**

Returns the date of the last selfalignment.

**Return values:**
<Date>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Date / Time / Overall alignment state" on page 104

**CALibration:TIME?**

Returns the time of the last selfalignment.

**Return values:**
<Time>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Date / Time / Overall alignment state" on page 104

**CALibration:RESult?**

Returns the result of the last selfalignment and the current alignment status. In remote mode, `*CAL?` provides more detailed information.

**Return values:**
<ResultState> PASSed | FAILed | NOALigndata
*RST: FAILed

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Date / Time / Overall alignment state" on page 104

### 24.7.6.2 Self-test

**DIAGnostic:SERVice:STST:STATe?**

Returns the summary result of the selftest.

**Return values:**
<State> PSSD | FAILed | UNDefined
*RST: UNDefined

**Usage:** Query only
Asynchronous command

**Manual operation:** See "State" on page 106

### 24.7.6.3 Hardware

**SYSTem:DFPRint** [<Path>]

The device footprint contains the configuration of the instrument, installed modules, installed software and software licenses. This information is written in the device footprint xml file might be useful in case of maintenance or support request.

The query returns the information as block data. The setting command saves the device footprint xml file in the specified path.

It is also possible to access the device footprint xml file via the instrument's web browser. Therefore, the directory containing the xml file must be enabled for sharing.

**Setting parameters:**
<Path> String parameter, specifying the target path of the footprint file.

**Return values:**
<DeviceFootprint> Content of the device footprint xml file as block data

**Manual operation:** See "System Info" on page 106

**DIAGnostic:SERVice:FWVersion?**

Returns the firmware version that is currently installed on the instrument.

**Return values:**
<FirmwareVersion> Version string

**Usage:** Query only
Asynchronous command

**DIAGnostic:SERVice:PARTnumber** <Number>

Returns the part number (order number) of your instrument. This number is required to order a new option, and in case of service.

**Parameters:**
<Number> Number string

**Usage:** Asynchronous command

**Manual operation:** See "System Info" on page 106

**DIAGnostic:SERVice:SERialnumber?**

Returns the serial number of your instrument. This number is required to order a new option, and in case of service.

**Return values:**
<SerialNumber> Number string

**Usage:** Query only
Asynchronous command

**Manual operation:** See "System Info" on page 106

**DIAGnostic:SERVice:CHANnelcount?**

Queries the number of available channels.

**Return values:**
<ChannelCount> Range: 0 to 4
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**Firmware/software:** V 2.00

## 24.8 Acquisition and setup

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"

- Chapter 24.3, "Programming examples", on page 1172

- Starting and stopping acquisition....1235
- Time base....1236
- Acquisition....1242
- Fast segmentation....1247
- Vertical....1248
- Waveform data....1252
- Probes....1254
- R&S RT-ZVC probe....1280
- Differential signals....1294
- Digital filter....1298
- Skew....1299
- AUX OUT....1301
- High definition mode....1302
- Reference clock....1303

## 24.8.1 Starting and stopping acquisition

RUNContinous....1235
RUN....1235
RUNSingle....1235
SINGle....1235
STOP....1235

**RUNContinous**
**RUN**

Starts the continuous acquisition.

**Usage:** Event
Asynchronous command

**Manual operation:** See "[Run / Stop]" on page 37

**RUNSingle**
**SINGle**

Starts a defined number of acquisition cycles. The number of cycles is set with `ACQuire:COUNt`.

**Usage:** Event
Asynchronous command

**Manual operation:** See "[Single]" on page 37

**STOP**

Stops the running acquistion.

**Usage:** Event
Asynchronous command

**Manual operation:** See "[Run / Stop]" on page 37

## 24.8.2 Time base

TIMebase:SCALe....1236
TIMebase:RANGe....1237
TIMebase:DIVisions?....1237
TIMebase:HORizontal:POSition....1237
TIMebase:REFerence....1237
TRIGger<m>:OFFSet:LIMited....1238
AUToscale....1238
ACQuire:POINts:AUTO....1238
ACQuire:POINts:AADJust....1239
ACQuire:POINts:MAXimum....1239
ACQuire:POINts:ARATe?....1239
ACQuire:SRATe....1240
ACQuire:SRReal....1240
ACQuire:RESolution....1240
ACQuire:POINts[:VALue]....1241
TIMebase:ROLL:ENABle....1241
TIMebase:ROLL:STATe?....1241
TIMebase:ROLL:MTIMe....1242

**TIMebase:SCALe** <TimeScale>

Sets the horizontal scale - the time per division on the x-axis - for all channel and math waveforms.

The setting accuracy depends on the current resolution (sample rate).

- No interpolation:
  The resolution is an integer multiple of the ADC sample rate.
- With interpolation:
  Any value for the horizontal scale can be set.

**Parameters:**

<TimeScale>
Range: 25E-12 to 10000 (RTO, RTP) | 5000 (RTE)
Increment: 1E-12
*RST: 10E-9
Default unit: s/div

**Usage:** Asynchronous command

**Manual operation:** See "Time scale" on page 123

**TIMebase:RANGe** <AcquisitionTime>

Defines the time of one acquisition, that is the time across the 10 divisions of the diagram: *TimeScale*10*.

**Parameters:**

<AcquisitionTime>
Range: 250E-12 to 100E+3 (RTO, RTP) | 50E+3 (RTE)
Increment: 1E-12
*RST: 0.5
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Acquisition time" on page 123

**TIMebase:DIVisions?**

Queries the number of horizontal divisions on the screen. The number cannot be changed.

**Return values:**

<HorizDivCnt>
Range: 4 to 20
Increment: 2
*RST: 10

**Usage:** Query only
Asynchronous command

**TIMebase:HORizontal:POSition** <RescalCenterTime>

Defines the time distance between the reference point and the trigger point (the zero point of the diagram). The reference point marks the rescaling center of the time scale.

**Parameters:**

<RescalCenterTime>
Range: -100E+24 to 100E+24
Increment: 1E-12
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "[Position]" on page 38

**TIMebase:REFerence** <RescaleCtrPos>

Sets the position of the reference point in % of the screen. The reference point marks the rescaling center of the time scale. If you modify the time scale, the reference point remains fixed on the screen, and the scale is stretched or compresses to both sides of the reference point.

**Parameters:**

<RescaleCtrPos>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "[Position]" on page 38

---

**TRIGger<m>:OFFSet:LIMited** <State>

If ON, the horizontal position cannot be set outside the visible waveform diagram.

See also: `TIMebase:HORizontal:POSition` on page 1237

**Suffix:**

<m> 1..3
The numeric suffix is irrelevant.

**Parameters:**

<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Restrict horizontal position to acquisition range" on page 123

---

**AUToscale**

Performs an autoset process: analyzes the enabled channel signals, and obtains appropriate horizontal, vertical, and trigger settings to display stable waveforms.

Rohde & Schwarz does not recommend using the autoset in remote control. To adjust the oscilloscope remotely, especially for automated testing applications, use the remote commands that adjust the horizontal, vertical and trigger settings."

**Usage:** Event
Asynchronous command

---

**ACQuire:POINts:AUTO** <ReclgthManual>

Selection to keep constant either the resolution or the record length when you adjust the time scale (`TIMebase:SCALe`) or acquisition time (`TIMebase:RANGe`).

**Parameters:**

<ReclgthManual> RESolution | RECLength

**RESolution**
Resolution is kept constant. Set the required resolution value with `ACQuire:RESolution`.

**RECLength**
The record length is kept constant. Set the required record length value with `ACQuire:POINts[:VALue]`.

*RST: RESolution

**Usage:** Asynchronous command

**Manual operation:** See "Sample rate mode" on page 131

---

**ACQuire:POINts:AADJust** <AutoAdjust>

Prevents undersampling and ensures a sufficient resolution to acquire the correct waveform if the time scale is changed. The setting takes effect if the changed parameter - resolution or record length - reaches a limit. The instrument automatically keeps this parameter constant at its limit, and changes the other parameter regardless of the `ACQuire:POINts:AUTO` setting.

**Parameters:**
<AutoAdjust> ON | OFF

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Auto adjustment" on page 131

---

**ACQuire:POINts:MAXimum** <ReclgthLim>

Sets a limit for the record length to prevent very large records. This value only takes effect if a constant resolution is selected with `ACQuire:POINts:AUTO`. If you increase the time scale, the resolution remains constant and the record length increases until the limit is reached. Further increase of the time scale changes the resolution and keeps the record length limit.

**Parameters:**
<ReclgthLim> Range: 1000 to 800 MSa. The actual maximum can be lower depending on the installed options, number of active channels, measurements and math waveforms.
Increment: 2
*RST: 10E+6
Default unit: pts

**Usage:** Asynchronous command

**Manual operation:** See "Record length limit" on page 131

---

**ACQuire:POINts:ARATe?**

Retrieves the sample rate of the ADC, that is the number of points that are sampled by the ADC in one second.

**Return values:**

<ADCSampleRate> Range: 20 Gsample/s, R&S RTP with serial number ≥ 103000 in interleaved mode (2 channels): 40 Gsample/s
Default unit: Sa/s

**Usage:** Query only
Asynchronous command

---

**ACQuire:SRATe** <SampleRate>

Defines the sample rate, that is the number of recorded waveform samples per second.

**Parameters:**

<SampleRate> Range: 20 Gsample/s, R&S RTP with serial number ≥ 103000 in interleaved mode (2 channels): 40 Gsample/s
Increment: 1
*RST: 20 Gsample/s
Default unit: Sa/s

**Usage:** Asynchronous command

---

**ACQuire:SRReal** <RealSampleRate>

Sets the number of captured waveform points per second. It considers the samples of the ADC, and the reduction of waveform points by decimation.

If interpolation is not active, the sample rate is the reciprocal value of the resolution and thus also depends on the acquisition time and the record length.

If interpolation is active, the sample rate is limited to the ADC sample rate.

**Parameters:**

<RealSampleRate> Range: 20 Gsample/s, R&S RTP with serial number ≥ 103000 in interleaved mode (2 channels): 40 Gsample/s
Increment: 1
*RST: 20 Gsample/s
Default unit: Sa/s

**Usage:** Asynchronous command

**Firmware/software:** FW 3.40

**Manual operation:** See "Sample rate" on page 129

---

**ACQuire:RESolution** <Resolution>

Indicates the time between two waveform points in the record.

**Parameters:**

<Resolution> A fine resolution with low values produces a more precise waveform record.

Range: 1E-15 to 0.5
Increment: 10E-12
*RST: 100E-12
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Resolution" on page 130

---

**ACQuire:POINts[:VALue]** <RecordLength>

Indicates the record length, the number of recorded waveform points that build the waveform across the acquisition time. `[:VALue]` can be omitted.

**Parameters:**

<RecordLength> Number of recorded waveform points.

Range: 1000 to 1000000000
Increment: 2
*RST: 1000
Default unit: pts

**Usage:** Asynchronous command

**Manual operation:** See "Record length" on page 129

---

**TIMebase:ROLL:ENABle** <Mode>

Activates the automatic roll mode.

**Parameters:**

<Mode> AUTO | OFF

AUTO: the instrument activates the roll mode under specific conditions.
See also "Roll mode" on page 124.

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Roll mode" on page 124

---

**TIMebase:ROLL:STATe?**

Returns the status of the roll mode.

**Return values:**

<State> ON | OFF

*RST: OFF

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Roll mode" on page 124

---

**TIMebase:ROLL:MTIMe** <MinHorizGain>

The roll mode is enabled automatically if the acquisition time exceeds the given value, and if `TIMebase:ROLL:ENABle` is set to AUTO.

**Parameters:**
<MinHorizGain> Treshold value for roll mode enabling.

Range: 1 to 600
Increment: 1
*RST: 10
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Start roll time" on page 125

### 24.8.3 Acquisition

ACQuire:INTerpolate ... 1242
ACQuire:CDTA ... 1243
ACQuire:MUWaveform ... 1243
CHANnel<m>[:WAVeform<n>][:STATe] ... 1243
CHANnel<m>[:WAVeform<n>]:TYPE ... 1244
CHANnel<m>[:WAVeform<n>]:ARIThmetics ... 1244
ACQuire:COUNt ... 1245
ACQuire:ARESet:IMMediate ... 1245
ACQuire:ARESet:MODE ... 1246
ACQuire:ARESet:TIME ... 1246
ACQuire:ARESet:COUNt ... 1246

---

**ACQuire:INTerpolate** <IntpolMd>

Selects the interpolation method.

See also: "Interpolation" on page 130.

**Parameters:**
<IntpolMd> LINear | SINX | SMHD

**LINear**
Linear interpolation between two adjacent sample points

**SINX**
Interpolation by means of a sin(x)/x curve.

**SMHD**
Sample/Hold causes a histogram-like interpolation.

*RST: SINX

**Usage:** Asynchronous command

**Manual operation:** See "Interpolation" on page 130

---

**ACQuire:CDTA** <CoupleAcquSet>

Sets the acquisition mode and the waveform arithmetic of all channels to the last set value.

If the acquisition settings are coupled, `ACQuire:MUWaveform` is not relevant, only one waveform per channel can be used.

**Parameters:**

<CoupleAcquSet> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Firmware/software:** FW 3.30
Substitutes `ACQuire:CMODode`

**Manual operation:** See "Couple channels" on page 133

---

**ACQuire:MUWaveform** <MultiWaveform>

For each channel, up to three waveforms can be shown and analyzed. The decimation mode and the waveform arithmetic are specific for each waveform. So you can analyze several aspects of the signal: For example, waveform1 shows the peaks, and waveform2 shows the average of the signal.

**Parameters:**

<MultiWaveform> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Firmware/software:** FW 3.20

**Manual operation:** See "Multi waveform" on page 133

---

**CHANnel<m>[:WAVeform<n>][:STATe]** <State>

Activates or deactivates a waveform. `[:STATe]` can be omitted.

Up to 3 waveforms per channel can be analyzed.

**Suffix:**

<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Parameters:**

<State> ON | OFF

*RST: OFF

**Example:** `CHAN2:WAV2 ON`
Activates waveform 2 of channel 2 (C2W2).

**Usage:** Asynchronous command

---

**CHANnel<m>[:WAVeform<n>]:TYPE** <DecimationMode>

Selects the method to reduce the data stream of the ADC to a stream of waveform points with lower sample rate.

See also:"Decimation" on page 132.

**Suffix:**

<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Parameters:**

<DecimationMode> SAMPle | PDETect | HRESolution | RMS

**SAMPle**
One of n samples in a sample interval of the ADC is recorded as waveform point.

**PDETect**
Peak Detect: the minimum and the maximum of n samples in a sample interval are recorded as waveform points.

**HRESolution**
High resolution: The average of n sample points is recorded as waveform point.

**RMS**
The waveform point is the root mean square of n sample values.

*RST: SAMPle

**Usage:** Asynchronous command

**Manual operation:** See "Decimation" on page 132

---

**CHANnel<m>[:WAVeform<n>]:ARIThmetics** <Arithmetics>

Selects the method to build the resulting waveform from several consecutive acquisitions of the signal. To define the number of acquisitions, use `ACQuire:COUNt`.

See also: "Arithmetic" on page 133.

**Suffix:**

<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Parameters:**

<Arithmetics> OFF | ENVelope | AVERage

**OFF**
The data of the current acquisition is recorded according to the decimation settings.

**ENVelope**
Detects the minimum and maximum values in an sample interval over a number of acquisitions. To define the reset method, use ...

**AVERage**
Calculates the average from the data of the current acquisition and a number of acquisitions before.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Arithmetic" on page 133

---

**ACQuire:COUNt** <MaxAcqCnt>

The acquisition and average count has a double effect:

- It sets the number of waveforms acquired with `RUNSingle`.
- It defines the number of waveforms used to calculate the average waveform.

**Parameters:**

<MaxAcqCnt> Range: 1 to 16777215
Increment: 10
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Average count (N-single count)" on page 130

---

**ACQuire:ARESet:IMMediate**

Forces the immediate restart of the envelope and average calculation for all waveforms.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Reset" on page 134

**ACQuire:ARESet:MODE** <ArtmRst>

Defines when the envelope and average evaluation restarts.

**Parameters:**

<ArtmRst> NONE | TIME | WFMS

**TIME**
Restarts the envelope and average calculation after the time defined with `ACQuire:ARESet:TIME`.

**WFMS**
Restarts the envelope and average calculation after a number of acquired waveforms defined with `ACQuire:ARESet:COUNt` on page 1246.

*RST: NONE

**Usage:** Asynchronous command

**Manual operation:** See "Arithmetic Reset > Mode" on page 134

**ACQuire:ARESet:TIME** <EnvelopeTimeout>

Defines the time after which the envelope and average evaluation restarts.

The setting is relevant if `ACQuire:ARESet:MODE` is set to `TIME`.

**Parameters:**

<EnvelopeTimeout>
Range: 0.1 to 10000
Increment: 0.01
*RST: 0.1
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Arithmetic Reset > Mode" on page 134

**ACQuire:ARESet:COUNt** <NofWaveforms>

Defines the number of acquired waveforms after which the envelope and average evaluation restarts.

The setting is relevant if `ACQuire:ARESet:MODE` is set to `WFMS`.

**Parameters:**

<NofWaveforms>
Range: 2 to 16777215
Increment: 10
*RST: 1000

**Usage:** Asynchronous command

**Manual operation:** See "Arithmetic Reset > Mode" on page 134

### 24.8.4 Fast segmentation

ACQuire:SEGMented:STATe ..... 1247
ACQuire:SEGMented:MAX ..... 1247
ACQuire:SEGMented:AUToreplay ..... 1247

**ACQuire:SEGMented:STATe** <State>

Switches the fast segmentation mode on and off.

**Parameters:**

<State> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Fast segmentation" on page 135

**ACQuire:SEGMented:MAX** <MaxAcquisitions>

The number of acquisitions in a fast segmentation acquisition series depends on the record length.

**Parameters:**

<MaxAcquisitions> ON | OFF

**ON**
Acquires the maximum possible number of acquisitions in a series.

**OFF**
Acquires the number of acquisitions defined using `ACQuire:COUNt`.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Number of acquisitions" on page 135

**ACQuire:SEGMented:AUToreplay** <ReplayAfterAcq>

If enabled, the instrument starts processing and displaying the data as soon as the acquisition series is captured completely. Depending on the number of acquisitions, it may take some time until the acquisition series is displayed. If the setting is disabled, the instrument only captures the data and stores it in the sample memory.

**Parameters:**

<ReplayAfterAcq> ON | OFF

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable automatic replay" on page 135

### 24.8.5 Vertical

CHANnel<m>:STATe ..... 1248
CHANnel<m>:COUPling ..... 1248
CHANnel<m>:SCALe ..... 1249
CHANnel<m>:RANGe ..... 1249
CHANnel<m>:POSition ..... 1250
CHANnel<m>:OFFSet ..... 1250
CHANnel<m>:INVert ..... 1250
CHANnel<m>:BANDwidth ..... 1251
CHANnel<m>:CPLing ..... 1251
CHANnel<m>:IMPedance ..... 1252
CHANnel<m>:OVERload ..... 1252

**CHANnel<m>:STATe** <State>

Switches the channel signal on or off.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "[C<n>]" on page 39

**CHANnel<m>:COUPling** <Value>

Selects the connection of the indicated channel signal.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<Value> DC | DCLimit | AC

**DC**
Direct connection with 50 Ω termination, passes both DC and AC components.

**DCLimit**
Direct connection with 1 MΩ termination, passes both DC and AC components.
Only available if the 1 MΩ adapter is used.

**AC**
Connection with 1 MΩ termination, removes DC and very low-frequency components.
Only available if the 1 MΩ adapter is used.

*RST: DC

**Manual operation:** See "Coupling" on page 142

---

**CHANnel<m>:SCALe** <Scale>

Sets the vertical scale for the indicated channel.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<Scale> Scale value, given in Volts per division.

Range: Depends on attenuation factors, coupling, and instrument model, see below.
Increment: Depends on vertical and probe settings
*RST: 0.05
Default unit: V/div
With 1:1 probe and external attenuation = 1, the vertical scale (input sensitivity) is minimum 2 mV/div (1 mV/div in high definition mode). If the probe and/or external attenuation is changed, multiply the values by the attenuation factors to get the actual scale range.

**Usage:** Asynchronous command

**Manual operation:** See "[Scale]" on page 40

---

**CHANnel<m>:RANGe** <Range>

Sets the voltage range across the 10 vertical divisions of the diagram. Use the command alternativly instead of `CHANnel<m>:SCALe`.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<Range> Voltage range value

Range: Dependencies see below.
Increment: Depends on vertical and probe settings
*RST: 0.5
Default unit: V/div
Depends on attenuation factors. With 1:1 probe and external attenuation = 1, the range is 20 mV to 10 V. If the probe and/or external attenuation is changed, multiply the range values by the attenuation factors.

**Usage:** Asynchronous command

**CHANnel<m>:POSition** <Position>

Sets the vertical position of the indicated channel as a graphical value.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<Position> Positive values move the waveform up, negative values move it down.

Range: -5 to 5
Increment: 0.01
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "[Position]" on page 40

**CHANnel<m>:OFFSet** <Offset>

The offset voltage is subtracted to correct an offset-affected signal. The offset of a signal is determined and set by the autoset procedure.

See also: "Offset" on page 141.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<Offset> Negative values move the waveform up, positive values move it down.

Range: Depends on attenuation factors, input coupling, and the offset compensation range of active probes. The nominal offset range for 1:1 attenuation and probe offset compensation = 0 is specified in the data sheet.
Increment: Depends on vertical and probe settings
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "[Position]" on page 40

**CHANnel<m>:INVert** <InvertChannel>

Turns the inversion of the signal amplitude on or off. To invert means to reflect the voltage values of all signal components against the ground level. If the inverted channel is the trigger source, the instrument triggers on the inverted signal.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<InvertChannel> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Firmware/software:** FW 3.30

**Manual operation:** See "Invert channel" on page 142

---

**CHANnel<m>:BANDwidth** <BandwidthLimit>

Selects the bandwidth limit for the indicated channel.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<BandwidthLimit> FULL
**FULL**
Use full bandwidth.
*RST: FULL

**Usage:** Asynchronous command

**Manual operation:** See "Analog" on page 143

---

**CHANnel<m>:CPLing** <Channel>, <State>
**CHANnel<m>:CPLing?** <Channel>

Sets the vertical settings of the coupled channel to the values of the active channel.

**Suffix:**
<m> 1..4
Active channel

**Parameters:**
<State> OFF | ON

**Parameters for setting and query:**
<Channel> CHAN1 | C1 | CHAN2 | C2 | CHAN3 | C2 | CHAN4 | C4
**CHAN1 = C1, CHAN2 = C2, CHAN3 = C2, CHAN4 = C4**
Input channel that takes the vertical settings of CHANnel<m>.

**Usage:** Asynchronous command

**Manual operation:** See "Coupled to C1 / C2 / C3 / C4" on page 144

**CHANnel<m>:IMPedance** <Impedance>

Sets the impedance of the channel for power calculations and measurements.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<Impedance> Range: 0.1 to 100E+3
Increment: 1
*RST: 50
Default unit: Ω

**Usage:** Asynchronous command

**Manual operation:** See "Impedance" on page 144

**CHANnel<m>:OVERload** <Overload>

Retrieves the overload status of the specified channel from the status bit. When the overload problem is solved, the command resets the status bit.

The overload status is returned asynchronously. Therefore, it is not possible to assign an overload to a specific acquisition during continuous acquisition. This assignment is only possible for a single acquisition.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<Overload> ON | OFF
Use OFF to reset the overload status bit.
*RST: OFF

**Example:**
```
ACQuire:COUNt 1
SINGle
CHANnel2:OVERload?
```
Queries the overload status of channel 2 after a single acquisition.
```
CHANnel2:OVERload OFF
```
Resets the overload status bit.

**Usage:** Asynchronous command

### 24.8.6 Waveform data

To set the export data format, see `FORMat[:DATA]` on page 1201.

CHANnel<m>[:WAVeform<n>]:DATA:HEADer? ..... 1253
CHANnel<m>[:WAVeform<n>]:DATA[:VALues]? ..... 1253

**CHANnel<m>[:WAVeform<n>]:DATA:HEADer?**

Returns the header of channel waveform data.

***Table 24-1: Header data***

| Position | Meaning | Example |
|---|---|---|
| 1 | XStart in s | -9.477E-008 = - 94,77 ns |
| 2 | XStop in s | 9.477E-008 = 94,77 ns |
| 3 | Record length of one waveform | 200000 |
| 4 | Number of values per sample interval. For most waveforms the result is 1, for peak detect and envelope waveforms it is 2. If the number is 2, the number of returned values is twice the number of samples (record length). | 1 |

If multichannel export is enabled, the number of returned samples is *Record length * Number of exported waveforms*. See also `EXPort:WAVeform:MULTichannel`.

**Suffix:**

<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Example:**
```
CHAN1:WAV1:DATA:HEAD?
-9.477E-008,9.477E-008,200000,1
```

**Usage:** Query only

**CHANnel<m>[:WAVeform<n>]:DATA[:VALues]?** [<Offset>], [<Length>]

Returns the data of the channel waveform points for transmission from the instrument to the controlling computer. The data can be used in MATLAB, for example.

Without parameters, the complete waveform is retrieved. Using the offset and length parameters, data can be retrieved in smaller portions, which makes the command faster.

To set the export format, use `FORMat[:DATA]`.

You can retrieve only Y-values (usually voltage values), or X- and Y-values. Use `EXPort:WAVeform:INCXvalues` to define which values are retrieved.

If multichannel export is active (`EXPort:WAVeform:MULTichannel`), the channel suffix is ignored. To select the channels to be exported, use `CHANnel<m>:EXPortstate`. The Y-values are written in interleaved order, for example, $YCh1_0$; $YCh2_0$; $YCh1_1$; $YCh2_1$... for a 2-channel instrument.

**Suffix:**

<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Query parameters:**

[<Offset>] Number of offset waveform points.
Range: 0 to m. Limit: n + m <= record length

[<Length>] Number of waveform points to be retrieved.
Range: 1 to n. Limit: n + m <= record length

**Return values:**

<Data> List of values according to the format and content settings.

**Example:** Retrieve the complete channel 1 waveform, only Y-values:

```
FORM ASC
EXP:WAV:INCX OFF
CHAN1:WAV1:DATA?
<-- -0.125000,-0.123016,-0.123016,-0.123016,-0.123016,-0.123016,...
```

**Example:** Retrieve the first 10 values of the waveform:

```
CHANnel:WAVeform:DATA:VALues? 0,10
<-- -0.10079051554203,-0.098814234137535,-0.098814234137535,
    -0.096837945282459,-0.094861663877964,-0.094861663877964,
    -0.092885382473469,-0.090909093618393,-0.090909093618393,
    -0.088932812213898
```

**Example:** Skip 5 samples and retrieve the next 5 samples:

```
CHANnel:WAVeform:DATA:VALues? 5,5
<-- -0.094861663877964,-0.092885382473469,-0.090909093618393,
    -0.090909093618393,-0.088932812213898
```

**Usage:** Query only

## 24.8.7 Probes

`TRPRobe:...` command are dedicated commands for the external trigger input.


- Common probe settings....1255
- Micro button and R&S ProbeMeter....1258
- Passive probes....1262
- Active voltage probes....1264
- Modular probes....1267
- Modular probes, deembedding....1271
- Predefined probes....1274
- Current probes....1275
- Probe attributes....1276
- Probe adapter R&S RT-Z1M....1278

### 24.8.7.1 Common probe settings

TRPRobe:SETup:STATe? 1255
PROBe<m>:SETup:STATe? 1255
TRPRobe:SETup:TYPE? 1255
PROBe<m>:SETup:TYPE? 1255
TRPRobe:SETup:NAME? 1256
PROBe<m>:SETup:NAME? 1256
TRPRobe:SETup:BANDwidth? 1256
PROBe<m>:SETup:BANDwidth? 1256
TRPRobe:SETup:ATTenuation[:AUTO]? 1256
PROBe<m>:SETup:ATTenuation[:AUTO]? 1256
PROBe<m>:SETup:OFFSet:AZERo 1257
PROBe<m>:SETup:OFFSet:USEautozero 1257
CHANnel<m>:EATScale 1257
CHANnel<m>:EATTenuation 1258

**TRPRobe:SETup:STATe?**
**PROBe<m>:SETup:STATe?**

Queries if the probe at the specified input channel is active (detected) or not active (not detected). To switch the probe on, use `CHANnel<m>:STATe`.

**Suffix:**
<m> 1..4

**Return values:**
<State> DETected | NDETected
*RST: NDETected

**Usage:** Query only
Asynchronous command

**TRPRobe:SETup:TYPE?**
**PROBe<m>:SETup:TYPE?**

Queries the type of the probe.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<Type> String containing one of the following values:
– None (no probe detected)
– Passive Probe
– active single-ended

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Type, Name, Bandwidth" on page 147

**TRPRobe:SETup:NAME?**
**PROBe<m>:SETup:NAME?**

Queries the name of the probe.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<Name> Name string

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Type, Name, Bandwidth" on page 147

**TRPRobe:SETup:BANDwidth?**
**PROBe<m>:SETup:BANDwidth?**

Queries the bandwidth of the probe.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<Bandwidth> Range: 10000 to 20E+9
*RST: 1E+9
Default unit: Hz

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Type, Name, Bandwidth" on page 147

**TRPRobe:SETup:ATTenuation[:AUTO]?**
**PROBe<m>:SETup:ATTenuation[:AUTO]?**

Queries the attenuation of the probe.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<PrbAttMdAuto> Range: 1E-3 to 1000
*RST: 1
Default unit: V/V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Probe unit, Auto attenuation" on page 149

**PROBe<m>:SETup:OFFSet:AZERo**

Performs an automatic correction of the zero error. If the DUT is ground-referenced, the AutoZero function can improve the measurement results.

See also: "Detect AutoZero, Use AutoZero" on page 147

**Suffix:**
<m> 1..4
Selects the input channel.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Detect AutoZero, Use AutoZero" on page 147

**PROBe<m>:SETup:OFFSet:USEautozero** <AutoZeroOffset>

Includes the AutoZero offset in measurement results. The auto zero error is detected with `PROBe<m>:SETup:OFFSet:AZERo`.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<AutoZeroOffset> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Firmware/software:** Version 2.70

**Manual operation:** See "Detect AutoZero, Use AutoZero" on page 147

**CHANnel<m>:EATScale** <ExtAttScl>

Sets the attenuation scale for an external divider.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<ExtAttScl> LIN | LOG
*RST: LIN

**Usage:** Asynchronous command

**Manual operation:** See "External attenuation: Scale, Attenuation" on page 144

**CHANnel<m>:EATTenuation** <ExtAtt>

Sets the attenuation of an external voltage divider that is part of the DUT before the measuring point. The external attenuation is included in the measurement, and the instrument shows the results that would be measured before the divider.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<ExtAtt> Values and unit depend on the selected scale (`CHANnel<m>:EATScale`).
Range: Linear scale: 1E-3 to 1E+6, logarithmic scale: -60 dB to 120 dB
Increment: 0.01
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "External attenuation: Scale, Attenuation" on page 144

### 24.8.7.2 Micro button and R&S ProbeMeter

PROBe<m>:SETup:MODE ... 1258
TRPRobe:SETup:DISPlaydiff ... 1260
PROBe<m>:SETup:DISPlaydiff ... 1260
TRPRobe:PMETer:VISibility ... 1260
PROBe<m>:PMETer:VISibility ... 1260
TRPRobe:PMETer:RESults:SINGle? ... 1260
PROBe<m>:PMETer:RESults:SINGle? ... 1260
TRPRobe:PMETer:RESults:COMMon? ... 1261
PROBe<m>:PMETer:RESults:COMMon? ... 1261
TRPRobe:PMETer:RESults:DIFFerential? ... 1261
PROBe<m>:PMETer:RESults:DIFFerential? ... 1261
TRPRobe:PMETer:RESults:NEGative? ... 1262
PROBe<m>:PMETer:RESults:NEGative? ... 1262
TRPRobe:PMETer:RESults:POSitive? ... 1262
PROBe<m>:PMETer:RESults:POSitive? ... 1262

**PROBe<m>:SETup:MODE** <Mode>

Select the action that is started when you press the micro button on the probe head.

See also: "Micro button action" on page 149.

**Suffix:**

<m> 1..4
Selects the input channel.

**Parameters:**

<Mode> RCONtinuous | RSINgle | AUToset | AZERo | SEToffsettomean | SITFile | NOACtion | FINDtriglevel | REPort | PROBemode | PRSetup

**RCONtinuous**
Run continuous: The acquisition is running as long as the probe button is pressed.

**RSINgle**
Run single: Starts a defined number of acquisitions (same as [Single] key).

**AUTOSET**
Starts the autoset procedure.

**AZero**
AutoZero: performs an automatic correction of the zero error.

**SEToffsettomean**
Set offset to mean: performs an automatic compensation for a DC component of the input signal.

**SITFile**
Save Image To File:
Directs the display image to a file. The `MMEMory:NAME` command defines the file name. The file format is defined with `HCOPy:DEVice<m>:LANGuage`.

**NOACtion**
Nothing is started on pressing the micro button.

**FINDtriglevel**
Sets the trigger level automatically to *0.5 * (MaxPeak – MinPeak)*. The function is not available for an external trigger source.

**REPort**
Creates and saves a report of the current results.

**PROBemode**
Only available for R&S RT-ZM probes. Changes the measurement mode of the probe.

**PRSetup**
Opens the "Probes Setup" dialog box.

*RST: RCONtinuous

**Usage:** Asynchronous command

**Manual operation:** See "Micro button action" on page 149

**TRPRobe:SETup:DISPlaydiff** <DisplayDiff>
**PROBe<m>:SETup:DISPlaydiff** <DisplayDiff>

Selects the input voltages to be measured by the ProbeMeter of an R&S differential active probe.

See also: "Differential active probes" on page 121.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<DisplayDiff> DIFFerential | SINGleended

**DIFFerential**
Measures differential and common mode voltages

**SINGleended**
Measures the voltage between the positive/negative signal socket and the ground.

*RST: DIFFerential

**Usage:** Asynchronous command

**Manual operation:** See "ProbeMeter" on page 150

**TRPRobe:PMETer:VISibility** <Visibility>
**PROBe<m>:PMETer:VISibility** <Visibility>

Activates the integrated R&S ProbeMeter of active R&S probes.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<Visibility> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Firmware/software:** FW 2.25

**Manual operation:** See "ProbeMeter" on page 150

**TRPRobe:PMETer:RESults:SINGle?**
**PROBe<m>:PMETer:RESults:SINGle?**

Returns the ProbeMeter measurement result of single-ended active R&S probes, the voltage measured between the probe tip and the ground.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**

<Result> Range: -100E+24 to 100E+24
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Firmware/software:** FW 2.25

**Manual operation:** See "ProbeMeter" on page 150

---

**TRPRobe:PMETer:RESults:COMMon?**
**PROBe<m>:PMETer:RESults:COMMon?**

Returns the ProbeMeter measurement result of differential active R&S probes: the common mode voltage, which is the mean voltage between the signal sockets and the ground socket.

**Suffix:**

<m> 1..4
Selects the input channel.

**Return values:**

<Result> Range: -100E+24 to 100E+24
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Firmware/software:** FW 2.25

**Manual operation:** See "ProbeMeter" on page 150

---

**TRPRobe:PMETer:RESults:DIFFerential?**
**PROBe<m>:PMETer:RESults:DIFFerential?**

Returns the ProbeMeter measurement result of differential active R&S probes, the differential voltage - the voltage between the positive and negative signal sockets.

**Suffix:**

<m> 1..4
Selects the input channel.

**Return values:**

<DiffMeasRes> Range: -100E+24 to 100E+24
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Firmware/software:** FW 2.25

**Manual operation:** See "ProbeMeter" on page 150

---

**TRPRobe:PMETer:RESults:NEGative?**
**PROBe<m>:PMETer:RESults:NEGative?**

Returns the ProbeMeter measurement result of differential active R&S probes, the voltage that is measured between the negative signal socket and the ground.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<SgEndNegMeasRes> Range: -100E+24 to 100E+24
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Firmware/software:** FW 2.25

**Manual operation:** See "ProbeMeter" on page 150

---

**TRPRobe:PMETer:RESults:POSitive?**
**PROBe<m>:PMETer:RESults:POSitive?**

Returns the ProbeMeter measurement result of differential active R&S probes, the voltage that is measured between the negative signal socket and the ground.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<Result> Range: -100E+24 to 100E+24
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Firmware/software:** FW 2.25

**Manual operation:** See "ProbeMeter" on page 150

### 24.8.7.3 Passive probes

TRPRobe:SETup:ATTenuation:MODE....1263
PROBe<m>:SETup:ATTenuation:MODE....1263

TRPRobe:SETup:ATTenuation:UNIT ..... 1263
PROBe<m>:SETup:ATTenuation:UNIT ..... 1263
TRPRobe:SETup:ATTenuation:MANual ..... 1263
PROBe<m>:SETup:ATTenuation:MANual ..... 1263
TRPRobe:SETup:GAIN:MANual ..... 1264
PROBe<m>:SETup:GAIN:MANual ..... 1264

**TRPRobe:SETup:ATTenuation:MODE** <AttenuationMode>
**PROBe<m>:SETup:ATTenuation:MODE** <PrbAttMd>

Set the mode to MANual if the instrument does not detect the probe.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<PrbAttMd> AUTO | MANual
*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 152

**TRPRobe:SETup:ATTenuation:UNIT** <AttenuationUnit>
**PROBe<m>:SETup:ATTenuation:UNIT** <AttenuationUnit>

Sets the unit for the connected probe type if `PROBe<m>:SETup:ATTenuation:MODE` on page 1263 is set to `MANual`.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<AttenuationUnit> V | A | W
Voltage probe (V), current probe (A), power probe (W)
*RST: V

**Usage:** Asynchronous command

**Manual operation:** See "Vertical unit, Attenuation, Gain" on page 153

**TRPRobe:SETup:ATTenuation:MANual** <PrbAttMdManual>
**PROBe<m>:SETup:ATTenuation:MANual** <ManualAttenuation>

Sets the attenuation for the connected probe if `PROBe<m>:SETup:ATTenuation:MODE` on page 1263 is set to `MANual`.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**

<ManualAttenuation> Range: 100E-6 to 10000
Increment: 0.1
*RST: 1
Default unit: Depends on the selected unit

**Usage:** Asynchronous command

**Manual operation:** See "Vertical unit, Attenuation, Gain" on page 153

---

**TRPRobe:SETup:GAIN:MANual** <GainManual>
**PROBe<m>:SETup:GAIN:MANual** <GainManual>

Sets the gain of a current probe.

**Suffix:**

<m> 1..4
Selects the input channel. The number of channels depends on the instrument.

**Parameters:**

<GainManual> Range: 100E-6 to 10000
Increment: 100E-6
*RST: 1
Default unit: V/A

**Usage:** Asynchronous command

**Manual operation:** See "Vertical unit, Attenuation, Gain" on page 153

### 24.8.7.4 Active voltage probes

TRPRobe:SETup:CMOFfset....1264
PROBe<m>:SETup:CMOFfset....1264
TRPRobe:SETup:ZAXV....1265
PROBe<m>:SETup:ZAXV....1265
TRPRobe:SETup:ACCoupling....1265
PROBe<m>:SETup:ACCoupling....1265
PROBe<m>:SETup:ADVanced:PMToffset....1265
PROBe<m>:SETup:ADVanced:RANGe....1266
TRPRobe:SETup:ADVanced:FILTer....1266
PROBe<m>:SETup:ADVanced:FILTer....1266
TRPRobe:SETup:ADVanced:AUDioverload....1267
PROBe<m>:SETup:ADVanced:AUDioverload....1267

---

**TRPRobe:SETup:CMOFfset** <CMOffset>
**PROBe<m>:SETup:CMOFfset** <CMOffset>

Sets the common-mode offset. The setting is only available for differential probes.

**Suffix:**

<m> 1..4
Selects the input channel.

**Parameters:**

<CMOffset>
Range: -100E+24 to 100E+24
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "CM offset" on page 155

---

**TRPRobe:SETup:ZAXV** <ExtAttRTZA15>
**PROBe<m>:SETup:ZAXV** <ExtAttRTZA15>

If you use the external attenuator R&S RT-ZA15 together with one of the differential active probes R&S RT-ZD10/20/30, enable it to include the external attenuation in the measurements.

**Suffix:**

<m>
1..4
Selects the input channel.

**Parameters:**

<ExtAttRTZA15>
ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Probe attenuator RT-ZA15" on page 155

---

**TRPRobe:SETup:ACCoupling** <ProbeCouplingAC>
**PROBe<m>:SETup:ACCoupling** <ProbeCouplingAC>

Enables AC coupling in the R&S RT-ZPR20 probe, which removes DC and very low-frequency components. The R&S RT-ZPR20 probe requires 50 Ω input termination, for which the channel AC coupling is not available. The probe setting allows AC coupling also at 50 Ω inputs.

**Suffix:**

<m>
1..4
Selects the input channel.

**Parameters:**

<ProbeCouplingAC>
ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "AC Coupling" on page 157

---

**PROBe<m>:SETup:ADVanced:PMToffset**

Sets the measured ProbeMeter value as offset. Make sure that the ProbeMeter is active before you use this command.

Only available for power rail probes R&S RT-ZPR.

**Suffix:**
<m> 1..4
Selects the input channel.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Copy to offset" on page 157

---

**PROBe<m>:SETup:ADVanced:RANGe** <ProbeRange>

Sets the voltage range of a R&S RT-ZHD probe.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<ProbeRange> AUTO | MHIGh | MLOW

**AUTO**
The voltage range is set with `CHANnel<m>:SCALe`.

**MHIGh**
Sets the higher voltage range of the connected probe. To query the value, use `PROBe<m>:SETup:ATTenuation[:AUTO]?`.

**MLOW**
Sets the lower voltage range of the connected probe. To query the value, use `PROBe<m>:SETup:ATTenuation[:AUTO]?`.

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 158

---

**TRPRobe:SETup:ADVanced:FILTer** <BwFilterSt>
**PROBe<m>:SETup:ADVanced:FILTer** <BwFilterSt>

Enables the lowpass filter in the probe control box.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<BwFilterSt> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "BW limit" on page 158

**TRPRobe:SETup:ADVanced:AUDioverload** <AudibOvrrg>
**PROBe<m>:SETup:ADVanced:AUDioverload** <Sound>

Activates the acustic overrange warning in the probe control box.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<Sound> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Buzzer" on page 158

### 24.8.7.5 Modular probes

The commands of this chapter are relevant for R&S RT-ZM modular probes. The commands are available in firmware version 3.40 and higher.

The suffix <m> selects the input channel to which the probe is connected.

TRPRobe:SETup:PRMode....1267
PROBe<m>:SETup:PRMode....1267
PROBe<m>:SETup:DMOFfset....1268
PROBe<m>:SETup:CMOFfset....1268
PROBe<m>:SETup:NOFFset....1269
PROBe<m>:SETup:POFFset....1269
PROBe<m>:SETup:TERM:STATe....1269
PROBe<m>:SETup:TERM:MODE....1270
PROBe<m>:SETup:TERM:MEASure?....1270
PROBe<m>:SETup:TERM:ADJust....1270

**TRPRobe:SETup:PRMode** <PrbMeasMd>
**PROBe<m>:SETup:PRMode** <MeasMode>

Sets the measurement mode of the modular probe.

**Suffix:**
<m> 1..4

**Parameters:**
<MeasMode> DMODe | CMODe | PMODe | NMODe

**DMODe**
Differential mode input voltage ($V_{dm}$), the voltage between the positive and negative input terminal.

**CMODe**
Common mode input voltage ($V_{cm}$) , the mean voltage between the positive and negative input terminal vs. ground.

**PMODe**
Positive single-ended input voltage ($V_p$). the voltage between the positive input terminal and ground.

**NMODe**
Negative single-ended input voltage ($V_N$). the voltage between the negative input terminal and ground.

*RST: DMODe

**Usage:** Asynchronous command

**Manual operation:** See "Probe mode" on page 159

---

**PROBe<m>:SETup:DMOFfset** <DMOffset>

Sets the differential offset to compensate a DC voltage applied to the positive and the negative input terminal.

**Suffix:**
<m> 1..4

**Parameters:**
<DMOffset> In "DM" probe mode (`DMODe`), this offset is used as channel offset and considered automatically for correction.

Range: -100E+24 to 100E+24
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "DM Offset, CM Offset, P Offset, N Offset" on page 160

---

**PROBe<m>:SETup:CMOFfset** <CMOffset>

Sets the common-mode offset. The setting is only available for differential probes.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<CMOffset>

Range: -100E+24 to 100E+24
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "CM offset" on page 155

**PROBe<m>:SETup:NOFFset** <NOffset>

Sets the negative offset to compensate a DC voltage applied to the negative input terminal (Vp) referenced to ground.

**Suffix:**
<m> 1..4

**Parameters:**
<NOffset> In "N" probe mode (`NMODe`), this offset is used as channel offset and considered automatically for correction.

Range: -100E+24 to 100E+24
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "DM Offset, CM Offset, P Offset, N Offset" on page 160

**PROBe<m>:SETup:POFFset** <POffset>

Sets the positive offset to compensate a DC voltage applied to the positive input terminal (Vp) referenced to ground.

**Suffix:**
<m> 1..4

**Parameters:**
<POffset> In "P" probe mode (`PMODe`), this offset is used as channel offset and considered automatically for correction.

Range: -100E+24 to 100E+24
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "DM Offset, CM Offset, P Offset, N Offset" on page 160

**PROBe<m>:SETup:TERM:STATe** <VoltageState>

Activates the instrument control of the termination voltage.

**Suffix:**
<m> 1..4

**Parameters:**
<VoltageState> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 162

**PROBe<m>:SETup:TERM:MODE** <Mode>

Setting the termination voltage is relevant if you use the R&S RT-ZMA40 SMA module.

You can set a termination voltage to correct the internal 50 Ω termination of the SMA module by the common mode voltage. To control the termination voltage by the instrument, connect the VT terminal of the R&S RT-ZM probe amplifier to the VT terminal of the SMA module using the red DC lead (see R&S RT-ZM User Manual).

**Suffix:**
<m> 1..4

**Parameters:**
<Mode> AUTO | MANual

**AUTO**
The instrument uses the measured common mode voltage to control the termination.

**MANual**
Enter the voltage to be used for termination.

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 162

**PROBe<m>:SETup:TERM:MEASure?**

Returns the measured common mode voltage.

**Suffix:**
<m> 1..4

**Return values:**
<VoltageMeas> Common mode voltage

Range: -100E+24 to 100E+24
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Measurement" on page 163

**PROBe<m>:SETup:TERM:ADJust** <VoltageAdjust>

Sets the voltage to be used for termination correction.

**Suffix:**
<m> 1..4

**Parameters:**

<VoltageAdjust> Correction voltage

Range: -100E+24 to 100E+24
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Adjustment" on page 162

### 24.8.7.6 Modular probes, deembedding

The commands of this chapter are relevant for R&S RT-ZM modular probes when used for measurements on signal frequencies of 4 GHz and higher. The commands are available in firmware version 4.00 and higher.

See also: `PROBe<m>:SETup:PRMode` on page 1267

PROBe<m>:DEEMbedding:PMODe....1271
PROBe<m>:DEEMbedding:TIPModule[:SELect]....1271
PROBe<m>:DEEMbedding:TIPModule:CSPFile....1272
PROBe<m>:DEEMbedding:TIPModule:ZMA<n>:SUBModule....1272
PROBe<m>:DEEMbedding:BANDwidth....1272
PROBe<m>:DEEMbedding:MODE....1273
PROBe<m>:DEEMbedding:REMProbeload....1273
PROBe<m>:DEEMbedding:LOAD:IMPedance....1274

---

**PROBe<m>:DEEMbedding:PMODe** <MeasMode>

**Suffix:**

<m> 1..4

**Parameters:**

<MeasMode> DMODe | CMODe | PMODe | NMODe

See `PROBe<m>:SETup:PRMode` on page 1267.

*RST: DMODe

**Usage:** Asynchronous command

---

**PROBe<m>:DEEMbedding:TIPModule[:SELect]** <ProbeTip>

Selects the tip module that is used for measurement.

**Suffix:**

<m> 1..4
Selects the channel to which the probe is connected.

**Parameters:**

<ProbeTip> NONE | ZMA10 | ZMA12 | ZMA14 | ZMA15 | ZMA30 | ZMA40 | ZMA50 | CUST

*RST: ZMA10

**Usage:** Asynchronous command

**Manual operation:** See "Select probe tip module, Pin spacing" on page 160

---

**PROBe<m>:DEEMbedding:TIPModule:CSPFile** <SParamFile>

Sets the file name, file format and path of the S-parameter file, and loads the S-parameters of user-defined tip modules

**Suffix:**
<m> 1..4

**Parameters:**
<SParamFile> String with path, file name, and extension.

**Usage:** Asynchronous command

---

**PROBe<m>:DEEMbedding:TIPModule:ZMA<n>:SUBModule** <Value>

Defines additional settings for some tip modules:

For the browser module R&S RT-ZMA30, measure the space between the pins, and select the appropriate value.

If R&S RT-ZMA40 is selected, choose the used submodule: semi-rigid cables or none.

**Suffix:**
<m> 1..4
Selects the channel to which the probe is connected.

<n> 30 (R&S RT-ZMA30) or 40 (R&S RT-ZMA40)

**Parameters:**
<Value> D16 | D25 | D45 | D75 | NONE | SRCS

**For R&S RT-ZMA30:**
D16 = 1.6 mm
D25 = 2.5 mm
D45 = 4.5 mm
D75 = 7.5 mm
**For R&S RT-ZMA40:**
NONE = no submodule
SRCS = semi-rigid cables

**Usage:** Asynchronous command

**Manual operation:** See "Select probe tip module, Pin spacing" on page 160

---

**PROBe<m>:DEEMbedding:BANDwidth** <ProbeBandwidth>

Sets the maximum bandwidth until which the signal is corrected. This maximum value is the minimum bandwidth value of probe, probe tip and oscilloscope bandwidth, and can not be higher than the highest frequency in a used S-parameter file. Consider that most tip modules support the full bandwidth of the probe amplifier, but some tip modules have limited bandwidth.

**Suffix:**
<m> 1..4
Selects the channel to which the probe is connected.

**Parameters:**
<ProbeBandwidth> Range: 10000 to 16E+9
Increment: 10
*RST: 4E+9
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Effective bandwidth" on page 161

---

**PROBe<m>:DEEMbedding:MODE** <Mode>

Sets the probing mode.

**Suffix:**
<m> 1..4
Selects the channel to which the probe is connected.

**Parameters:**
<Mode> PROBing | TERMinating

**PROBing**
For measurements with a high-ohmic probe and measure on an existing line in parallel to the load.

**TERMinating**
For measurements if the measuring equipment is the load of the line.

*RST: PROBing

**Usage:** Asynchronous command

**Manual operation:** See "Probing, Terminating" on page 161

---

**PROBe<m>:DEEMbedding:REMProbeload** <RemoveProbeLoad>

Removes the loading of the probe if `DEEMbedding<m>:COMPonent<n>:MODE` is set to `PROBing`.

If the probe loading is removed, you see the signal that would be at the measurement point if the probe's input impedance is ideal (infinite impedance).

Without removing the loading, you see the real signal at the measurement point, including the probe loading.

**Suffix:**
<m> 1..4
Selects the channel to which the probe is connected.

**Parameters:**
<RemoveProbeLoad> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Remove probe load" on page 161

---

**PROBe<m>:DEEMbedding:LOAD:IMPedance** <SourceImpedance>

Shows the source impedance of the DUT. If an R&S RT-ZM probe is connected, the value depends on the selected probe mode: common mode, differential, or single-ended measurement.

**Suffix:**
<m> 1..4
Selects the channel to which the probe is connected.

**Parameters:**
<SourceImpedance> Range: 100E-15 to 1E+9
Increment: 1
*RST: 50
Default unit: Ω

**Usage:** Asynchronous command

**Manual operation:** See "Source impedance" on page 161

### 24.8.7.7 Predefined probes

TRPRobe:SETup:ATTenuation:DEFProbe ..... 1274
PROBe<m>:SETup:ATTenuation:DEFProbe ..... 1274
PROBe<m>:SETup:OFFSet:TOMean ..... 1275

---

**TRPRobe:SETup:ATTenuation:DEFProbe** <SelcPredefPrb>
**PROBe<m>:SETup:ATTenuation:DEFProbe** <PredefinedProbe>

Selects a predefined probe. These are probes that are not recognized automatically but the parameters of the probe are known to the instrument.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<PredefinedProbe> ZC10 | ZC20 | ZC30 | ZD01A100 | ZD01A1000 | ZZ80 | FREE | ZS10L | ZD02 | ZD08 | ZC02100 | ZC021000 | ZC03 | ZD002A10 | ZD002A100 | ZD003A20 | ZD003A200 | ZC3110 | ZC311 | ZC3101

**ZC10 | ZC20 | ZC30 | ZZ80 | ZS10L | ZD02 | ZD08 | ZC03**
Type of the probe

**ZD01A100 | ZD01A1000 | ZD002A10 | ZD002A100 | ZD003A20 | ZD003A200**
High voltage differential probes R&S RT-ZD0xx, attenuation ratio according to the setting on the probe.
A10 = 10:1
A20 = 20:1
A100 = 100:1
A200 = 200:1
A1000 = 1000:1

**ZC02100 | ZC021000**
Current probes 100 A or 1000 A according to the setting on the probe.

**FREE**
Any other probe that is not recognized by the instrument.

*RST: FREE

**Usage:** Asynchronous command

**Manual operation:** See "Predefined probe" on page 164

---

**PROBe<m>:SETup:OFFSet:TOMean**

Performs an automatic compensation for a DC component of the specified input signal using the result of a background mean measurement.

**Suffix:**
<m> 1..4
Selects the input channel.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Offset to mean" on page 165

### 24.8.7.8 Current probes

To set up R&S RT-ZC10 and R&S RT-ZC20, use `PROBe<m>:SETup:ATTenuation:DEFProbe`.

TRPRobe:SETup:DEGauss....1275
PROBe<m>:SETup:DEGauss....1275
PROBe<m>:SETup:OFFSet:STPRobe....1276
PROBe<m>:SETup:OFFSet:ZADJust....1276

---

**TRPRobe:SETup:DEGauss**
**PROBe<m>:SETup:DEGauss**

Demagnetizes the core if it has been magnetized by switching the power on and off, or by an excessive input. Always carry out demagnetizing before measurement. The demagnetizing process takes about one second.

**Suffix:**
<m> 1..4
Selects the input channel.

**Usage:** Event
Asynchronous command

**Firmware/software:** FW 2.50

**Manual operation:** See "DeGauss" on page 167

---

**PROBe<m>:SETup:OFFSet:STPRobe**

Saves the zero adjust value in the probe box. If you connect the probe to another channel or to another R&S RTx oscilloscope, the value is read out again.

**Suffix:**
<m> 1..4
Selects the input channel.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save to probe" on page 168

---

**PROBe<m>:SETup:OFFSet:ZADJust** <ZroAdjVal>

set the waveform to zero position. It corrects the effect of a voltage offset or temperature drift. To set the value by the instrument, use `PROBe<m>:SETup:OFFSet:AZERo`.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<ZroAdjVal> Range: -100 to 100
Increment: 0.1
*RST: 0
Default unit: %

**Usage:** Asynchronous command

**Firmware/software:** FW 2.50

**Manual operation:** See "Zero adjust" on page 168

### 24.8.7.9 Probe attributes

TRPRobe:ID:SWVersion? 1277
PROBe<m>:ID:SWVersion? 1277
TRPRobe:ID:PRDate? 1277
PROBe<m>:ID:PRDate? 1277
TRPRobe:ID:PARTnumber? 1277
PROBe<m>:ID:PARTnumber? 1277

TRPRobe:ID:SRNumber?....1278
PROBe<m>:ID:SRNumber?....1278
TRPRobe:SETup:CAPacitance?....1278
PROBe<m>:SETup:CAPacitance?....1278
TRPRobe:SETup:IMPedance?....1278
PROBe<m>:SETup:IMPedance?....1278

**TRPRobe:ID:SWVersion?**
**PROBe<m>:ID:SWVersion?**

Queries the version of the probe firmware.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<Softwareversion> Version number in a string.

**Usage:** Query only
Asynchronous command

**TRPRobe:ID:PRDate?**
**PROBe<m>:ID:PRDate?**

Queries the production date of the probe.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<ProductionDate> Date in a string.

**Usage:** Query only
Asynchronous command

**TRPRobe:ID:PARTnumber?**
**PROBe<m>:ID:PARTnumber?**

Queries the R&S part number of the probe.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<PartNumber> Part number in a string.

**Usage:** Query only
Asynchronous command

**TRPRobe:ID:SRNumber?**
**PROBe<m>:ID:SRNumber?**

Queries the serial number of the probe.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<SerialNo> Serial number in a string.

**Usage:** Query only
Asynchronous command

**TRPRobe:SETup:CAPacitance?**
**PROBe<m>:SETup:CAPacitance?**

Queries the input capacitance of the probe.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<InputCapacity> Range: 100E-15 to 1E-9
*RST: 10E-12
Default unit: F

**Usage:** Query only
Asynchronous command

**TRPRobe:SETup:IMPedance?**
**PROBe<m>:SETup:IMPedance?**

Queries the termination of the probe.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<InputImpedance> Range: 0 to 1E+9
*RST: 50
Default unit: Ω

**Usage:** Query only
Asynchronous command

### 24.8.7.10 Probe adapter R&S RT-Z1M

PROBe<m>:CONVerter:TYPE? ..... 1279
PROBe<m>:CONVerter:BANDwidth ..... 1279
PROBe<m>:CONVerter:SELFalign ..... 1279

**PROBe<m>:CONVerter:TYPE?**

Queries the adapter status, whether the instrument identified the adapter.

**Suffix:**
<m> 1..4
Selects the input channel.

**Return values:**
<Type> NONE | Z1M
*RST: NONE

**Usage:** Query only
Asynchronous command

**PROBe<m>:CONVerter:BANDwidth** <BandwidthLimit>

Sets the bandwidth limit of the R&S RT-Z1M probe adapter.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<BandwidthLimit> FULL | B200 | B20
B200 = 200 MHz, B20 = 20 MHz
*RST: FULL

**Usage:** Asynchronous command

**Manual operation:** See "Bandwidth limit" on page 174

**PROBe<m>:CONVerter:SELFalign**

Starts the self-alignment of the R&S RT-Z1M probe adapter. When the adapter is detached from the oscilloscope, the values of the self-alignment are discarded, and the original factory-alignment is reloaded. Thus, repeat the self-alignment when you reattach the probe.

**Suffix:**
<m> 1..4
Selects the input channel.

**Example:**
```
PROBe2:CONVerter:TYPE?
<-- Z1M
PROBe2:CONVerter:BANDwidth B200
PROBe2:CONVerter:SELFalign
```

Checks the adapter state on channel 2, sets the bandwidth limit to 200 MHz, and starts the self-alignment of the adapter.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Selfalignment" on page 174

## 24.8.8 R&S RT-ZVC probe


- Probe setup....1280
- Resolution and data....1292


### 24.8.8.1 Probe setup

In the `ZVC:Z<m>:` commands, the following suffixes are used:

- `<m>` selects the multi-channel probe. The available range is `1..2`
- `<n>` selects the voltage or current channel. The available range depends on the probe characteristics and is `1..2` or `1..4`

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172


ZVC:BANDwidth....1281
ZVC:TYPE....1281
ZVC:Z<m>:I<n>:BANDwidth?....1281
ZVC:Z<m>:I<n>:IMPedance....1282
ZVC:Z<m>:I<n>:OFFSet....1282
ZVC:RESCoupled....1283
ZVC:Z<m>:I<n>:OVERload:RSTO....1283
ZVC:Z<m>:I<n>:OVERload:VALue?....1284
ZVC:Z<m>:I<n>:POSition....1284
ZVC:Z<m>:I<n>:SCALe....1284
ZVC:Z<m>:I<n>:SHUNt:EVALue....1285
ZVC:Z<m>:I<n>:SHUNt:MODE....1285
ZVC:Z<m>:I<n>:SHUNt:MAXCurrent....1285
ZVC:Z<m>:I<n>:SHUNt:MXCValue?....1286
ZVC:Z<m>:I<n>:SHUNt:MAXVoltage....1286
ZVC:Z<m>:I<n>:SKEW....1286
ZVC:Z<m>:I<n>[:STATe]....1287
ZVC:Z<m>:V<n>:BANDwidth?....1287
ZVC:Z<m>:V<n>:IMPedance....1288
ZVC:Z<m>:V<n>:OFFSet....1288
ZVC:Z<m>:V<n>:OVERload:RSTO....1288
ZVC:Z<m>:V<n>:OVERload:VALue?....1288

ZVC:Z<m>:V<n>:POSition ..... 1289
ZVC:Z<m>:V<n>:SCALe ..... 1289
ZVC:Z<m>:V<n>:SKEW ..... 1290
ZVC:Z<m>:V<n>[:STATe] ..... 1290
ZVC:Z<m>:ID:NAME? ..... 1290
ZVC:Z<m>:ID:PARTnumber? ..... 1290
ZVC:Z<m>:ID:SRNumber? ..... 1291
ZVC:Z<m>:ID:SWVersion? ..... 1291
ZVC:Z<m>:I<n>:ZERComp:DETect ..... 1291
ZVC:Z<m>:I<n>:ZERComp:USE ..... 1291
ZVC:Z<m>:I<n>:ZERComp:STATe? ..... 1292

**ZVC:BANDwidth** <Bandwidth>

Sets the bandwidth limit of the probe. The bandwidth specifies the maximum frequency at which a purely sinusoidal signal is still transferred at 89 % (0.1 dB) of its amplitude.

**Parameters:**

<Bandwidth>
Range: 5000 to 1E+6
Increment: 5000
*RST: 1E+6
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Bandwidth" on page 177

**ZVC:TYPE** <DecimationMode>

Sets the decimation mode for the R&S RT-ZVC probe. Decimation reduces the data stream of the ADC to a stream of waveform points with lower sample rate and a less precise time resolution.

**Parameters:**

<DecimationMode>
SAMPle | PDETect | HRESolution
*RST: SAMPle

**Usage:** Asynchronous command

**Manual operation:** See "Decimation mode" on page 177

**ZVC:Z<m>:I<n>:BANDwidth?**

Queries the bandwidth of the current channel. You can set the probe bandwidth with `ZVC:BANDwidth`.

The bandwidth of some current channels is restricted to 300KHz due to their vertical settings.

**Suffix:**

<m> 1..2

<n> 1..4

**Return values:**

<Bandwidth> Range: 5000 to 1E+6
Increment: 5000
*RST: 1E+6
Default unit: Hz

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Bandwidth" on page 182

---

**ZVC:Z<m>:I<n>:IMPedance** <MeasImp>

Sets the current impedance of the probe channel for power calculations and measurements.

**Suffix:**

<m> 1..2

<n> 1..4

**Parameters:**

<MeasImp> Range: 1 to 100E+3
Increment: 1
*RST: 50
Default unit: Ω

**Usage:** Asynchronous command

**Manual operation:** See "Measurement Impedance" on page 178

---

**ZVC:Z<m>:I<n>:OFFSet** <VerticalOffset>

Sets the offset current for the current channel.

**Suffix:**

<m> 1..2

<n> 1..4

**Parameters:**

<VerticalOffset> Range: -1 to 1
Increment: 0.01
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Offset" on page 176

**ZVC:RESCoupled** <CoupleToAnalog>

Sets the resolution of all R&S RT-ZVC channels.

**Parameters:**

<CoupleToAnalog> ON | OFF

**ON**
The resolution of the analog channels is applied to R&S RT-ZVC channels. The signal is automatically interpolated or decimated to get the analog resolution.

**OFF**
The resolution of R&S RT-ZVC channels is set in a way so that the record length of the waveforms is minimum 1000 samples.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Couple resolution" on page 178

**ZVC:Z<m>:I<n>:OVERload:RSTO** <Settings>

Resets the overload indication at the probe.

In internal shunt mode (`ZVC:Z<m>:I<n>:SHUNt:MODE` is set to `INTShunt`), chooses whether to keep the operation range adjusted by the R&S RT-ZVC during overload or to restore the original settings for the given current channel.

In external shunt mode (`ZVC:Z<m>:I<n>:SHUNt:MODE` is set to `EXTShunt`), there is no adjustment of the operation range during overload.

**Suffix:**

<m> 1..2

<n> 1..4

**Setting parameters:**

<Settings> ADJusted | ORIGinal

**ADJusted**
The operation range that is automatically adjusted by the probe during overload is kept.
In internal shunt mode, during an overload at a current channel, the probe switches automatically to the next higher range. If the 10A range is also exceeded, the amperemeter switches to external shunt mode to protect the probe against permanent damage.

**ORIGinal**
The original operation range is restored as before the overload and the adjustment of the operation range.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Reset Overload" on page 183

**ZVC:Z<m>:I<n>:OVERload:VALue?**

Queries if an overload of the current channel was detected.

**Suffix:**
<m> 1..2

<n> 1..4

**Return values:**
<Overload> ON | OFF
*RST: OFF

**Usage:** Query only
Asynchronous command

**ZVC:Z<m>:I<n>:POSition** <VertPosi>

Sets the vertical position of the indicated current channel as a graphical value.

**Suffix:**
<m> 1..2

<n> 1..4

**Parameters:**
<VertPosi> Range: -5 to 5
Increment: 0.02
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Position" on page 182

**ZVC:Z<m>:I<n>:SCALe** <VerticalScale>

Sets the vertical scale for the current channel in Volts per division. Increasing the scale compresses the display of the signal.

**Suffix:**
<m> 1..2

<n> 1..4

**Parameters:**
<VerticalScale> Range: 1E-15 to 10E+24
Increment: 10E-6
*RST: 3
Default unit: V/div

**Usage:** Asynchronous command

**Manual operation:** See "Scale" on page 176

**ZVC:Z<m>:I<n>:SHUNt:EVALue** <ExtShuntVal>

Defines the value of the external shunt resistor to calculate the correct current values.

**Suffix:**
<m> 1..2

<n> 1..4

**Parameters:**
<ExtShuntVal> Range: 10E-6 to 10000
Increment: 0.02
*RST: 1
Default unit: Ω

**Usage:** Asynchronous command

**Manual operation:** See " Shunt value" on page 183

**ZVC:Z<m>:I<n>:SHUNt:MODE** <ShuntMode>

Sets the internal or external shunt mode.

**Suffix:**
<m> 1..2

<n> 1..4

**Parameters:**
<ShuntMode> INTShunt | EXTShunt
*RST: INTShunt

**Usage:** Asynchronous command

**Manual operation:** See "Shunt mode" on page 182

**ZVC:Z<m>:I<n>:SHUNt:MAXCurrent** <IntShuntMeasRg>

Sets the maximum current and the internal shunt value.

The internal shunt and maximum current parameter pairs are defined as described in the table below.

| Parameter | Internal shunt | Maximum current |
|---|---|---|
| A100r01 | 10 A | 10 mΩ |
| A40R01 | 4.5 A | 10 mΩ |
| A45M10r | 45 mA | 10 Ω |
| A4M510r | 4.5 mA | 10 Ω |
| A45u10k | 45 µA | 10 KΩ |
| A4U510k | 4.5 µA | 10 KΩ |

**Suffix:**
<m> 1..2
<n> 1..4

**Parameters:**
<IntShuntMeasRg> A100r01 | A40R01 | A45M10r | A4M510r | A45u10k | A4U510k
*RST: A100r01

**Usage:** Asynchronous command

**Manual operation:** See "Max current / shunt" on page 183

---

**ZVC:Z<m>:I<n>:SHUNt:MXCValue?**

Queries the maximum current.

**Suffix:**
<m> 1..2
<n> 1..4

**Return values:**
<MaxCurrentValue> Default unit: A

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Max current" on page 183

---

**ZVC:Z<m>:I<n>:SHUNt:MAXVoltage** <ExtShuntMeasRg>

Sets the maximum voltage for the external shunt.

**Suffix:**
<m> 1..2
<n> 1..4

**Parameters:**
<ExtShuntMeasRg> V450m | V045m
V450m: 450 mV
V045m: 45 mV
*RST: V450m

**Usage:** Asynchronous command

**Manual operation:** See "Max voltage" on page 183

---

**ZVC:Z<m>:I<n>:SKEW** <DeskewOffset>

Sets the skew offset value for the current probe channel. This is a delay value, that is known from the circuit specifics but cannot be compensated by the instrument automatically.

**Suffix:**
<m> 1..2
<n> 1..4

**Parameters:**
<DeskewOffset> Range: -6E-6 to 6E-6
Increment: 200E-9
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Skew" on page 182

---

**ZVC:Z<m>:I<n>[:STATe]** <State>

Enables the corresponding current channel of the probe.

**Suffix:**
<m> 1..2
<n> 1..4

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 176

---

**ZVC:Z<m>:V<n>:BANDwidth?**

Queries the bandwidth of the voltage channel. You can set the probe bandwidth with `ZVC:BANDwidth`.

**Suffix:**
<m> 1..2
<n> 1..4

**Return values:**
<Bandwidth> Range: 5000 to 1E+6
Increment: 5000
*RST: 1E+6
Default unit: Hz

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Bandwidth" on page 180

**ZVC:Z<m>:V<n>:IMPedance** <MeasImp>

Sets the voltage impedance of the probe channel for power calculations and measurements.

**Suffix:**
<m> 1..2

<n> 1..4

**Parameters:**
<MeasImp>
Range: 1 to 100E+3
Increment: 1
*RST: 50
Default unit: Ω

**Usage:** Asynchronous command

**Manual operation:** See "Measurement Impedance" on page 178

**ZVC:Z<m>:V<n>:OFFSet** <VerticalOffset>

Sets the vertical offset for the voltage channel.

**Suffix:**
<m> 1..2

<n> 1..4

**Parameters:**
<VerticalOffset>
Range: -1 to 1
Increment: 0.01
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Offset" on page 176

**ZVC:Z<m>:V<n>:OVERload:RSTO**

Resets the overload indication at the probe.

**Suffix:**
<m> 1..2

<n> 1..4

**Usage:** Event
Asynchronous command

**Manual operation:** See "Reset Overload" on page 181

**ZVC:Z<m>:V<n>:OVERload:VALue?**

Queries if an overload of the voltage channel was detected.

**Suffix:**
<m> 1..2
<n> 1..4

**Return values:**
<Overload> ON | OFF
*RST: OFF

**Usage:** Query only
Asynchronous command

---

**ZVC:Z<m>:V<n>:POSition** <VertPosi>

Sets the vertical position of the indicated voltage channel as a graphical value.

**Suffix:**
<m> 1..2
<n> 1..4

**Parameters:**
<VertPosi> Range: -5 to 5
Increment: 0.02
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Position" on page 180

---

**ZVC:Z<m>:V<n>:SCALe** <VerticalScale>

Sets the vertical scale for the voltage channel in Volts per division. Increasing the scale compresses the display of the signal.

**Suffix:**
<m> 1..2
<n> 1..4

**Parameters:**
<VerticalScale> Range: 1E-15 to 10E+24
Increment: 10E-6
*RST: 3
Default unit: V/div

**Usage:** Asynchronous command

**Manual operation:** See "Scale" on page 176

**ZVC:Z<m>:V<n>:SKEW** <DeskewOffset>

Sets the skew offset value for the voltage probe channel. This is a delay value, that is known from the circuit specifics but cannot be compensated by the instrument automatically.

**Suffix:**
<m> 1..2

<n> 1..4

**Parameters:**
<DeskewOffset> Range: -6E-6 to 6E-6
Increment: 200E-9
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Skew" on page 180

**ZVC:Z<m>:V<n>[:STATe]** <State>

Enables the corresponding voltage channel of the probe.

**Suffix:**
<m> 1..2

<n> 1..4

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 176

**ZVC:Z<m>:ID:NAME?**

Queries the name of the probe.

**Suffix:**
<m> 1..2

**Return values:**
<Name>

**Usage:** Query only
Asynchronous command

**ZVC:Z<m>:ID:PARTnumber?**

Queries the R&S part number of the probe.

**Suffix:**
<m> 1..2

**Return values:**
<PartNumber>

**Usage:** Query only
Asynchronous command

---

**ZVC:Z<m>:ID:SRNumber?**

Queries the serial number of the probe.

**Suffix:**
<m> 1..2

**Return values:**
<SerialNo>

**Usage:** Query only
Asynchronous command

---

**ZVC:Z<m>:ID:SWVersion?**

Queries the version of the probe firmware.

**Suffix:**
<m> 1..2

**Return values:**
<Softwareversion>

**Usage:** Query only
Asynchronous command

---

**ZVC:Z<m>:I<n>:ZERComp:DETect**

Measures the zero offset, the mean value on a currentless DUT. If temperature changes, or if you change the shunt or other probe settings, repeat the measurement.

**Suffix:**
<m> 1..2

<n> 1..4

**Usage:** Event
Asynchronous command

**Manual operation:** See "ZeroComp" on page 179

---

**ZVC:Z<m>:I<n>:ZERComp:USE** <UseZeroComp>

Enables the automatic compensation of the measured zero offset.

**Suffix:**
<m> 1..2
<n> 1..4

**Parameters:**
<UseZeroComp> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Use" on page 179

---

**ZVC:Z<m>:I<n>:ZERComp:STATe?**

Returns the status of the zero compensation of the indicated current channel.

**Suffix:**
<m> 1..2
<n> 1..4

**Return values:**
<ZeroCompState> ON | OFF
ON: Zero compensation is used
Off: Zero Compensation is not used
*RST: OFF

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Zero Comp" on page 177

### 24.8.8.2 Resolution and data

ACQuire:POINts:ZVALue? 1292
ACQuire:ZRESolution? 1293
ZVC:Z<m>:I<n>:DATA:HEADer? 1293
ZVC:Z<m>:V<n>:DATA:HEADer? 1293
ZVC:Z<m>:I<n>:DATA[:VALues]? 1293
ZVC:Z<m>:V<n>:DATA[:VALues]? 1293

---

**ACQuire:POINts:ZVALue?**

Returns the current record length used by the R&S RT-ZVCmulti-channel power probe channel.

**Return values:**
<ZUIRecordLength> Range: 1000 to 400E6
Increment: 2
*RST: 1000
Default unit: pts

**Usage:** Query only
Asynchronous command

---

**ACQuire:ZRESolution?**

Returns the current resolution of the R&S RT-ZVCmulti-channel power probe channel.

**Return values:**

<ZUIResolution> Range: 1E-15 to 0.5
Increment: 10E-12
*RST: 500E-6
Default unit: s

**Usage:** Query only
Asynchronous command

---

**ZVC:Z<m>:I<n>:DATA:HEADer?**
**ZVC:Z<m>:V<n>:DATA:HEADer?**

Return the header of amperemeter and voltmeter waveform data.

***Table 24-2: Header data***

| Position | Meaning | Example |
|---|---|---|
| 1 | Start time in s | -0.0001 |
| 2 | Stop time in s | 0.0001 |
| 3 | Number of values in one waveform (record length) | 1000 |
| 4 | Number of values per sample interval. For most waveforms the result is 1, for peak detect and envelope waveforms it is 2. If the number is 2, the number of returned values is twice the number of values (record length).. | 1 |

**Suffix:**

<m> 1..2

<n> 1..4

**Example:**
```
ZVC:Z2:V2:DATA:HEADer?
<-- -0.0001,0.0001,1000,1
```

**Usage:** Query only

---

**ZVC:Z<m>:I<n>:DATA[:VALues]?**
**ZVC:Z<m>:V<n>:DATA[:VALues]?**

Return the waveform data of the amperemeter and voltmeter for transmission from the instrument to the controlling computer. The data can be used in MATLAB, for example.

Set the export format to ASCii or REAL,32 using `FORMat[:DATA]`. Export of raw data is not supported.

**Suffix:**

<m> 1..2

<n> 1..4

**Return values:**

<Data> Liste of values according to the format setting.

**Example:**

```
FORMat:DATA ASCii
ZVC:Z2:V2:DATA:HEADer?
<-- -0.0001,0.0001,1000,1
ZVC:Z2:V2:DATA:VaLues?
<-- 0.00083049136446789,-0.0042279558256269,-0.0076707201078534,
-0.011294682510197,-0.015915233641863,-0.018905002623796,
-0.022423267364502, ….
```

1000 values are returned.

**Usage:** Query only

### 24.8.9 Differential signals

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

DIFFerential<m>:STATe....1295
DIFFerential<m>:PSIGnal[:SELect]....1295
DIFFerential<m>:NSIGnal[:SELect]....1295
DIFFerential<m>:AOUTput....1295
DIFFerential<m>:BOUTput....1295
DIFFerential<m>:COMMon:STATe....1296
DIFFerential<m>:DIFFerential:STATe....1296
DIFFerential<m>:NSIGnal:STATe....1296
DIFFerential<m>:PSIGnal:STATe....1296
DIFFerential<m>:COMMon:SCALe....1296
DIFFerential<m>:DIFFerential:SCALe....1296
DIFFerential<m>:COUPling....1297
DIFFerential<m>:COMMon:OFFSet....1297
DIFFerential<m>:DIFFerential:OFFSet....1297
DIFFerential<m>:COMMon:POSition....1297
DIFFerential<m>:DIFFerential:POSition....1297

**DIFFerential<m>:STATe** <State>

Activates the differential signal. Two differential signals are available.

**Suffix:**
<m> 1..2
Differential signal

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 185

**DIFFerential<m>:PSIGnal[:SELect]** <Signal>
**DIFFerential<m>:NSIGnal[:SELect]** <Signal>

Select the analog input channel of the positive and negative signal.

Using the same input channel for both differential signals is not possible.

**Suffix:**
<m> 1..2
Differential signal

**Parameters:**
<Signal> C1W1 | C2W1 | C3W1 | C4W1
*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "P, N" on page 185

**DIFFerential<m>:AOUTput** <Output>
**DIFFerential<m>:BOUTput** <Output>

Select the waveform that is the result of differential processing. 2 output waveforms per digital signal can be defined.

The number of output waveforms depends on deembedding, see "Output: P, N, Diff, Common" on page 185.

If deembedding by software is active, use the `DIFFerential:...:STATe` commands instead of AOUTput and BOUTput.

If realtime deembedding is active, only `DIFFerential<m>:AOUTput` is relevant, because only one output waveform is available.

**Suffix:**
<m> 1..2
Differential signal

**Parameters:**
<Output> P | N | DIFF | COM

**P | N**
Positive or negative waveform.

**DIFF**
Differential waveform.

**COM**
Common mode waveform.

*RST: COM

**Usage:** Asynchronous command

**Manual operation:** See "Output: P, N, Diff, Common" on page 185

---

**DIFFerential<m>:COMMon:STATe** <State>
**DIFFerential<m>:DIFFerential:STATe** <State>
**DIFFerential<m>:NSIGnal:STATe** <State>
**DIFFerential<m>:PSIGnal:STATe** <State>

The commands enable the output waveforms if deembedding by software is active. In this case, all output waveforms can be displayed at the same time, and each output is enabled separately.

**Suffix:**
<m> 1..2
Differential signal

**Parameters:**
<State> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Output: P, N, Diff, Common" on page 185

---

**DIFFerential<m>:COMMon:SCALe** <Scale>
**DIFFerential<m>:DIFFerential:SCALe** <Scale>

Set the vertical scale of differential and common mode waveforms, respectively.

Vertical settings of P and N output are the vertical settings of the input channels.

**Suffix:**
<m> 1..2
Differential signal

**Parameters:**
<Scale> Scale value, given in Volts per division.

Range: 1E-3 to 100
Increment: 1E-3
*RST: 0.05
Default unit: V/div

**Usage:** Asynchronous command

**Manual operation:** See "Scale" on page 186

**DIFFerential<m>:COUPling** <ScaleCoupling>

If enabled for a differential signal, the vertical scales of P, N, differential and common mode waveforms are coupled. The scales are related as follows:

$VertScale_P = VertScale_N = VertScale_{CM} = VertScale_{Diff}/2$

You can disable the scale coupling and set an individual scale for each waveform.

**Suffix:**
<m> 1..2
Differential signal

**Parameters:**
<ScaleCoupling> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Scale coupling" on page 187

**DIFFerential<m>:COMMon:OFFSet** <Offset>
**DIFFerential<m>:DIFFerential:OFFSet** <Offset>

Set the offset of differential and common mode waveforms.

Vertical settings of P and N output are the vertical settings of the input channels.

**Suffix:**
<m> 1..2
Differential signal

**Parameters:**
<Offset> Offset value
Range: -1 to 1
Increment: 0.01
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Offset" on page 187

**DIFFerential<m>:COMMon:POSition** <Position>
**DIFFerential<m>:DIFFerential:POSition** <Position>

Set the vertical position of differential and common mode waveforms.

Vertical settings of P and N output are the vertical settings of the input channels.

**Suffix:**
<m> 1..2
Differential signal

**Parameters:**

<Position> Position value

Range: -5 to 5
Increment: 0.02
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Position" on page 187

### 24.8.10 Digital filter

CHANnel<m>:DIGFilter:STATe ..... 1298
CHANnel<m>:DIGFilter:CUToff ..... 1298
TRIGger<m>:RFReject? ..... 1299

**CHANnel<m>:DIGFilter:STATe** <State>

Enables the DSP filter for input channels.

**Suffix:**

<m> 1..4

**Parameters:**

<State> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "BW limit" on page 145

**CHANnel<m>:DIGFilter:CUToff** <CutOffLP>

Sets the limit frequency of the lowpass filter for input channels.

**Suffix:**

<m> 1..4
Selects the input channel. One filter is applied to a pair of channels - one filter for channels 1 and 2 and another filter for channels 3 and 4 (only 4-channel models).

**Parameters:**

<CutOffLP> Range: 100E+3 to instrument bandwidth
*RST: 1E+6
Default unit: Hz
Available filter values are:
100E+3, 200E+3, 300E+3, 500E+3, 1.0E+6, 2.0E+6, 3.0E+6, 5.0E+6, 10E+6, 20E+6, 30E+6, 50E+6, 100E+6, 500E+6, 1.0E+9, 1.5E+9, 2.0E+9, 2.5E+9, 3.0E+9, 3.5E+9, 4.0E+9, 4.5E+9, 5.0E+9, 5.5E+9, 6.0E+9, 6.5E+9, 7.0E+9, 7.5E+9, 8.0E+9, 8.5E+9, 9.0E+9, 9.5E+9, 10.0E+9, 10.5E+9, 11.0E+9, 11.5E+9, 12.0E+9, 12.5E+9, 13.0E+9, 13.5E+9, 14.0E+9, 14.5E+9, 15.0E+9, 15.5E+9, 16.0E+9

**Usage:** Asynchronous command

**Manual operation:** See "Acquisition BW" on page 145

---

**TRIGger<m>:RFReject?** <Bandwidth>

Returns the BW limit of the trigger source.

**Suffix:**

<m> 1..3
Irrelevant, omit the suffix.

**Return values:**

<Bandwidth> Default unit: Hz

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Trigger BW" on page 145

### 24.8.11 Skew

CHANnel<m>:SKEW:MANual ..... 1299
CHANnel<m>:SKEW:TIME ..... 1300
PROBe<m>:SKEState ..... 1300
CHANnel<m>:SKEW:EXTended ..... 1300

---

**CHANnel<m>:SKEW:MANual** <ManualCompens>

If enabled, the skew offset value (`CHANnel<m>:SKEW:TIME`) is used for compensation. This improves horizontal and trigger accuracy.

**Suffix:**

<m> 1..4
Selects the input channel.

**Parameters:**

<ManualCompens> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "User skew" on page 126

---

**CHANnel<m>:SKEW:TIME** <Offset>

Sets an delay value, that is known from the circuit specifics but cannot be compensated by the instrument automatically. It affects only the selected input channel.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<Offset> Range: -100E-9 to 100E-9
Increment: 10E-15
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Skew offset" on page 126

---

**PROBe<m>:SKEState** <ProbeSkew>

If enabled, the skew of all connected active probes is measured, displayed, and used for deskewing.

**Suffix:**
<m> 1..4
The suffix is irrelevant. The setting affects all active channels.

**Parameters:**
<ProbeSkew> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Probe skew" on page 126

---

**CHANnel<m>:SKEW:EXTended** <ExtendSkew>

Enables a higher skew value range that can be set with `CHANnel<m>:SKEW:TIME` on page 1300. The skew extension reduces the maximum number of acquisitions in the memory.

**Suffix:**
<m> 1..4
The suffix is irrelevant. The setting affects all active channels.

**Parameters:**
<ExtendSkew> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Extend user skew" on page 127

## 24.8.12 AUX OUT

CALibration:SOURce:FREQuency ..... 1301
CALibration:SOURce:STATe ..... 1301

**CALibration:SOURce:FREQuency** <Frequency>

Frequency value and waveform type of the internal calibration source, which is output at the [Aux Out] connector.

**Parameters:**

<Frequency> KHZ7900 | MHZ10 | KHZ15800 | MHZ31_25 | KHZ31600 | KHZ63200 | GHZ1 | MHZ250MB | MHZ250mb | GHZ1_20DB | MHZ62 | MHZ125 | MHZ250 | MHZ500 | KHZ505500 | MHZ1R | MHZ31R | MHZ31_25R | MHZ498S | MHZ1257_2S | MHZ2491S | MHZ4060S | MHZ4350S | MHZ4810S | MHZ4991S | MHZ5940S | MHZ1000S | MHZ1R | MHZ31R | MHZ31_25R | MHZ498S | MHZ1000S | MHZ1257_2S | MHZ2491S | MHZ4350S | MHZ4810S | MHZ4991S | MHZ4060S | MHZ5940S

**MHZ1R | MHZ31R | MHZ31_25R | MHZ498S | MHZ1000S | MHZ1257_2S | MHZ2491S | MHZ4350S | MHZ4810S | MHZ4991S | MHZ4060S | MHZ5940S**
Square wave 1MHz, 31 MHz, 31.25 MHz
Sine wave 498 MHz, 1000 MHz, 1257.2 MHz, 2491 MHz, 4350 MHz, 4810 MHz, 4991 MHz, 4060 MHz, 5940 MHz

*RST: MHZ1R

**Example:**
```
CAL:SOUR:FREQ GHZ1
CAL:SOUR:STAT ON
```
The commands activate the 1 GHz reference frequency.

**Usage:** Asynchronous command

**Manual operation:** See "Activate 1GHz Reference" on page 128

**CALibration:SOURce:STATe** <State>

Defines the state of the internal calibration source.

**Parameters:**

<State> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Activate 1GHz Reference" on page 128

### 24.8.13 High definition mode

HDEFinition:STATe....1302
HDEFinition:BWIDth....1302
HDEFinition:RESolution?....1302
FORMat:BORDer....1303

**HDEFinition:STATe** <State>

Activates the high definition mode of the instrument.

**Parameters:**

<State> ON | OFF

ON: high definition mode, up to 16 bit digital resolution
OFF: normal oscilloscope mode

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 137

**HDEFinition:BWIDth** <Bandwidth>

Sets the filter bandwidth for the high definition mode.

**Parameters:**

<Bandwidth> Range: 10000 to max. 2 GHz, depending on the instrument bandwidth.
Increment: 1000
*RST: 1E+6
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Bandwidth" on page 137

See "Bandwidth" on page 137 for bandwidth limits.

**HDEFinition:RESolution?**

Returns the resulting digital resolution in high definition mode.

**Return values:**

<Resolution> Range: 0 to 16
Increment: 0.1
*RST: 0
Default unit: bit

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Resolution" on page 137

**FORMat:BORDer** <ByteOrder>

Sets the endianess.

The command is only relevant for raw data export in high definition mode (16 bit word length).

**Parameters:**

<ByteOrder> LSBFirst | MSBFirst

LSB first: little endian, least significant byte first
MSB first: big endian, most significant byte first

*RST: LSBFirst

**Usage:** Asynchronous command

**Manual operation:** See "Byte order" on page 1153

### 24.8.14 Reference clock

Commands are related to the external reference input on the rear panel, and the output of the OCXO reference signal.

SENSe[:ROSCillator]:SOURce ..... 1303
SENSe[:ROSCillator]:EXTernal:FREQuency ..... 1303

**SENSe[:ROSCillator]:SOURce** <RefSource>

Enables the use of the external reference signal instead of the internal OCXO reference.

**Parameters:**

<RefSource> INTernal | EXTernal

*RST: INTernal

**Usage:** Asynchronous command

**Manual operation:** See "Use external reference" on page 128

**SENSe[:ROSCillator]:EXTernal:FREQuency** <ExternalRef>

Sets the frequency of an external reference input signal that is connected to the external reference input on the rear panel.

**Parameters:**

<ExternalRef> *RST: 10E+6
Default unit: Hz
Range is 1E+6 to 20E+6, and the increment is 1E+6.

**Usage:** Asynchronous command

**Manual operation:** See "External reference" on page 127

## 24.9 Trigger

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172


- Basic trigger settings....1304
- Edge trigger....1308
- Glitch trigger....1308
- Width trigger....1309
- Runt trigger....1311
- Window trigger....1313
- Timeout trigger....1316
- Interval trigger....1317
- Slew rate trigger....1319
- Setup & Hold....1321
- State trigger....1324
- Pattern trigger....1324
- High-speed serial pattern trigger (option R&S RTP-K140/K141)....1327
- CDR trigger....1348
- TV/Video trigger....1351
- Line trigger....1356
- Trigger qualification....1357
- Holdoff....1360
- Conditioning....1364
- Trigger sequence....1367
- Trigger control....1370
- Actions on trigger....1372
- Zone trigger (option R&S RTP-K19)....1374
- External trigger input....1375
- Acquisition info....1378


### 24.9.1 Basic trigger settings


TRIGger<m>:SOURce[:SELect]....1305
TRIGger<m>:TYPE....1306
TRIGger<m>:LEVel<n>[:VALue]....1306
TRIGger<m>:FINDlevel....1307
DISPlay:TRIGger:LINes....1307

**TRIGger<m>:SOURce[:SELect]** <SourceDetailed>

Selects the source of the trigger signal.

**Suffix:**

<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger
Available values depend on the selected trigger source. For input channels CHAN1...4, a trigger sequence can be configured.
For all other trigger sources, only suffix 1 is allowed.

**Parameters:**

<SourceDetailed> CHAN1 | CHANnel1 | CHAN2 | CHANnel2 | CHAN3 | CHANnel3 | CHAN4 | CHANnel4 | EXTernanalog | SBUS | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | LOGIC | MSOB1 | MSOB2 | MSOB3 | MSOB4 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | LINE

**CHAN1 = CHANnel1, CHAN2 = CHANnel2, CHAN3 = CHANnel3, CHAN4 = CHANnel4**
Input channels

**EXTernanalog**
External analog signal connected to the External Trigger Input. For this source, only the analog edge trigger is available.

**LINE**
The instrument generates the trigger from the AC power input and synchronizes the signal to the AC power frequency.

**SBUS**
Serial bus

**D0...D15**
Digital channels (option R&S RTP-B1)
See also: Chapter 24.18.4, "Trigger settings for digital signals and parallel buses", on page 2321

**LOGIc**
Logic combination of digital channels, used as trigger source (option R&S RTP-B1)

**MSOB1 | MSOB2 | MSOB3 | MSOB4**
Parallel bus (option R&S RTP-B1)

**Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4**
Input channels of R&S RT-ZVCmulti-channel power probe. Only available in the A-trigger with trigger type EDGE.

**DIFF1 | DIFF2 | COMMON1 | COMMON2**
Differential signals

*RST: CHAN1

**Usage:** Asynchronous command

**Manual operation:** See "[Source]" on page 37

---

**TRIGger<m>:TYPE** <Type>

Selects the trigger type to trigger on analog channels or the external trigger input.

See also: Chapter 6.3, "Trigger types", on page 196.

To trigger on digital channels and parallel buses, use `TRIGger<m>:PARallel:TYPE`.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger
For suffixes 2 and 3, the following trigger types are availabe: EDGE, GLITch, WIDTh, RUNT, WINDow, TIMeout, INTerval, SLEWrate.

**Parameters:**
<Type> EDGE | GLITch | WIDTh | RUNT | WINDow | TIMeout | INTerval | SLEWrate | STATe | PATTern | ANEDge | SERPattern | TV | | SETHold

Most of the type values are self-explanatory.

**SETHold**
SETHold: analyzes the relative timimg between a data signal and the synchronous clock signal. For trigger settings, see Chapter 24.9.10, "Setup & Hold", on page 1321.

**ANEDge**
Edge trigger for external trigger input. Only available if the trigger source is the external trigger input. This trigger type uses the analog input signal. For trigger settings, see Chapter 6.11.1, "External trigger setup", on page 257.

**SERPattern**
Serial pattern for signals with serial data patterns in relation to a clock signal, requires option R&S RTP-K140/K141. For trigger settings, see Chapter 24.9.13, "High-speed serial pattern trigger (option R&S RTP-K140/K141)", on page 1327.

*RST: EDGE

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 198

---

**TRIGger<m>:LEVel<n>[:VALue]** <Level>

Sets the trigger level for the specified event and source.

If the trigger source is serial bus, the trigger level is set by the thresholds in the protocol configuration.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

<n> 1..31
Indicates the trigger source:
1...4 = channel 1 to 4
5 = external trigger input
6...11 = not available
12...27 = R&S RT-ZVC input channels
28...31 = differential signals Diff1, Diff2 and Common1, Common2

**Parameters:**

<Level> Voltage for the trigger level.

Range: Depends on vertical scale, channel offset and other settings. The trigger level must be within the current display range.
Increment: 1E-3
*RST: 0
Default unit: V

**Example:** `TRIG:LEV5 0.01`
Sets the trigger level for the external trigger signal to 10 mV.
`TRIG2:LEV3 0.2`
Sets the trigger level for the B-event and B-trigger source channel 3 to 200 mV.

**Usage:** Asynchronous command

**Manual operation:** See "[Level]" on page 36

---

**TRIGger<m>:FINDlevel**

Sets the trigger level automatically. The command is only relevant if the trigger source is an analog channel CHAN1...4.

**Suffix:**

<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Usage:** Event
Asynchronous command

**Manual operation:** See "Find level" on page 199

---

**DISPlay:TRIGger:LINes** <State>

Hides or shows the trigger levels in the diagrams.

**Parameters:**

<State> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show trigger lines" on page 229

### 24.9.2 Edge trigger

TRIGger<m>:EDGE:SLOPe ..... 1308

**TRIGger<m>:EDGE:SLOPe** <Slope>

Defines the edge for the edge trigger event.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<Slope> POSitive | NEGative | EITHer
See Chapter 24.4.3, "Slope parameter", on page 1194.
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 199

### 24.9.3 Glitch trigger

TRIGger<m>:GLITch:POLarity ..... 1308
TRIGger<m>:GLITch:RANGe ..... 1308
TRIGger<m>:GLITch:WIDTh ..... 1309

**TRIGger<m>:GLITch:POLarity** <Polarity>

Defines the polarity of a pulse, that is the direction of the first pulse slope.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<Polarity> POSitive | NEGative | EITHer
See Chapter 24.4.4, "Polarity parameter", on page 1194.
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 200

**TRIGger<m>:GLITch:RANGe** <RangeMode>

Selects which glitches are identified: shorter or longer than the width specified using `TRIGger<m>:GLITch:WIDTh`.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<RangeMode> SHORter | LONGer

**SHORter**
Glitches shorter than the specified width are identified.

**LONGer**
Glitches longer than the specified width are identified.

*RST: SHORter

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 200

---

**TRIGger<m>:GLITch:WIDTh** <Width>

Sets the length of a glitch. The instrument triggers on pulses shorter or longer than this value, depnding on the `TRIGger<m>:GLITch:RANGe` command.

You need to know the expected pulse widths of the circuit to set the glitch width correctly.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<Width> Range: 100E-12 to 10000
Increment: 100E-6
*RST: 1E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Width" on page 200

## 24.9.4 Width trigger

TRIGger<m>:WIDTh:POLarity ..... 1309
TRIGger<m>:WIDTh:RANGe ..... 1310
TRIGger<m>:WIDTh:WIDTh ..... 1310
TRIGger<m>:WIDTh:DELTa ..... 1311

---

**TRIGger<m>:WIDTh:POLarity** <Polarity>

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**

<Polarity> POSitive | NEGative

See Chapter 24.4.4, "Polarity parameter", on page 1194.

*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 201

---

**TRIGger<m>:WIDTh:RANGe** <RangeMode>

Defines how the range of a pulse width is defined in relation to the width and delta specified using `TRIGger<m>:WIDTh:WIDTh` and `TRIGger<m>:WIDTh:DELTa`, respectively.

**Suffix:**

<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**

<RangeMode> WITHin | OUTSide | SHORter | LONGer

**WITHin**
Triggers on pulses inside a given range. The range is defined by the width ±delta.

**OUTSide**
Triggers on pulses outside a given range. The range is defined by the width ±delta.

**SHORter**
Triggers on pulses shorter than the given width.

**LONGer**
Triggers on pulses longer than the given width.

*RST: WITHin

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 202

---

**TRIGger<m>:WIDTh:WIDTh** <Width>

For the ranges "Within" and "Outside" (defined using `TRIGger<m>:WIDTh:RANGe`), the width defines the center of a range which is defined by the limits "±Delta" (see `TRIGger<m>:WIDTh:DELTa` on page 1311).

For the ranges "Shorter" and "Longer", the width defines the maximum and minimum pulse width, respectively.

**Suffix:**

<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**

<Width>
Range: 100E-12 to 10000
Increment: 100E-9
*RST: 5E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Width" on page 202

---

**TRIGger<m>:WIDTh:DELTa** <WidthDelta>

Defines a range around the width value specified using `TRIGger<m>:WIDTh:WIDTh`.

**Suffix:**

<m>
1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**

<WidthDelta>
Range: 0 to 432
Increment: 500E-12
*RST: 0
Default unit: s

**Usage:** Asynchronous command

## 24.9.5 Runt trigger

TRIGger<m>:RUNT:POLarity....1311
TRIGger<m>:LEVel<n>:RUNT:LOWer....1312
TRIGger<m>:LEVel<n>:RUNT:UPPer....1312
TRIGger<m>:RUNT:RANGe....1312
TRIGger<m>:RUNT:WIDTh....1313
TRIGger<m>:RUNT:DELTa....1313

---

**TRIGger<m>:RUNT:POLarity** <Polarity>

**Suffix:**

<m>
1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**

<Polarity>
POSitive | NEGative | EITHer
See Chapter 24.4.4, "Polarity parameter", on page 1194.
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 200

**TRIGger<m>:LEVel<n>:RUNT:LOWer** <Level>
**TRIGger<m>:LEVel<n>:RUNT:UPPer** <Level>

Set the lower and upper voltage thresholds.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

<n> 1..31
Indicates the trigger source:
1...4 = channel 1...4
5...27 = not available
28...31 = differential signals Diff1, Diff2 and Common1, Common2

**Parameters:**
<Level> Range: -10 to 10
Increment: 1E-3
*RST: Lower = -0.1, upper =0.1
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level" on page 203

**TRIGger<m>:RUNT:RANGe** <Mode>

Defines the time limit of the runt pulse in relation to the `TRIGger<m>:RUNT:WIDTh` and `TRIGger<m>:RUNT:DELTa` settings.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<Mode> ANY | LONGer | SHORter | WITHin | OUTSide

**ANY**
Triggers on all runts fulfilling the level condition, without time limitation.

**LONGer**
Triggers on runts longer than the given "Runt width".

**SHORter**
Triggers on runts shorter than the given "Runt width".

**WITHin**
Triggers if the runt length is inside a given time range. The range is defined by "Runt width" and "±Delta".

**OUTSide**
Triggers if the runt length is outside a given time range. The range is defined by "Runt width" and "±Delta".

*RST: ANY

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 203

---

**TRIGger<m>:RUNT:WIDTh** <Width>

Defines the upper or lower voltage threshold. This command is not available if `TRIGger<m>:RUNT:RANGe` is set to "ANY".

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<Width>
Range: 100E-12 to 10000
Increment: 100E-9
*RST: 5E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Runt width" on page 204

---

**TRIGger<m>:RUNT:DELTa** <WidthDelta>

Defines a range around the runt width specified using `TRIGger<m>:RUNT:WIDTh`. This command is only available if `TRIGger<m>:RUNT:RANGe` is set to "WITHin" or "OUTSide".

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<WidthDelta>
Range: 100E-12 to 864
Increment: 100E-9
*RST: 100E-12
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "±Delta" on page 204

### 24.9.6 Window trigger

TRIGger<m>:LEVel<n>:WINDow:LOWer....1314
TRIGger<m>:LEVel<n>:WINDow:UPPer....1314
TRIGger<m>:WINDow:RANGe....1314
TRIGger<m>:WINDow:TIME....1315
TRIGger<m>:WINDow:WIDTh....1315
TRIGger<m>:WINDow:DELTa....1316

**TRIGger<m>:LEVel<n>:WINDow:LOWer** <Level>
**TRIGger<m>:LEVel<n>:WINDow:UPPer** <Level>

Set the lower and upper voltage limits for the window.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

<n> 1..31
Indicates the trigger source:
1...4 = channel 1...4
5...27 = not available
28...31 = differential signals Diff1, Diff2 and Common1, Common2

**Parameters:**
<Level> Range: -10 to 10
Increment: 1E-3
*RST: Lower = -0.1, upper =0.1
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level" on page 205

---

**TRIGger<m>:WINDow:RANGe** <RangeMode>

Defines the signal run in relation to the window:

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<RangeMode> ENTer | EXIT | WITHin | OUTSide

**ENTer**
Triggers when the signal crosses the upper or lower level and thus enters the window made up of these two levels.

**EXIT**
Triggers when the signal leaves the window.

**WITHin**
Triggers if the signal stays between the upper and lower level for a specified time. The time is defined using the `TRIGger<m>:WINDow:TIME` command.

**OUTSide**
Triggers if the signal stays above the upper level or below the lower level for a specified time. The time is defined using the `TRIGger<m>:WINDow:TIME` command.

*RST: ENTer

**Usage:** Asynchronous command

**Manual operation:** See "Vertical condition" on page 205

---

**TRIGger<m>:WINDow:TIME** <TimeRangeMode>

Defines the limit of the window in relation to the time specified using `TRIGger<m>:WINDow:WIDTh` and `TRIGger<m>:WINDow:DELTa`. Time conditioning is available for `TRIGger<m>:WINDow:RANGe`= "WITHin" and "OUTSide".

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<TimeRangeMode> WITHin | OUTSide | SHORter | LONGer

**WITHin**
Triggers if the signal stays inside or outside the vertical window limits at least for the time *Width - Delta* and for *Width + Delta* at the most.

**OUTSide**
"Outside" is the opposite definition of "Within". The instrument triggers if the signal stays inside or outside the vertical window limits for a time shorter than *Width - Delta* or longer than *Width + Delta*.

**SHORter**
Triggers if the signal crosses vertical limits before the specified "Width" time is reached.

**LONGer**
Triggers if the signal crosses vertical limits before the specified "Width" time is reached.

*RST: WITHin

**Usage:** Asynchronous command

**Manual operation:** See "Time condition" on page 205

---

**TRIGger<m>:WINDow:WIDTh** <Width>

For the ranges "Within" and "Outside" (defined using `TRIGger<m>:WINDow:RANGe`), the width defines the center of a time range which is defined by the limits "±Delta" (see `TRIGger<m>:WINDow:DELTa` on page 1316).

For the ranges "Shorter" and "Longer", it defines the maximum and minimum time lapse, respectively.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**

<Width>
Range: 100E-12 to 10000
Increment: 100E-9
*RST: 5E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Width" on page 206

---

**TRIGger<m>:WINDow:DELTa** <WidthDelta>

Defines a range around the "Width" value specified using `TRIGger<m>:WINDow:WIDTh`.

**Suffix:**

<m>
1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**

<WidthDelta>
Range: 0 to 432
Increment: 500E-12
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "±Delta" on page 206

## 24.9.7 Timeout trigger

TRIGger<m>:TIMeout:RANGe....1316
TRIGger<m>:TIMeout:TIME....1317

---

**TRIGger<m>:TIMeout:RANGe** <TimeoutMode>

Defines the relation of the signal level to the trigger level.

**Suffix:**

<m>
1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**

<TimeoutMode>
HIGH | LOW | EITHer

**HIGH**
The signal level stays above the trigger level.

**LOW**
The signal level stays below the trigger level.

**EITHer**
The signal level stays above or below the trigger level.

*RST: HIGH

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 207

---

**TRIGger<m>:TIMeout:TIME** <Time>

Defines the time limit for the timeout at which the instrument triggers.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<Time> Range: 100E-12 to 10000
Increment: 100E-9
*RST: 100E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Time" on page 207

### 24.9.8 Interval trigger

TRIGger<m>:INTerval:SLOPe ..... 1317
TRIGger<m>:INTerval:RANGe ..... 1317
TRIGger<m>:INTerval:WIDTh ..... 1318
TRIGger<m>:INTerval:DELTa ..... 1318

---

**TRIGger<m>:INTerval:SLOPe** <Slope>

Sets the edge for the trigger. You can analyze the interval between positive edges or between negative edges.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<Slope> POSitive | NEGative
See Chapter 24.4.3, "Slope parameter", on page 1194.
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 208

---

**TRIGger<m>:INTerval:RANGe** <RangeMode>

Defines the range of an interval in relation to the interval width specified using `TRIGger<m>:INTerval:WIDTh` and `TRIGger<m>:INTerval:DELTa`.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<RangeMode> WITHin | OUTSide | SHORter | LONGer

**WITHin**
Triggers on pulses inside a given range. The range is defined by the interval width ±delta.

**OUTSide**
Triggers on pulses outside a given range. The range is defined by the interval width ±delta.

**SHORter**
Triggers on pulses shorter than the given interval width.

**LONGer**
Triggers on pulses longer than the given interval width.

*RST: OUTSide

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 208

---

**TRIGger<m>:INTerval:WIDTh** <Width>

Defines the time between two pulses.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<Width> Range: 100E-12 to 10000
Increment: 100E-9
*RST: 5E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Interv. width" on page 208

---

**TRIGger<m>:INTerval:DELTa** <WidthDelta>

Defines a range around the "Interval width" value specified using `TRIGger<m>:INTerval:WIDTh` on page 1318.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**

<WidthDelta>
Range: 0 to 10
Increment: 100E-9
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "±Delta" on page 208

### 24.9.9 Slew rate trigger

TRIGger<m>:SLEW:SLOPe ..... 1319
TRIGger<m>:LEVel<n>:SLEW:LOWer ..... 1319
TRIGger<m>:LEVel<n>:SLEW:UPPer ..... 1319
TRIGger<m>:SLEW:RANGe ..... 1320
TRIGger<m>:SLEW:RATE ..... 1320
TRIGger<m>:SLEW:DELTa ..... 1321

---

**TRIGger<m>:SLEW:SLOPe** <Slope>

Selects the edge type for the trigger event.

**Suffix:**

<m>
1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**

<Slope>
POSitive | NEGative | EITHer
See Chapter 24.4.3, "Slope parameter", on page 1194.
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 199

---

**TRIGger<m>:LEVel<n>:SLEW:LOWer** <Level>
**TRIGger<m>:LEVel<n>:SLEW:UPPer** <Level>

Set the lower and upper voltage thresholds. When the signal crosses a level, the slew rate measurement starts or stops depending on the selected slope.

**Suffix:**

<m>
1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

<n>
1..31
Indicates the trigger source:
1...4 = channel 1...4
5...27 = not available
28...31 = differential signals Diff1, Diff2 and Common1, Common2

**Parameters:**

<Level>
Range: -10 to 10
Increment: 1E-3
*RST: Lower = -0.1, upper =0.1
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Upper level" on page 209

---

**TRIGger<m>:SLEW:RANGe** <RangeMode>

Defines the time limit for the slew rate in relation to the upper or lower trigger level (see `TRIGger<m>:SLEW:RATE` on page 1320 and `TRIGger<m>:SLEW:DELTa` on page 1321). The time measurement starts when the signal crosses the first trigger level - the upper or lower level depending on the selected slope - and stops when the signal crosses the second level.

**Suffix:**

<m>
1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**

<RangeMode> INSRange | OUTRange | LTHan | GTHan

**INSRange**
Triggers on pulses inside a given range. The range is defined by the slew rate ±delta.

**OUTRange**
Triggers on pulses outside a given range. The range is defined by the slew rate ±delta.

**LTHan**
Triggers on pulses shorter than the given slew rate.

**GTHan**
Triggers on pulses longer than the given slew rate.

*RST: GTHan

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 209

---

**TRIGger<m>:SLEW:RATE** <Time>

For the ranges "Within" and "Outside", the slew rate defines the center of a range which is defined by the limits "±Delta".

For the ranges "Shorter" and "Longer", the slew rate defines the maximum and minimum slew rate limits, respectively. When the signal crosses this level, the slew rate measurement starts or stops depending on the selected slope (see `TRIGger<m>:SLEW:SLOPe` on page 1319).

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<Time> Range: 100E-12 to 10000
Increment: 100E-9
*RST: 100E-12
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Slew rate" on page 210

---

**TRIGger<m>:SLEW:DELTa** <TimeDelta>

Defines a time range around the slew rate specified using `TRIGger<m>:SLEW:RATE`.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<TimeDelta> Range: 0 to 10
Increment: 100E-9
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "±Delta" on page 210

### 24.9.10 Setup & Hold

The Setup & Hold trigger is only available for the A-event (Suffix = 1).

TRIGger<m>:SETHold:CSOurce:EDGE ..... 1321
TRIGger<m>:SETHold:CSOurce:LEVel ..... 1322
TRIGger<m>:SETHold:CSOurce[:VALue] ..... 1322
TRIGger<m>:SETHold:HTIMe ..... 1323
TRIGger<m>:SETHold:STIMe ..... 1323
TRIGger<m>:SCOupling ..... 1323

---

**TRIGger<m>:SETHold:CSOurce:EDGE** <ClockEdge>

Sets the edge of the clock signal to define the time reference point for the setup and hold time.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**

<ClockEdge> POSitive | NEGative | EITHer

See Chapter 24.4.3, "Slope parameter", on page 1194.

*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Clock edge" on page 211

---

**TRIGger<m>:SETHold:CSOurce:LEVel** <ClockLevel>

Sets the voltage level for the clock signal. Both this command and `TRIGger<m>:SETHold:CSOurce:EDGE` define the starting point for calculation of the setup and hold time.

**Suffix:**

<m> 1..3

Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**

<ClockLevel> Range: -10 to 10

Increment: 1E-3

*RST: 0

Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Clock level" on page 211

---

**TRIGger<m>:SETHold:CSOurce[:VALue]** <ClockSource>

Selects the source of the clock signal.

**Suffix:**

<m> 1..3

Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**

<ClockSource> CHAN1 | CHANnel1 | CHAN2 | CHANnel2 | CHAN3 | CHANnel3 | CHAN4 | CHANnel4

CHAN1 = CHANnel1, CHAN2 = CHANnel2, CHAN3 = CHANnel3, CHAN4 = CHANnel4

Input channel of the clock signal

*RST: CHAN1

**Usage:** Asynchronous command

**Manual operation:** See "Clock source" on page 211

**TRIGger<m>:SETHold:HTIMe** <HoldTime>

Sets the minimum time **after** the clock edge while the data signal must stay steady above or below the data level.

The hold time can be negative. In this case, the setup time is always positive. The setup/hold interval starts before the clock edge (setup time) and ends before the clock edge (hold time). If you change the negative hold time, the setup time is adjusted by the intrument.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<HoldTime> Range: -99.999E-9 to 100E-9
Increment: 1E-9
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Hold time" on page 212

**TRIGger<m>:SETHold:STIMe** <SetupTime>

Sets the minimum time **before** the clock edge while the data signal must stay steady above or below the data level.

The setup time can be negative. In this case, the hold time is always positive. The setup/hold interval starts after the clock edge (setup time) and ends after the clock edge (hold time). If you change the negative setup time, the hold time is adjusted by the instrument.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<SetupTime> Range: -99.999E-9 to 100E-9
Increment: 1E-9
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Setup time" on page 212

**TRIGger<m>:SCOupling** <TrigLevSrcCpl>

Sets the trigger levels of all used channels to the level of the data line (Data2clock trigger), clock line (state trigger) or source line (pattern trigger).

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<TrigLevSrcCpl> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Use data level for clock level" on page 212

### 24.9.11 State trigger

The state trigger combines the edge trigger settings with trigger qualification. It is only available for the A-event (Suffix = 1).

Use the following commands:

- `TRIGger<m>:EDGE:SLOPe` on page 1308
- `TRIGger<m>:LEVel<n>[:VALue]` on page 1306
- `TRIGger<m>:SCOupling` on page 1323
- To define the state, use the `TRIG:QUAL...[ENABle]` and `TRIG:QUAL...LOGic` commands, which are described in Chapter 24.9.17, "Trigger qualification", on page 1357.

### 24.9.12 Pattern trigger

The pattern trigger is only available for the A-event (Suffix = 1).

The pattern is defined using the commands:

- `TRIGger<m>:QUALify<n>:A[:ENABle]` on page 1358
- `TRIGger<m>:QUALify<n>:A:LOGic` on page 1359
- `TRIGger<m>:QUALify<n>:AB:LOGic` on page 1359

These are the commands for channel 1, use the similar commands for channels 2, 3, and 4.

TRIGger<m>:PATTern:MODE....1324
TRIGger<m>:PATTern:TIMeout:MODE....1325
TRIGger<m>:PATTern:TIMeout[:TIME]....1325
TRIGger<m>:PATTern:WIDTh:RANGe....1326
TRIGger<m>:PATTern:WIDTh[:WIDTh]....1326
TRIGger<m>:PATTern:WIDTh:DELTa....1327

---

**TRIGger<m>:PATTern:MODE** <Mode>

Adds additional time limitation to the pattern definition.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<Mode> OFF | TIMeout | WIDTh

**OFF**
No time limitation. The event occurs if the pattern condition is fulfilled.

**TIMeout**
Defines how long the result of the pattern condition must be true or false. The duration of the timeout is defined using `TRIGger<m>:PATTern:TIMeout[:TIME]`.

**WIDTh**
Defines a time range for keeping up the true result of the pattern condition. The range is defined using `TRIGger<m>:PATTern:WIDTh:RANGe`.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Additional settings: Timing" on page 215

---

**TRIGger<m>:PATTern:TIMeout:MODE** <TimeoutMode>

Defines the condition for the timeout.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<TimeoutMode> HIGH | LOW

**HIGH**
The result stays high.

**LOW**
The result stays low.

*RST: HIGH

**Usage:** Asynchronous command

**Manual operation:** See "Additional settings: Timing" on page 215

---

**TRIGger<m>:PATTern:TIMeout[:TIME]** <Time>

Defines how long the result of the pattern condition must be true or false.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**

<Time> Range: 100E-12 to 10000
Increment: 100E-9
*RST: 100E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Additional settings: Timing" on page 215

---

**TRIGger<m>:PATTern:WIDTh:RANGe** <WidthRangeMode>

Defines how the range of a pulse width is defined for keeping up the true result of the pattern condition. The width and delta are specified using `TRIGger<m>:PATTern:WIDTh[:WIDTh]` and `TRIGger<m>:PATTern:WIDTh:DELTa`, respectively.

**Suffix:**

<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**

<WidthRangeMode> WITHin | OUTSide | SHORter | LONGer

**WITHin**
Triggers on pulses inside a given range. The range is defined by the width ±delta.

**OUTSide**
Triggers on pulses outside a given range. The range is defined by the width ±delta.

**SHORter**
Triggers on pulses shorter than the given width.

**LONGer**
Triggers on pulses longer than the given width.

*RST: WITHin

**Usage:** Asynchronous command

**Manual operation:** See "Additional settings: Timing" on page 215

---

**TRIGger<m>:PATTern:WIDTh[:WIDTh]** <Width>

For the ranges "Within" and "Outside" (defined using `TRIGger<m>:PATTern:WIDTh:RANGe`), the width defines the center of a range which is defined by the limits "±Delta" (see `TRIGger<m>:PATTern:WIDTh:DELTa` on page 1327).

For the ranges "Shorter" and "Longer", the width defines the maximum and minimum pulse width, respectively.

**Suffix:**

<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**

<Width>
Range: 100E-12 to 10000
Increment: 100E-9
*RST: 5E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Additional settings: Timing" on page 215

---

**TRIGger<m>:PATTern:WIDTh:DELTa** <WidthDelta>

Defines a range around the width value specified using `TRIGger<m>:PATTern:WIDTh[:WIDTh]`.

**Suffix:**

<m>
1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**

<WidthDelta>
Range: 0 to 432
Increment: 500E-12
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Additional settings: Timing" on page 215

### 24.9.13 High-speed serial pattern trigger (option R&S RTP-K140/K141)

This chaptes decribes the remote commands of serial pattern trigger option R&S RTP-K140/K141.

The serial pattern trigger is only available for the A-event (Suffix = 1).

The following provides examples on how to set up the serial pattern trigger :

**Example: Basic settings of the serial pattern trigger**

Sets a clock signal:

```
TRIGger1:TYPE SERPattern
TRIGger1:SPATtern:CSOurce:EDGE POSitive
TRIGger1:SPATtern:CSOurce:LEVel 0
TRIGger1:LEVel 0
```

Set a CDR clock signal:

```
TRIGger1:TYPE SERPattern
CDR:HARDware:SERStandard CUSTOM
CDR:HARDware:UIOFfset 0.5
TRIGger1:LEVel 0
CDR:HARDware:BREStimate
```

**Example: Setting up a dual bit serial pattern trigger**

Set the pattern type to dual pattern and sets the length and the value for bit patter A and B:

```
TRIGger1:SERPattern:PTMDe:DBP
TRIGger1:SERPattern:BITPattern:PTALength 1
TRIGger1:SERPattern:BITPattern:APATtern 1
TRIGger1:SERPattern:BITPattern:PTBLength 1
TRIGger1:SERPattern:BITPattern:BPATtern 0
```

**Example: Setting up a complex word serial pattern trigger**

```
TRIGger1:SERPattern:PTMDe:CBP
// Sets the alignment to pattern
TRIGger1:SERPattern:COMPlex:SOFState PATTern
// Sets the alignment to pattern
TRIGger1:SERPattern:COMPlex:SOFLength 8
TRIGger1:SERPattern:COMPlex:SOFPattern A5
TRIGger1:SERPattern:COMPlex:CLOGic AND
TRIGger1:SERPattern:COMPlex:BORDer MSB
// Sets the length, bit pattern, logic and bit order
TRIGger1:SERPattern:COMPlex:PATTern1:ENABle ON
//Enables pattern condition 1
TRIGger1:SERPattern:COMPlex:PATTern1:BOFFset 4
TRIGger1:SERPattern:COMPlex:PATTern1:PATLength 8
TRIGger1:SERPattern:COMPlex:PATTern1:SRNGe 1
TRIGger1:SERPattern:COMPlex:PATTern1:PCONdition EQUal
TRIGger1:SERPattern:COMPlex:PATTern1:FRPTtern 37
// Sets the bit offset, pattern length, search range and pattern condition
```

### 24.9.13.1 Basic serial pattern trigger settings

TRIGger<m>:SERPattern:PTMDe ..... 1328
TRIGger<m>:LEVel<n>[:VALue] ..... 1329
TRIGger<m>:SPATtern:CSOurce[:VALue] ..... 1330
TRIGger<m>:SPATtern:CSOurce:EDGE ..... 1330
TRIGger<m>:SPATtern:CSOurce:LEVel ..... 1330
TRIGger<m>:SPATtern:REFSource ..... 1331
CDR:HARDware:SOURce ..... 1331
CDR:HARDware:SERStandard ..... 1331
CDR:HARDware:BREStimate ..... 1332
CDR:HARDware:BITRate ..... 1332
CDR:HARDware:ESLope ..... 1332
CDR:HARDware:UIOFfset ..... 1333

**TRIGger<m>:SERPattern:PTMDe** <PatternMode>

Selects the pattern type for the serial pattern trigger.

**Suffix:**
<m> 1..3

**Parameters:**
<PatternMode> SBP | DBP | CBP | PRBS | EBTB | I30B | I32B

**SBP**
Single bit pattern

**DBP**
Dual bit pattern

**CBP**
Complex word pattern

**PRBS**
PRBS

**EBTB**
8b/10b

**I32B**
128b/132b

*RST: SBP

**Usage:** Asynchronous command

**Manual operation:** See "Pattern mode" on page 247

---

**TRIGger<m>:LEVel<n>[:VALue]** <Level>

Sets the trigger level for the specified event and source.

If the trigger source is serial bus, the trigger level is set by the thresholds in the protocol configuration.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

<n> 1..31
Indicates the trigger source:
1...4 = channel 1 to 4
5 = external trigger input
6...11 = not available
12...27 = R&S RT-ZVC input channels
28...31 = differential signals Diff1, Diff2 and Common1, Common2

**Parameters:**
<Level> Voltage for the trigger level.

Range: Depends on vertical scale, channel offset and other settings. The trigger level must be within the current display range.
Increment: 1E-3
*RST: 0
Default unit: V

**Example:** `TRIG:LEV5 0.01`
Sets the trigger level for the external trigger signal to 10 mV.
`TRIG2:LEV3 0.2`
Sets the trigger level for the B-event and B-trigger source channel 3 to 200 mV.

**Usage:** Asynchronous command

**Manual operation:** See "[Level]" on page 36

---

**TRIGger<m>:SPATtern:CSOurce[:VALue]** <ClockSource>

Defines the source of the clock signal.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<ClockSource> CHAN1 | CHANNEL1 | CHAN2 | CHANNEL2 | CHAN3 | CHANNEL3 | CHAN4 | CHANNEL4
CHAN1 = CHANnel1, CHAN2 = CHANnel2, CHAN3 = CHANnel3, CHAN4 = CHANnel4
Input channel of the clock signal
*RST: CHAN1

**Usage:** Asynchronous command

---

**TRIGger<m>:SPATtern:CSOurce:EDGE** <ClockEdge>

Together with the clock level (see `TRIGger<m>:SPATtern:CSOurce:LEVel` on page 1330), the clock edge defines the point in time when the state of the data signal is checked.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<ClockEdge> POSitive | NEGative | EITHer
See Chapter 24.4.3, "Slope parameter", on page 1194.
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Clock edge" on page 244

---

**TRIGger<m>:SPATtern:CSOurce:LEVel** <ClockLevel>

Defines the voltage level for the clock signal.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<ClockLevel> Range: -10 to 10
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Clock level " on page 245

---

**TRIGger<m>:SPATtern:REFSource** <ReferenceSource>

Selects if an explicit clock or a CDR signal is used as a reference signal.

**Suffix:**
<m> 1..3

**Parameters:**
<ReferenceSource> CLOCk | CDR
*RST: CDR

**Usage:** Asynchronous command

**Manual operation:** See "Clock / CDR" on page 243

---

**CDR:HARDware:SOURce** <AnalogSource>

Selects the channel signal that is used for clock recovery.

The source cannot be changed if the serial pattern trigger is selected in the trigger setup and the CDR is enabled. In this case, the instrument triggers on the recovered clock, trigger source and CDR source are the same.

**Parameters:**
<AnalogSource> C1 | C2 | C3 | C4
*RST: C1

**Usage:** Asynchronous command

---

**CDR:HARDware:SERStandard** <SerialStandard>

Sets the serial standard. Set `CUSTOM` to adjust the bit rate and the bandwidth manually.

**Parameters:**

<SerialStandard> CUSTOM | USB20 | USB31GEN1 | DISPPORT11 | DISPPORT12 | FIBCHAN1 | FIBCHAN2 | FIBCHAN4 | FIBCHAN8 | HDMI12 | HDMI14 | HDMI20 | PCIGEN1 | PCIGEN2 | SATAGEN1 | SATAGEN2 | SATAGEN3 | SASGEN1 | SASGEN2 | XAUI

*RST: CUSTOM

**Usage:** Asynchronous command

**Manual operation:** See "Serial standard" on page 245

---

**CDR:HARDware:BREStimate**

Estimates the bite rate once, for the current acquisition.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Estimate bit rate" on page 247

---

**CDR:HARDware:BITRate** <Bitrate>

Sets the frequency of the feed forward CDR. It corresponds to the data rate of the data stream from which the clock is to be recovered.

**Parameters:**

<Bitrate> Range: 200E+3 to 2.5E+9 for 10 GSa/s; range depends on sample rate
Increment: 10
*RST: 1E+9
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "Nominal bit rate" on page 246

---

**CDR:HARDware:ESLope** <Edge>

Selects the edges of the data stream that are used for the clock recovery.

- "Either": Both positive and negative edges are used
- "Positive / Negative": Only one edge direction is used. Use one of these settings if the other edge might deliver unreliable results.

**Parameters:**

<Edge> POSitive | NEGative | EITHer

*RST: EITHer

**Usage:** Asynchronous command

**CDR:HARDware:UIOFfset** <UntIntvlOffs>

Defines the position of the last bit of the bit pattern to be triggered on, relative to the trigger time. The unit interval offset is a number between 0 and 1:

- Value 0: start/"left edge" of the last bit
- Value 0.5: middle of the last bit
- Value 1: end/"right edge" of the last bit

**Parameters:**

<UntIntvlOffs>
Range: 0 to 1
Increment: 0.01
*RST: 0.5
Default unit: UI

**Usage:** Asynchronous command

**Manual operation:** See "Sampling time" on page 246

### 24.9.13.2 Single/dual bit pattern

TRIGger<m>:SERPattern:BITPattern:APATtern ..... 1333
TRIGger<m>:SERPattern:BITPattern:BPATtern ..... 1333
TRIGger<m>:SERPattern:BITPattern:PTALength ..... 1334
TRIGger<m>:SERPattern:BITPattern:PTBLength ..... 1334

**TRIGger<m>:SERPattern:BITPattern:APATtern** <BPatternA>

Defines the bit pattern/ bit pattern A, if `TRIGger<m>:SERPattern:PTMDe` is set to `SBP` or `DBP`.

**Suffix:**

<m> 1..3

**Parameters:**

<BPatternA> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Usage:** Asynchronous command

**Manual operation:** See "Bit pattern, Bit pattern A, Bit pattern B" on page 248

**TRIGger<m>:SERPattern:BITPattern:BPATtern** <BPatternB>

Defines the bit pattern B, if `TRIGger<m>:SERPattern:PTMDe` is set to `DBP`.

**Suffix:**

<m> 1..3

**Parameters:**

<BPatternB> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Usage:** Asynchronous command

**Manual operation:** See "Bit pattern, Bit pattern A, Bit pattern B" on page 248

---

**TRIGger<m>:SERPattern:BITPattern:PTALength** <PatternALen>

Sets the length of the bit pattern A.

**Suffix:**

<m> 1..3

**Parameters:**

<PatternALen>
Range: 1 to 160
Increment: 1
*RST: 8

**Manual operation:** See "Bit pattern length, Bit pattern length A, Bit pattern length B" on page 248

---

**TRIGger<m>:SERPattern:BITPattern:PTBLength** <PatternBLen>

Sets the length of the bit pattern B.

**Suffix:**

<m> 1..3

**Parameters:**

<PatternBLen>
Range: 1 to 160
Increment: 1
*RST: 8

**Usage:** Asynchronous command

**Manual operation:** See "Bit pattern length, Bit pattern length A, Bit pattern length B" on page 248

### 24.9.13.3 Complex word pattern

TRIGger<m>:SERPattern:COMPlex:BORDer ... 1335
TRIGger<m>:SERPattern:COMPlex:CLOGic ... 1335
TRIGger<m>:SERPattern:COMPlex:IDLRange ... 1335
TRIGger<m>:SERPattern:COMPlex:SOFState ... 1336
TRIGger<m>:SERPattern:COMPlex:SOFTimeout ... 1336
TRIGger<m>:SERPattern:COMPlex:SOFLength ... 1336
TRIGger<m>:SERPattern:COMPlex:SOFPattern ... 1337
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:BOFFset ... 1337
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:ENABle ... 1337
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:FRPTtern ... 1337

TRIGger<m>:SERPattern:COMPlex:PATTern<n>:PATLength ..... 1338
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:PCONdition ..... 1338
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:SRNGe ..... 1339
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:TOPTtern ..... 1339

**TRIGger<m>:SERPattern:COMPlex:BORDer** <BitOrder>

Sets the most significant bit (MSB) or least significant bit (LSB) to define the data bit order.

**Suffix:**
<m> 1..3

**Parameters:**
<BitOrder> LSB | MSB

*RST: MSB

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 250

**TRIGger<m>:SERPattern:COMPlex:CLOGic** <CondLogic>

Selects the logic for the condition, if more than one pattern conditions are defined.

**Suffix:**
<m> 1..3

**Parameters:**
<CondLogic> AND | OR

**AND**
All the defined pattern conditions have to be met.

**OR**
At least one of the defined pattern conditions have to be met.

*RST: AND

**Usage:** Asynchronous command

**Manual operation:** See "Logic" on page 250

**TRIGger<m>:SERPattern:COMPlex:IDLRange** <IdleRange>

Selects if the timeout should be within the range or out of range.

**Suffix:**
<m> 1..3

**Parameters:**
<IdleRange> INR | OOR

INR: In range
OOR: Out of range

*RST: OOR

**Usage:** Asynchronous command

**Manual operation:** See "Idle range" on page 250

---

**TRIGger<m>:SERPattern:COMPlex:SOFState** <SOFState>

Selects if the alignment for the complex word pattern triggering is done on a pattern value or on a defined timeout.

**Suffix:**
<m> 1..3

**Parameters:**
<SOFState> PATTERN | TIMEOUT
*RST: PATTERN

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 249

---

**TRIGger<m>:SERPattern:COMPlex:SOFTimeout** <SOFTimeout>

Sests the timeout, a duration that defines an idle mark, if `TRIGger<m>:SERPattern:COMPlex:SOFState` is set to `TIMEout`.

**Suffix:**
<m> 1..3

**Parameters:**
<SOFTimeout> Range: 40E-9 to 150
Increment: 10E-6
*RST: 5E-6
Default unit: s

**Manual operation:** See "Timeout" on page 250

---

**TRIGger<m>:SERPattern:COMPlex:SOFLength** <SOFPatternLen>

Sets the length of the pattern for the trigger alignment condition.

**Suffix:**
<m> 1..3

**Parameters:**
<SOFPatternLen> Range: 1 to 132
Increment: 1
*RST: 8

**Usage:** Asynchronous command

**Manual operation:** See "Length" on page 250

**TRIGger<m>:SERPattern:COMPlex:SOFPattern** <SOFPattern>

Sets the bit pattern value for the trigger alignment condition, if `TRIGger<m>:SERPattern:COMPlex:SOFState` is set to `PATTern`.

**Suffix:**
<m> 1..3

**Parameters:**
<SOFPattern>

**Usage:** Asynchronous command

**Manual operation:** See "Bit Pattern" on page 250

**TRIGger<m>:SERPattern:COMPlex:PATTern<n>:BOFFset** <BitOffset>

Sets the number of arbitrary bits between the first bit after the frame id and the first bit of the word pattern.

**Suffix:**
<m> 1..3

<n> 1..4
Selects the complex pattern condition.

**Parameters:**
<BitOffset> Range: 0 to 255
Increment: 1
*RST: 0

**Manual operation:** See "Bit offset" on page 250

**TRIGger<m>:SERPattern:COMPlex:PATTern<n>:ENABle** <Enable>

Enables a new pattern condition.

**Suffix:**
<m> 1..3

<n> 1..4
Selects the complex pattern condition.

**Parameters:**
<Enable> ON | OFF
*RST: OFF

**Manual operation:** See "Select to add condition" on page 251

**TRIGger<m>:SERPattern:COMPlex:PATTern<n>:FRPTtern** <PatternFrom>

Sets the pattern or a start value of a pattern range.

**Suffix:**
<m> 1..3

<n> 1..4
Selects the complex pattern condition.

**Parameters:**
<PatternFrom>

**Manual operation:** See "Pattern" on page 251

---

**TRIGger<m>:SERPattern:COMPlex:PATTern<n>:PATLength** <PatternLen>

Sets the word pattern length in bits. The maximal bit length of all selected pattern conditions is 160 bits.

**Suffix:**
<m> 1..3

<n> 1..4
Selects the complex pattern condition.

**Parameters:**
<PatternLen> Range: 1 to 160
Increment: 1
*RST: 8

**Manual operation:** See "Pattern length" on page 250

---

**TRIGger<m>:SERPattern:COMPlex:PATTern<n>:PCONdition** <Operator>

Sets the operator to define the exact position or range where the instrument looks for the specified pattern.

**Suffix:**
<m> 1..3

<n> 1..4
Selects the complex pattern condition.

**Parameters:**
<Operator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These operators require one identifier pattern to be set with `TRIGger<m>:SERPattern:COMPlex:PATTern<n>:FRPTtern`

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `TRIGger<m>:SERPattern:COMPlex:PATTern<n>:FRPTtern` and `TRIGger<m>:SERPattern:COMPlex:PATTern<n>:TOPTtern`.

*RST: EQUal

**Manual operation:** See "Pattern condition" on page 251

**TRIGger<m>:SERPattern:COMPlex:PATTern<n>:SRNGe** <SearchRange>

Sets the search range of the word pattern.

**Suffix:**
<m> 1..3

<n> 1..4
Selects the complex pattern condition.

**Parameters:**
<SearchRange> Range: 1 to 65535
Increment: 1
*RST: 1

**Manual operation:** See "Search range" on page 251

**TRIGger<m>:SERPattern:COMPlex:PATTern<n>:TOPTtern** <PatternTo>

Sets the the end value of a pattern range if `TRIGger<m>:SERPattern:COMPlex:PATTern<n>:PCONdition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

<n> 1..4
Selects the complex pattern condition.

**Parameters:**
<PatternTo>

**Manual operation:** See "Pattern" on page 251

### 24.9.13.4 PRBS error

TRIGger<m>:SERPattern:PRBS:ERATe....1339
TRIGger<m>:SERPattern:PRBS:NBRTe....1340
TRIGger<m>:SERPattern:PRBS:PSEQuence....1340

**TRIGger<m>:SERPattern:PRBS:ERATe** <PRBSBtErrRate>

Sets the bit error rate for the PRBS sequence.

**Suffix:**
<m> 1..3

**Parameters:**
<PRBSBtErrRate> Range: 0 to 100
Increment: 100E-6
*RST: 1
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "PRBS Bit error rate" on page 252

**TRIGger<m>:SERPattern:PRBS:NBRTe** <PRBSNomBitRate>

Sets the nominal bit rate for the PRBS sequence. It corresponds to the data rate of the data stream from which the clock is to be recovered.

**Suffix:**
<m> 1..3

**Parameters:**
<PRBSNomBitRate> Range: 200E+3 to 2.5E+9
Increment: 10
*RST: 1E+9
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "Nominal bit rate" on page 252

**TRIGger<m>:SERPattern:PRBS:PSEQuence** <PRBSSequence>

Selects a predefined pseudorandom binary sequence.

**Suffix:**
<m> 1..3

**Parameters:**
<PRBSSequence> PRBS7 | PRBS9 | PRBS11 | PRBS15 | PRBS16 | PRBS17 | PRBS20 | PRBS23 | PRBS29 | PRBS31
*RST: PRBS7

**Usage:** Asynchronous command

**Manual operation:** See "PRBS Sequence" on page 251

### 24.9.13.5 8b/10b

TRIGger<m>:SERPattern:EBTB:ALIGnment....1340
TRIGger<m>:SERPattern:EBTB:APPend....1341
TRIGger<m>:SERPattern:EBTB:CLEar....1341
TRIGger<m>:SERPattern:EBTB:COUNt?....1341
TRIGger<m>:SERPattern:EBTB:ERRor<n>:DISParity....1342
TRIGger<m>:SERPattern:EBTB:ERRor<n>:SYMBol....1342
TRIGger<m>:SERPattern:EBTB:FID....1342
TRIGger<m>:SERPattern:EBTB:SYMBols<n>:DMINus?....1342
TRIGger<m>:SERPattern:EBTB:SYMBols<n>:DPLus?....1343
TRIGger<m>:SERPattern:EBTB:SYMBols<n>:EBIT....1343
TRIGger<m>:SERPattern:EBTB:SYMBols<n>:KDSYmbol....1343
TRIGger<m>:SERPattern:EBTB:SCRMble....1344

**TRIGger<m>:SERPattern:EBTB:ALIGnment** <EBTBAlignment>

Selects one of the comma control symbols used for synchronization.

**Suffix:**
<m> 1..3

**Parameters:**
<EBTBAlignment> K281 | K285 | K287
*RST: K285

**Usage:** Asynchronous command

**Manual operation:** See "Alignment" on page 253

---

**TRIGger<m>:SERPattern:EBTB:APPend**

Creates an empty row and adds it to the end of the active symbol description in the 8b/10b table.

**Suffix:**
<m> 1..3

**Usage:** Event
Asynchronous command

**Manual operation:** See "8b/10b table" on page 253

---

**TRIGger<m>:SERPattern:EBTB:CLEar**

Removes all entries from the 8b/10b table.

**Suffix:**
<m> 1..3

**Usage:** Event
Asynchronous command

**Manual operation:** See "8b/10b table" on page 253

---

**TRIGger<m>:SERPattern:EBTB:COUNt?**

Selects the number of 8b/10b symbols.

**Suffix:**
<m> 1..3

**Return values:**
<EBTBSymbolCount> Range: 0 to 16
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "8b/10b table" on page 253

**TRIGger<m>:SERPattern:EBTB:ERRor<n>:DISParity** <EBTBDisparityErr>

Enables triggering on disparity errors.

**Suffix:**
<m> 1..3
<n> *

**Parameters:**
<EBTBDisparityErr> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Disparity error" on page 253

**TRIGger<m>:SERPattern:EBTB:ERRor<n>:SYMBol** <EBTBSymbolError>

Enables triggering on symbol errors.

**Suffix:**
<m> 1..3
<n> *

**Parameters:**
<EBTBSymbolError> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Symbol error" on page 253

**TRIGger<m>:SERPattern:EBTB:FID** <EBTBFrameId>

Selects the frame ID.

**Suffix:**
<m> 1..3

**Parameters:**
<EBTBFrameId> K280
*RST: K280

**Usage:** Asynchronous command

**TRIGger<m>:SERPattern:EBTB:SYMBols<n>:DMINus?**

Returns the value for the negative running disparity RD-.

**Suffix:**
<m> 1..3

<n> 1...16
Sets the index of the symbol.

**Return values:**
<DMinus>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "8b/10b table" on page 253

---

**TRIGger<m>:SERPattern:EBTB:SYMBols<n>:DPLus?**

Returns the value for the positive running disparity RD+.

**Suffix:**
<m> 1..3

<n> 1...16
Sets the index of the symbol.

**Return values:**
<DPlus>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "8b/10b table" on page 253

---

**TRIGger<m>:SERPattern:EBTB:SYMBols<n>:EBIT** <EightBit>

Sets the 8b bit value.

**Suffix:**
<m> 1..3

<n> 1...16
Sets the index of the symbol.

**Parameters:**
<EightBit>

**Usage:** Asynchronous command

**Manual operation:** See "8b/10b table" on page 253

---

**TRIGger<m>:SERPattern:EBTB:SYMBols<n>:KDSYmbol** <KDSymbol>

Sets the data characters (Dx.y) or the control characters (Kx.y)

**Suffix:**
<m> 1..3

<n> 1...16
Sets the index of the symbol.

**Parameters:**
<KDSymbol>

**Usage:** Asynchronous command

**Manual operation:** See "8b/10b table" on page 253

---

**TRIGger<m>:SERPattern:EBTB:SCRMble** <EBTBScrambleMd>

Selects if the data is scrambled or unscrambled.

**Suffix:**
<m> 1..3

**Parameters:**
<EBTBScrambleMd> SCRM | NSCR

`SCRM`: scrambled
`NSCR`: unscrambled

*RST: NSCR

**Usage:** Asynchronous command

**Manual operation:** See "Scramble" on page 253

### 24.9.13.6 128b/132b

TRIGger<m>:SERPattern:I32B:APATtern ..... 1344
TRIGger<m>:SERPattern:I32B:PATTern ..... 1344
TRIGger<m>:SERPattern:I32B:SKAFter ..... 1345
TRIGger<m>:SERPattern:I32B:SSTP ..... 1345
TRIGger<m>:SERPattern:I32B:SSTR ..... 1345
TRIGger<m>:SERPattern:I32B:SYMType ..... 1345

---

**TRIGger<m>:SERPattern:I32B:APATtern** <Alignment>

Sets the alignment pattern.

**Suffix:**
<m> 1..3

**Parameters:**
<Alignment>

**Usage:** Asynchronous command

**Manual operation:** See "Alignment / Sync" on page 254

---

**TRIGger<m>:SERPattern:I32B:PATTern** <Pattern>

Defines the control block or data block pattern in binary or hex format.

**Suffix:**
<m> 1..3

**Parameters:**
<Pattern>

**Usage:** Asynchronous command

**Manual operation:** See "Pattern" on page 255

---

**TRIGger<m>:SERPattern:I32B:SKAFter** <SkipAfter>

Sets the skip bytes symbols send after a skip end symbol .

**Suffix:**
<m> 1..3

**Parameters:**
<SkipAfter>
Range: 1 to 32
Increment: 1
*RST: 3

**Usage:** Asynchronous command

**Manual operation:** See "Skip bytes" on page 254

---

**TRIGger<m>:SERPattern:I32B:SSTP** <SkipStop>

Sets the skip end symbol.

**Suffix:**
<m> 1..3

**Parameters:**
<SkipStop>

**Usage:** Asynchronous command

**Manual operation:** See "Skip end" on page 254

---

**TRIGger<m>:SERPattern:I32B:SSTR** <SkipStart>

Sets the start skip value.

**Suffix:**
<m> 1..3

**Parameters:**
<SkipStart>

**Usage:** Asynchronous command

**Manual operation:** See "Skip start" on page 254

---

**TRIGger<m>:SERPattern:I32B:SYMType** <SymbolType>

Selects the type of pattern, control block or the data block.

**Suffix:**
<m> 1..3

**Parameters:**
<SymbolType> CTL | DAT
*RST: DAT

**Usage:** Asynchronous command

**Manual operation:** See "Block type" on page 254

### 24.9.13.7 128b/130b

TRIGger<m>:SERPattern:I30B:APATtern ..... 1346
TRIGger<m>:SERPattern:I30B:PATTern ..... 1346
TRIGger<m>:SERPattern:I30B:SKAFter ..... 1346
TRIGger<m>:SERPattern:I30B:SSTP ..... 1347
TRIGger<m>:SERPattern:I30B:SSTR ..... 1347
TRIGger<m>:SERPattern:I30B:SYMType ..... 1347

---

**TRIGger<m>:SERPattern:I30B:APATtern** <Alignment>

Sets the alignment pattern.

**Suffix:**
<m> 1..3

**Parameters:**
<Alignment>

**Usage:** Asynchronous command

**Manual operation:** See "Alignment / Sync" on page 254

---

**TRIGger<m>:SERPattern:I30B:PATTern** <Pattern>

Defines the OS or data block pattern in binary or hex format.

**Suffix:**
<m> 1..3

**Parameters:**
<Pattern>

**Usage:** Asynchronous command

**Manual operation:** See "Pattern" on page 255

---

**TRIGger<m>:SERPattern:I30B:SKAFter** <SkipAfter>

Sets the skip bytes symbols send after a skip end symbol .

**Suffix:**
<m> 1..3

**Parameters:**
<SkipAfter> Range: 1 to 32
Increment: 1
*RST: 3

**Usage:** Asynchronous command

**Manual operation:** See "Skip bytes" on page 254

---

**TRIGger<m>:SERPattern:I30B:SSTP** <SkipStop>

Sets the skip end symbol.

**Suffix:**
<m> 1..3

**Parameters:**
<SkipStop>

**Usage:** Asynchronous command

**Manual operation:** See "Skip end" on page 254

---

**TRIGger<m>:SERPattern:I30B:SSTR** <SkipStart>

Sets the start skip value.

**Suffix:**
<m> 1..3

**Parameters:**
<SkipStart>

**Usage:** Asynchronous command

**Manual operation:** See "Skip start" on page 254

---

**TRIGger<m>:SERPattern:I30B:SYMType** <SymbolType>

Selects the type of pattern, control block or the data block.

**Suffix:**
<m> 1..3

**Parameters:**
<SymbolType> OS | DAT
*RST: DAT

**Usage:** Asynchronous command

**Manual operation:** See "Block type" on page 254

### 24.9.14 CDR trigger

The CDR trigger is only available for the A-event (Suffix = 1).

**TRIGger<m>:CDR:EYE:ALGorithm** <SerStandardType>

Sets the software algorithm that is used for clock data recovery.

**Suffix:**
<m> 1..3

**Parameters:**
<SerStandardType> CFRequency | FF

**CFRequency**
Constant frequency. CDR uses the nominal bit rate to generate the clock signal. The method assumes that the frequency of the signal is constant during the complete acquisition.

**FF**
Feed forward algorithm, a mathematical optimization method. Feed forward is available if none of the listed serial standards is used (`TRIGger<m>:CDR:EYE:SERStandard` is set to `CUSTom`).

*RST: FF

**Usage:** Asynchronous command

**Manual operation:** See "Algorithm" on page 224

**TRIGger<m>:CDR:EYE:BITRate** <Bitrate>

Sets the frequency of the feed forward CDR. It corresponds to the data rate of the data stream from which the clock is to be recovered.

**Suffix:**
<m> 1..3

**Parameters:**
<Bitrate>
Range: 200E+3 to 2.5E+9
Increment: 10
*RST: 1E+9
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "Bit rate estimation" on page 224

**TRIGger<m>:CDR:EYE:BREStimate**

Estimates the bite rate once, for the current acquisition.

**Suffix:**
<m> 1..3

**Usage:** Event
Asynchronous command

**Manual operation:** See "Bit rate estimation" on page 224

---

**TRIGger<m>:CDR:EYE:BREStimate:CONTinuous** <ContBitrateestim>

Enables ongoing correction of the bit rate for the hardware CDR.

**Suffix:**
<m> 1..3

**Parameters:**
<ContBitrateestim> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Bit rate estimation" on page 224

---

**TRIGger<m>:CDR:EYE:BWIDth** <Bandwidth>

Sets the PLL bandwidth. It defines the part of the spectrum that the PLL can follow during synchronization. The PLL bandwidth is usually defined by the transmission standard.

**Suffix:**
<m> 1..3

**Parameters:**
<Bandwidth> Range: 20E+3 to 10E+6
Increment: 10
*RST: 599.88E+3
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Algorithm settings" on page 225

---

**TRIGger<m>:CDR:EYE:LEVel** <Level>

Sets the edge detection threshold for the hardware CDR.

**Suffix:**
<m> 1..3

**Parameters:**
<Level> Range: -100E+24 to 100E+24
Increment: 0.1
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Data level" on page 224

**TRIGger<m>:CDR:EYE:RELBwidth** <RelBw>

Sets the relative bandwidth, that is the ratio of the nominal bit rate to the PLL bandwidth.

**Suffix:**
<m> 1..3

**Parameters:**
<RelBw> Range: 500 to 3000
Increment: 1
*RST: 1667

**Usage:** Asynchronous command

**Manual operation:** See "Algorithm settings" on page 225

**TRIGger<m>:CDR:EYE:SAMTime** <SamplingTime>

Sets a sampling time for the clock signal, an offset for the clock edge in relation to the bit start. The clock edge sets the beginning of the unit interval. The sampling time is a number between 0 and 1. Value 0 sets the clock edge to the beginning of the bit period; value 0.5 sets the clock edge to the middle of the bit period.

**Suffix:**
<m> 1..3

**Parameters:**
<SamplingTime> Range: 0 to 1
Increment: 0.01
*RST: 0.5
Default unit: UI

**Usage:** Asynchronous command

**Manual operation:** See "Sampling time" on page 224

**TRIGger<m>:CDR:EYE:SERStandard** <SerStandardType>

Sets the serial standard. Set `CUSTOM` to adjust the bit rate and the bandwidth manually.

**Suffix:**
<m> 1..3

**Parameters:**

<SerStandardType> CUSTOM | USB20 | USB31GEN1 | USB31GEN2 | DISPPORT11 | DISPPORT12 | FIBCHAN1 | FIBCHAN2 | FIBCHAN4 | FIBCHAN8 | HDMI12 | HDMI14 | HDMI20 | PCIGEN1 | PCIGEN2 | PCIGEN3 | PCIGEN4 | SATAGEN1500 | SATAGEN11667 | SATAGEN1 | SATAGEN2500 | SATAGEN21667 | SATAGEN2 | SATAGEN3500 | SATAGEN31667 | SATAGEN3 | SASGEN1 | SASGEN2 | SASGEN3 | TB10G | XAUI | CUSTOM | USB20 | USB31GEN1 | DISPPORT11 | DISPPORT12 | FIBCHAN1 | FIBCHAN2 | FIBCHAN4 | FIBCHAN8 | HDMI12 | HDMI14 | HDMI20 | PCIGEN1 | PCIGEN2 | SATAGEN1 | SATAGEN2 | SATAGEN3 | SASGEN1 | SASGEN2 | XAUI

*RST: CUSTOM

**Usage:** Asynchronous command

**Manual operation:** See "Serial standard" on page 222

---

**TRIGger<m>:CDR:EYE:SOURce?**

Selects the channel signal that is used for clock recovery.

The source cannot be changed if the serial pattern trigger is selected in the trigger setup and the CDR is enabled. In this case, the instrument triggers on the recovered clock, trigger source and CDR source are the same.

**Suffix:**

<m> 1..3

**Return values:**

<SourceType> C1 | C2 | C3 | C4

*RST: C1

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Source" on page 222

### 24.9.15 TV/Video trigger

The TV or video trigger requires a channel input as trigger source (`TRIGger<m>:SOURce[:SELect]`). It is only available for the A-event (Suffix = 1).

Make sure to set the trigger level - the threshold of the sync pulse - with `TRIGger<m>:LEVel<n>[:VALue]`.

TRIGger<m>:TV:STANdard....1352
TRIGger<m>:TV:MODE....1352
TRIGger<m>:TV:POLarity....1353
TRIGger<m>:TV:LINE....1353
TRIGger<m>:TV:LFIeld....1354

TRIGger<m>:TV:CUSTom:SCANmode....1355
TRIGger<m>:TV:CUSTom:LDURation....1355
TRIGger<m>:TV:CUSTom:STYPe....1355
TRIGger<m>:TV:CUSTom:SDURation....1356

**TRIGger<m>:TV:STANdard** <Standard>

Sets the TV standard.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<Standard> CUSTom | PAL | PALM | NTSC | SECam | P480L60HZ | P720L30HZ | P720L50HZ | P720L60HZ | I1080L50HZ | I1080L60HZ | P1080L24HZ | P1080L24HZSF | P1080L25HZ | P1080L30HZ | P1080L50HZ | P1080L60HZ

**CUSTom**
User-defined signal. Configure the signal using:
`TRIGger<m>:TV:CUSTom:SCANmode`
`TRIGger<m>:TV:CUSTom:STYPe`
`TRIGger<m>:TV:CUSTom:LDURation`
`TRIGger<m>:TV:CUSTom:SDURation`

**PAL | PALM | NTSC | SECam**
SDTV standards. PALM = PAL-M

**PxxxxLyyHZ**
HDTV standards using progressive scanning (P). xxxx indicates the number of active lines, yy is the frame rate.

**IxxxxLxxHZ**
HDTV standards using interlaced scanning (I). xxxx indicates the number of active lines, yy is the field rate.

**P1080L24HZSF**
1080p/24sF is a HDTV standard using progressive segmented frame scanning.

*RST: PAL

**Usage:** Asynchronous command

**Manual operation:** See "Standard" on page 217

**TRIGger<m>:TV:MODE** <Mode>

Selects the lines or fields on which the instrument can trigger. Available modes depend on the scanning system.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<Mode> ALL | ODD | EVEN | ALINe | LINE

**ALL**
All fields, triggers on the frame start (progressive scanning) or field start (interlaced and progressive segmented frame scanning)

**ODD | EVEN**
Only available for interlaced scanning and progressive segmented frame scanning. Triggers on the field start of the odd or even field.

**ALINe**
All lines, triggers on all line starts.

**LINE**
Triggers on a specified line. To set the line number, use `TRIGger<m>:TV:LINE`. For NTSC signals, set also the field with `TRIGger<m>:TV:LFIeld`.

*RST: ALL

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 217

---

**TRIGger<m>:TV:POLarity** <Polarity>

Sets the polarity of the *signal*. Note that the sync pulse has the opposite polarity, for example, a positive signal has a negative sync pulse.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<Polarity> POSitive | NEGative

*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Signal polarity" on page 218

---

**TRIGger<m>:TV:LINE** <LineNumber>

Specifies the line number to trigger on. The command is relevant if `TRIGger<m>:TV:MODE` is set to `LINe`.

Usually the lines of the frame are counted beginning from the frame start. For NTSC signals, the lines are counted per field, not per frame. For these signals, set also the field with `TRIGger<m>:TV:LFIeld`.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<LineNumber> Range: Depends on the standard, see table below
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Line #" on page 218

| Standard | Minimum value | Maximum value |
|---|---|---|
| PAL | 1 | 625 |
| PAL-M | 1 | 525 |
| NTSC | 1 | 263 in odd field<br>262 in even field |
| SECAM | 1 | 625 |
| 480p/60 (`P480L60HZ`) | 1 | 525 |
| 720p/30 (`P720L30HZ`)<br>720p/50 (`P720L50HZ`)<br>720p/60 (`P720L60HZ`) | 1 | 750 |
| 1080i/50 (`I1080L50HZ`)<br>1080i/60 (`I1080L60HZ`)<br>1080p/24 (`P1080L24HZ`)<br>1080p/24sF (`P1080L24HZSF`)<br>1080p/25 (`P1080L25HZ`)<br>1080p/30 (`P1080L30HZ`)<br>1080p/50 (`P1080L50HZ`)<br>1080p/60 (`P1080L60HZ`) | 1 | 1125 |

---

**TRIGger<m>:TV:LFIeld** <LineField>

The commands is only relevant for NTSC signals and sets the field in which the line number is counted.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<LineField> FIELD1 | FIELD2
FIELD1 = odd field
FIELD2 = even field
*RST: FIELD1

**Usage:** Asynchronous command

**Manual operation:** See "Line #" on page 218

---

**TRIGger<m>:TV:CUSTom:SCANmode** <ScanMode>

Sets the scanning system. Only relevant if `TRIGger<m>:TV:STANdard` is set to `CUSTom`.

See also: "Scan" on page 220.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<ScanMode> INTerlaced | PROGressive | SEGMented

SEGMented: Progressive segmented frame uses progressive scanning to capture the frame, and interlaced scanning for transmission and display.

*RST: INTerlaced

**Usage:** Asynchronous command

**Manual operation:** See "Scan" on page 220

---

**TRIGger<m>:TV:CUSTom:LDURation** <LinePeriod>

Sets the duration of a line, the time between two successive sync pulses. Only relevant if `TRIGger<m>:TV:STANdard` is set to `CUSTom`.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<LinePeriod> Range: 1E-6 to 500E-6
Increment: 100E-9
*RST: 64E-6
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Line period" on page 219

---

**TRIGger<m>:TV:CUSTom:STYPe** <SyncPulseType>

Sets the type of the sync pulse. Only relevant if `TRIGger<m>:TV:STANdard` is set to `CUSTom`.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<SyncPulseType> BILevel | TRILevel

**BILevel**
Bi-level sync pulse, usually used in SDTV signals

**TRILevel**
Tri-level sync pulse, used in HDTV signals

*RST: BILevel

**Usage:** Asynchronous command

**Manual operation:** See "Pulse type" on page 219

---

**TRIGger<m>:TV:CUSTom:SDURation** <SyncPulseDurat>

Sets the width of the sync pulse. Only relevant if `TRIGger<m>:TV:STANdard` is set to `CUSTom`.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<SyncPulseDurat> Range: 100E-9 to 100E-6
Increment: 100E-9
*RST: 4.7E-6
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Pulse width" on page 219

### 24.9.16 Line trigger

To select the line trigger, set `TRIGger<m>:SOURce[:SELect]` to `LINE`.

---

**TRIGger<m>:POWerline:SLOPe** <Slope>

Selects the rising or falling edges of the AC power input for the trigger condition.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<Slope> POSitive | NEGative
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 220

### 24.9.17 Trigger qualification

The A-event can have its own trigger qualification. Qualification is not available for B- and R-events (event suffix m = 2 | 3) and some trigger types.

A suffix defines the trigger type to which the qualification belongs.

***Table 24-3: Trigger type suffixes***

| Suffix | Trigger type |
|---|---|
| 1 | EDGE |
| 2 | GLITch |
| 3 | WIDTh |
| 4 | RUNT |
| 5 | WINDow |
| 6 | TIMeout |
| 7 | INTerval |
| 8 | Qualification is not supported (SLEWrate) |
| 9 | Qualification is not supported (SETHold) |
| 10 | STATe |
| 11 | PATTern |
| 12 | qualification is not supported (ANEDge, ext. trigger input) |
| 13 | Currently not used |
| 14 | Currently not used |
| 15 | Qualification is not supported (SERPattern) |
| 16 | Qualification is not supported (NFC) |
| 17 | Qualification is not supported (TV) |
| 18 | Qualification is not supported (CDR) |

TRIGger<m>:QUALify<n>:STATe....1358
TRIGger<m>:QUALify<n>:A[:ENABle]....1358
TRIGger<m>:QUALify<n>:B[:ENABle]....1358
TRIGger<m>:QUALify<n>:C[:ENABle]....1358
TRIGger<m>:QUALify<n>:D[:ENABle]....1358
TRIGger<m>:QUALify<n>:A:LOGic....1359
TRIGger<m>:QUALify<n>:B:LOGic....1359
TRIGger<m>:QUALify<n>:C:LOGic....1359
TRIGger<m>:QUALify<n>:D:LOGic....1359

TRIGger<m>:QUALify<n>:AB:LOGic....1359
TRIGger<m>:QUALify<n>:CD:LOGic....1359
TRIGger<m>:QUALify<n>:ABCD:LOGic....1359

**TRIGger<m>:QUALify<n>:STATe** <AddTrigLogi>

Enables the use of the qualification definition for the selected trigger event.

**Suffix:**

<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> 1..18
Defines the trigger type, see Table 24-3.

**Parameters:**

<AddTrigLogi> ON | OFF

**ON**
The qualification expression is considered for the trigger event.

**OFF**
The qualification expression is ignored for the trigger event.

*RST: OFF

**Usage:** Asynchronous command

**TRIGger<m>:QUALify<n>:A[:ENABle]** <State>
**TRIGger<m>:QUALify<n>:B[:ENABle]** <State>
**TRIGger<m>:QUALify<n>:C[:ENABle]** <State>
**TRIGger<m>:QUALify<n>:D[:ENABle]** <State>

The command is relevant for qualification, for pattern trigger and for state trigger.

Select the channels to be considered:

- A[:ENABle]: C1
- B[:ENABle]: C2
- C[:ENABle]: C3
- D[:ENABle]: C4

The trigger source cannot be enabled.

**Suffix:**

<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> 1..18
Trigger type, see Table 24-3.

**Parameters:**

<State> ON | OFF

**ON**
The qualification expression is considered.

**OFF**
The qualification expression is ignored.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Pattern" on page 214

---

**TRIGger<m>:QUALify<n>:A:LOGic** <Operator>
**TRIGger<m>:QUALify<n>:B:LOGic** <Operator>
**TRIGger<m>:QUALify<n>:C:LOGic** <Operator>
**TRIGger<m>:QUALify<n>:D:LOGic** <Operator>

Defines the logic for the indicated channel:

- A: C1
- B: C2
- C: C3
- D: C4

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> 1..18
Trigger type, see Table 24-3.

**Parameters:**
<Operator> DIRect | HIGH | NOT | LOW

**DIRect**
Input value remains unchanged

**NOT**
Input value is inverted

*RST: DIRect

**Usage:** Asynchronous command

**Manual operation:** See "Pattern" on page 214

---

**TRIGger<m>:QUALify<n>:AB:LOGic** <Operator>
**TRIGger<m>:QUALify<n>:CD:LOGic** <Operator>
**TRIGger<m>:QUALify<n>:ABCD:LOGic** <Operator>

Defines the logical combination of the indicated channels after evaluating the previous logical operations:

- AB: CH1 and CH2
- CD: CH3 and CH4
- ABCD: result of AB and CD

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> 1..18
Trigger type, see Table 24-3.

**Parameters:**
<Operator> AND | NAND | OR | NOR

**AND**
logical AND, conjunctive combination

**NAND**
logical NOT AND

**OR**
logical OR, disjunctive combination

**NOR**
logical NOT OR

*RST: AND

**Usage:** Asynchronous command

**Manual operation:** See "Pattern" on page 214

### 24.9.18 Holdoff

TRIGger<m>:HOLDoff:MODE ..... 1360
TRIGger<m>:HOLDoff:TIME ..... 1361
TRIGger<m>:HOLDoff:EVENts ..... 1362
TRIGger<m>:HOLDoff:MIN ..... 1362
TRIGger<m>:HOLDoff:MAX ..... 1363
TRIGger<m>:HOLDoff:AUTotime? ..... 1363
TRIGger<m>:HOLDoff:SCALing ..... 1364

**TRIGger<m>:HOLDoff:MODE** <Mode>

Selects the method to define the holdoff condition.

The trigger holdoff defines when the next trigger after the current will be recognized. Thus, it affects the next trigger to occur after the current one. Holdoff helps to obtain stable triggering when the oscilloscope is triggering on undesired events.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<Mode> TIME | EVENts | RANDom | AUTO | OFF

**TIME**
Defines the holdoff directly as a time period. The next trigger occurs only after the "Holdoff time" has passed (defined using `TRIGger<m>:HOLDoff:TIME`).

**EVENts**
Defines the holdoff as a number of trigger events. The next trigger occurs only when this number of events is reached. The number of triggers to be skipped is defined using `TRIGger<m>:HOLDoff:EVENts`.

**RANDom**
Defines the holdoff as a random time limited by `TRIGger<m>:HOLDoff:MIN` on page 1362 and `TRIGger<m>:HOLDoff:MAX` on page 1363. For each acquisition cycle, the instrument selects a new random holdoff time from the specified range.

**AUTO**
The holdoff time is calculated automatically based on the current horizontal scale.

**OFF**
No holdoff

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Holdoff mode" on page 227

---

**TRIGger<m>:HOLDoff:TIME** <Time>

Defines the holdoff time period. The next trigger occurs only after this time has passed. The setting is relevant if the holdoff mode is set to TIME.

See also:

- `TRIGger<m>:HOLDoff:MODE`

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<Time>
Range: 100E-9 to 10
Increment: 200E-6
*RST: 1E-3
Default unit: s

**Example:**
```
TRIGger:HOLDoff:MODE TIME
TRIGger:HOLDoff:TIME 1ms
```
The holdoff time is set to 1 ms.

**Usage:** Asynchronous command

**Manual operation:** See "Holdoff mode" on page 227

**TRIGger<m>:HOLDoff:EVENts** <Events>

Defines the number of triggers to be skipped. The next trigger only occurs when this number of events is reached. The setting is relevant if the holdoff mode is set to EVENts.

See also:

- `TRIGger<m>:HOLDoff:MODE`

**Suffix:**

<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**

<Events> Range: 1 to 2147483647
Increment: 1
*RST: 1

**Example:**

```
TRIGger:HOLDoff:MODE EVENts
TRIGger:HOLDoff:EVENts 5
```

**Usage:** Asynchronous command

**Manual operation:** See "Holdoff mode" on page 227

**TRIGger<m>:HOLDoff:MIN** <RandomMinTime>

Defines the lower limit for the random time holdoff. The setting is relevant if the holdoff mode is set to RANDom.

See also:

- `TRIGger<m>:HOLDoff:MODE`
- `TRIGger<m>:HOLDoff:MAX`

**Suffix:**

<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**

<RandomMinTime> Range: 100E-9 to 5
Increment: 200E-6
*RST: 1E-3
Default unit: s

**Example:**

```
TRIGger:HOLDoff:MODE RANDom
TRIGger:HOLDoff:MIN 1ms
TRIGger:HOLDoff:MAX 2ms
```

The holdoff time is set randomly between 1 ms and 2 ms.

**Usage:** Asynchronous command

**Manual operation:** See "Holdoff mode" on page 227

**TRIGger<m>:HOLDoff:MAX** <RandomMaxTime>

Defines the upper limit for the random time holdoff. The setting is relevant if the holdoff mode is set to RANDom.

See also:

- `TRIGger<m>:HOLDoff:MODE`
- `TRIGger<m>:HOLDoff:MIN`

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<RandomMaxTime> Range: 100E-9 to 10
Increment: 200E-6
*RST: 2E-3
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Holdoff mode" on page 227

**TRIGger<m>:HOLDoff:AUTotime?**

Returns the resulting holdoff time if the holdoff mode is set to AUTO: *Auto time* = *Auto time scaling* * *Horizontal scale*. The auto time scaling factor is defined with `TRIGger<m>:HOLDoff:SCALing`.

See also: `TRIGger<m>:HOLDoff:MODE`

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Return values:**
<AutoTime> Holdoff time
Range: 100E-9 to 10
*RST: 1E-3
Default unit: s

**Example:**
```
TRIGger:HOLDoff:MODE AUTO
TRIGger:HOLDoff:SCALing 0.5
TRIGger:HOLDoff:AUTotime?
1ms
```
Result if the horizontal scale is 1 ns/div

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Holdoff mode" on page 227

**TRIGger<m>:HOLDoff:SCALing** <AutoTimeScaling>

Sets the auto time scaling factor the horizontal scale is multipied with: *Auto time = Auto time scaling * Horizontal scale*. The setting is relevant if the holdoff mode is set to AUTO.

See also:

- `TRIGger<m>:HOLDoff:MODE`
- `TRIGger<m>:HOLDoff:AUTotime?` on page 1363

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<AutoTimeScaling> Range: 1E-3 to 1000
Increment: 1
*RST: 0.5

**Usage:** Asynchronous command

**Manual operation:** See "Holdoff mode" on page 227

## 24.9.19 Conditioning

TRIGger<m>:LEVel<n>:NOISe[:STATe] ..... 1364
TRIGger<m>:LEVel<n>:NOISe:MODE ..... 1365
TRIGger<m>:LEVel<n>:NOISe:ABSolute ..... 1365
TRIGger<m>:LEVel<n>:NOISe:PERDivision ..... 1366
TRIGger<m>:LEVel<n>:NOISe:RELative ..... 1366

**TRIGger<m>:LEVel<n>:NOISe[:STATe]** <HysteresisMode>

Selects how the hysteresis is set.

**Suffix:**
<m> 1..3
Irrelevant, omit the suffix.

<n> 1..31
Indicates the trigger source:
1...4 = channel 1 to 4
5 = external trigger input
6...9 and 11 = not available
10 = line trigger
12...27 = R&S RT-ZVC input channels
28...31 = differential signals Diff1, Diff2 and Common1, Common2

**Parameters:**
<HysteresisMode> AUTO | MANual

**AUTO**
This is the recommended mode. The hysteresis is set by the instrument to reject at least the internal noise of the instrument. You can define a higher minimum value using `TRIGger<m>:LEVel<n>:NOISe:ABSolute`.

**MANual**
The hysteresis is defined directly with `TRIGger<m>:LEVel<n>:NOISe:ABSolute`.

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis mode" on page 228

---

**TRIGger<m>:LEVel<n>:NOISe:MODE** <HystMode>

Selects how the hysteresis is set.

**Suffix:**
<m> 1..3
Irrelevant, omit the suffix.

<n> 1..31
Indicates the trigger source: see `TRIGger<m>:LEVel<n>:NOISe[:STATe]` on page 1364.

**Parameters:**
<HystMode> ABS | REL

**ABS**
The hysteresis is set in absolute values (voltage).

**REL**
The hysteresis is defined in relative values (div).

*RST: ABS

**Usage:** Asynchronous command

**Manual operation:** See "Size mode" on page 229

---

**TRIGger<m>:LEVel<n>:NOISe:ABSolute** <HystAbs>

Defines a range in absolute values around the trigger level. If the signal oscillates inside this range and crosses the trigger level thereby, no trigger event occurs.

**Suffix:**
<m> 1..3
Irrelevant, omit the suffix.

<n> 1..31
Indicates the trigger source: see `TRIGger<m>:LEVel<n>:NOISe[:STATe]` on page 1364.

**Parameters:**

<HystAbs>
Range: 0 to The value corresponding to full division range. The exact maximum value depends on the selected vertical scale.
Increment: 1E-3
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Absolute hysteresis" on page 229

---

**TRIGger<m>:LEVel<n>:NOISe:PERDivision** <HystInDivs>

Defines a range in divisions around the trigger level. If the signal oscillates inside this range and crosses the trigger level thereby, no trigger event occurs.

**Suffix:**

<m>
1..3
Irrelevant, omit the suffix.

<n>
1..31
Indicates the trigger source: see `TRIGger<m>:LEVel<n>:NOISe[:STATe]` on page 1364.

**Parameters:**

<HystInDivs>
Hysteresis size in divisions
Range: 0 to 5
Increment: 0.01
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Relative hysteresis" on page 229

---

**TRIGger<m>:LEVel<n>:NOISe:RELative** <HystRel>

Defines a range in percent around the trigger level. If the signal oscillates inside this range and crosses the trigger level thereby, no trigger event occurs.

**Suffix:**

<m>
1..3
Irrelevant, omit the suffix.

<n>
1..31
Indicates the trigger source: see `TRIGger<m>:LEVel<n>:NOISe[:STATe]` on page 1364.

**Parameters:**

<HystRel> Hysteresis in %. 10% = 1 div

Range: 0 to 50
Increment: 1
*RST: 0
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Relative hysteresis" on page 229

## 24.9.20 Trigger sequence

TRIGger<m>:EVENt[:EVENt] 1367
TRIGger<m>:SEQuence:TYPE 1367
TRIGger<m>:ECOupling 1368
TRIGger<m>:SEQuence:DELay 1368
TRIGger<m>:SEQuence:COUNt 1369
TRIGger<m>:SEQuence:RESet:EVENt 1369
TRIGger<m>:SEQuence:RESet:TIMeout[:ENABle] 1369
TRIGger<m>:SEQuence:RESet:TIMeout:TIME 1370

**TRIGger<m>:EVENt[:EVENt]** <Class>

Selects, if you want to trigger on a single event, or on a series of events.

**Suffix:**

<m> 1..3

**Parameters:**

<Class> SINGle | SEQuence

*RST: SINGle

**Usage:** Asynchronous command

**Manual operation:** See "Trigger on" on page 197

**TRIGger<m>:SEQuence:TYPE** <Sequence>

Selects the type of the sequence. All trigger sequences require that input channels CH1...4 are set as trigger sources for all conditions.

See also: Chapter 6.8, "Sequence", on page 234.

**Suffix:**

<m> 1..3
Irrelevant, omit the suffix.

**Parameters:**

<Sequence> ABR | AZ | ABRZ | AORB | AORBZ

**ABR**
Triggers if all conditions of A- and B-events, as well as additional delay and reset timeout or R-event (reset) conditions are fulfilled.

**AZ**
Triggers if the conditions of the A-event and the zone trigger are fulfilled. Additionally, a holdoff condition can be set.
Requires option R&S RTP-K19.

**ABRZ**
Triggers if all conditions of A- and B-events, reset timeout or R-event (reset) conditions are fulfilled for the defined zone.
Requires option R&S RTP-K19.

**AORB**
Triggers if the conditions of A-event or the conditions of the B-event are fulfilled.

**AORBZ**
Triggers if the conditions of A-event or the conditions of the B-event are fulfilled for the defined zone. Additionally, a holdoff condition can be set.
Requires option R&S RTP-K19.

*RST: ABR

**Usage:** Asynchronous command

**Manual operation:** See "Sequence" on page 235

---

**TRIGger<m>:ECOupling** <TrigLevEvtCpl>

Event coupling of trigger levels: Sets the trigger levels to the values of the indicated event. Thus, channel 1 has one trigger level for all events, channel 2 has one trigger level and so on.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger

**Parameters:**
<TrigLevEvtCpl> ON | OFF

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Couple sequence thresholds" on page 236

---

**TRIGger<m>:SEQuence:DELay** <Delay>

Sets the time the instrument waits after an A-event until it recognizes B-events.

**Suffix:**
<m> 1..3
Irrelevant, omit the suffix.

**Parameters:**

<Delay>
Range: 0 to 50
Increment: 1E-12
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "After 1st event wait" on page 236

---

**TRIGger<m>:SEQuence:COUNt** <Events>

Sets the number of B-events to be fulfilled after an A-event. The last B-event causes the trigger.

**Suffix:**

<m>
1..3
Irrelevant, omit the suffix.

**Parameters:**

<Events>
Range: 1 to 2147483647
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "2nd event count" on page 236

---

**TRIGger<m>:SEQuence:RESet:EVENt** <EnabRstEvt>

If set to ON, the trigger sequence is restarted by the R-event if the specified number of B-event does not occur.

**Suffix:**

<m>
1..3
Irrelevant, omit the suffix.

**Parameters:**

<EnabRstEvt>
ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Reset on event/Reset event" on page 236

---

**TRIGger<m>:SEQuence:RESet:TIMeout[:ENABle]** <State>

If set to ON, the instrument waits for the time defined using `TRIGger<m>:SEQuence:RESet:TIMeout:TIME` for the specified number of B-events. If no trigger occurs during that time, the sequence is restarted with the A-event.

**Suffix:**

<m>
1..3
Irrelevant, omit the suffix.

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Reset on timeout/Reset timeout" on page 236

---

**TRIGger<m>:SEQuence:RESet:TIMeout:TIME** <ResetTimeout>

The time the instrument waits for the number of B-events specified using `TRIGger<m>:SEQuence:COUNt` before the sequence is restarted with the A-event.

**Suffix:**
<m> 1..3
Irrelevant, omit the suffix.

**Parameters:**
<ResetTimeout> Range: 0 to 50
Increment: 1E-12
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Reset on timeout/Reset timeout" on page 236

### 24.9.21 Trigger control

TRIGger<m>:MODE ..... 1370
TRIGger<m>:FORCe ..... 1371
TRIGger<m>:OUT:POLarity ..... 1371
TRIGger<m>:OUT:PLENgth ..... 1371
TRIGger<m>:OUT:DELay ..... 1372

---

**TRIGger<m>:MODE** <TriggerMode>

Sets the trigger mode which determines the behaviour of the instrument if no trigger occurs.

**Suffix:**
<m> 1..3
Irrelevant, omit the suffix.

**Parameters:**
<TriggerMode> AUTO | NORMal | FREerun

**AUTO**
The instrument triggers repeatedly after a time interval if the trigger conditions are not fulfilled. If a real trigger occurs, it takes precedence. The time interval depends on the time base.

**NORMal**
The instrument acquires a waveform only if a trigger occurs.

**FREerun**
The instrument triggers after a very short time interval - faster than in AUTO mode. Real triggers are ignored.

*RST: AUTO

**Usage:** Asynchronous command

**Manual operation:** See "[Auto Norm]" on page 37

---

**TRIGger<m>:FORCe**

If the acquisition is running in normal mode and no valid trigger occurs, forcing the trigger provokes an immediate single acquisition. Thus you can confirm that a signal is available and use the waveform display to determine how to trigger on it.

**Suffix:**
<m> 1..3
Irrelevant, omit the suffix.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Force Trigger" on page 227

---

**TRIGger<m>:OUT:POLarity** <Polarity>

Sets the polarity of the trigger out pulse.

**Suffix:**
<m> 1..3
Irrelevant, omit the suffix.

**Parameters:**
<Polarity> POSitive | NEGative
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Trigger out signal setup" on page 230

---

**TRIGger<m>:OUT:PLENgth** <PulseLength>

Sets the length of the trigger out pulse.

**Suffix:**
<m> 1..3
Irrelevant, omit the suffix.

**Parameters:**
<PulseLength> Range: 4E-9 to 0.06
Increment: 20E-9
*RST: 100E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Trigger out signal setup" on page 230

---

**TRIGger<m>:OUT:DELay** <Delay>

Sets the delay of the first pulse edge to the trigger point.

The setting is not available if a mask test or measurement is running and the on-violation event is set to trigger out pulse.

**Suffix:**
<m> 1..3
Irrelevant, omit the suffix.

**Parameters:**
<Delay> Range: 1.8E-6 to 512E-3
Increment: 1E-9
*RST: 1.8E-6
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Trigger out signal setup" on page 230

## 24.9.22 Actions on trigger

TRIGger<m>:EVENt:BEEP ..... 1372
TRIGger<m>:EVENt:PRINt ..... 1372
TRIGger<m>:EVENt:WFMSave ..... 1373
TRIGger<m>:EVENt:RUNexec ..... 1373
TRIGger<m>:OUT:ACTion ..... 1374

---

**TRIGger<m>:EVENt:BEEP** <Beep>

Generates a beep sound if the command is set to `TRIGger`.

**Suffix:**
<m> 1..3
The suffix is irrelevant.

**Parameters:**
<Beep> NOACtion | TRIGger
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Actions on trigger" on page 230

---

**TRIGger<m>:EVENt:PRINt** <Print>

Saves a screenshot at each trigger if the command is set to `TRIGger`.

For screenshot settings, see Chapter 24.16.8, "Screenshots", on page 1627.

**Suffix:**
<m> 1..3
The suffix is irrelevant.

**Parameters:**
<Print> NOACtion | TRIGger
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Actions on trigger" on page 230

---

**TRIGger<m>:EVENt:WFMSave** <SaveWfm>

Saves the waveform data to file at each trigger if the command is set to `TRIGger`.

For data export settings, see Chapter 24.16.4, "Waveform data export to file", on page 1613

**Suffix:**
<m> 1..3
The suffix is irrelevant.

**Parameters:**
<SaveWfm> NOACtion | TRIGger
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Actions on trigger" on page 230

---

**TRIGger<m>:EVENt:RUNexec** <RunExecutable>

Starts an external application if the command is set to `TRIGger`.

Use the following commands to set up the application:

- `EXECutable:NAME` on page 1635
- `EXECutable:PARameter` on page 1635
- `EXECutable:WDIRectory` on page 1636

**Suffix:**
<m> 1..3
The suffix is irrelevant.

**Parameters:**
<RunExecutable> NOACtion | TRIGger
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Actions on trigger" on page 230

**TRIGger<m>:OUT:ACTion** <StateAction>

Enables/disbales the trigger out signal that is provided to the [Trigger Out] connector on the rear panel when a trigger occurs.

**Suffix:**
<m> 1..3

**Parameters:**
<StateAction> NOACtion | TRIGger
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Actions on trigger" on page 230

### 24.9.23 Zone trigger (option R&S RTP-K19)

**TRIGger<m>:ZONE:EXPRession[:DEFine]** <LogiExpr>

Defines the zone trigger.

**Suffix:**
<m> 1..3
The suffix is irrelevant.

**Parameters:**
<LogiExpr> String with the logical expression

**Example:** Prerequisite: mask tests MT1 and MT2 are defined, and also Zone1.

```
TRIGger:ZONE:EXPRession 'MT1 and MT2 and Zone1'
TRIGger:ZONE:EXPRession?
<-- MT1 and MT2 and Zone1
```

**Usage:** Asynchronous command

**Firmware/software:** FW 3.20

**Manual operation:** See "[Zone]" on page 37

**SWTRigger:HISTory** <ApplyZnTrigHistory>

Applies the zone trigger condition to the acquisitions in the history memory during history replay.

See: "Apply zone trigger to history" on page 240.

**Parameters:**
<ApplyZnTrigHistory> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Apply zone trigger to history" on page 240

### 24.9.24 External trigger input

External trigger signals, which are connected to the external trigger input, can be triggered with an edge trigger.

To control the external trigger signal, the TRPRobe commands are used. The required commands depend on the used probe type. They work in the same way as the PROBe commands. For details, see Chapter 24.8.7, "Probes", on page 1254

---

**TRIGger<m>:ANEDge:COUPling** <Coupling>

Sets the coupling for the external trigger signal.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<Coupling> DC

**DC**
Direct connection with 50 Ω termination, passes both DC and AC components of the trigger signal.

*RST: DC

**Usage:** Asynchronous command

**Manual operation:** See "Coupling" on page 257

---

**TRIGger<m>:ANEDge:FILTer** <Filter>

Sets a filter for the external trigger signal to reject high or low frequencies.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<Filter> OFF | LFReject | RFReject

**OFF**
The trigger signal is not filtered.

**LFReject**
Frequencies higher than the "Cut-off" frequency are rejected, lower frequencies pass the filter.
You can adjust the "Cut-off" frequency using the `TRIGger<m>:ANEDge:CUToff:LOWPass` command, the default is 50 kHz.

**RFReject**
Frequencies below the "Cut-off" frequency are rejected, higher frequencies pass the filter.
You can adjust the "Cut-off" frequency using the `TRIGger<m>:ANEDge:CUToff:HIGHpass` command, the default is 50 kHz.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Filter" on page 257

---

**TRIGger<m>:ANEDge:CUToff:HIGHpass** <AnalogCutOffHP>

Frequencies below the "Cut-off" frequency are rejected, higher frequencies pass the filter.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<AnalogCutOffHP> KHZ5 | KHZ50 | MHZ50

Cut-off frequency

**KHZ5**
5 kHz

**KHZ50**
50 kHz

**MHZ50**
50 MHz

*RST: KHZ50

**Usage:** Asynchronous command

---

**TRIGger<m>:ANEDge:CUToff:LOWPass** <AnalogCutOffLP>

Frequencies higher than the "Cut-off" frequency are rejected, lower frequencies pass the filter.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<AnalogCutOffLP> KHZ5 | KHZ50 | MHZ50

**KHZ5**
5 kHz

**KHZ50**
50 kHz

**MHZ50**
50 MHz

*RST: KHZ50

**Usage:** Asynchronous command

---

**TRIGger<m>:ANEDge:GND** <Ground>

Connects the analog signal to the ground.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<Ground> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Ground" on page 257

---

**TRIGger<m>:ANEDge:SLOPe** <Slope>

Sets the edge for the trigger event.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<Slope> POSitive | NEGative
See Chapter 24.4.3, "Slope parameter", on page 1194.
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 199

---

**TRIGger<m>:EXTern:OVERload** <Overload>

The query returns the overload status of the external trigger input.

`:TRIGger:EXTern:OVERload 0` confirms the information in the message box, it has same effect as OK.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Parameters:**
<Overload> ON | OFF

**ON | 1**
lindicates an overload of the probe

**OFF | 0**
In a query: no overlaod.
Ssetting: confirms the information in the message box

*RST: OFF

**Usage:** Asynchronous command

### 24.9.25 Acquisition info

**ACQuire:CURRent?**

Shows the current number of acquisitions that have been acquired.

**Return values:**

<CurrAcqCnt> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

## 24.10 Waveform analysis

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

- Zoom....1378
- Reference waveforms....1388
- Mathematics....1400
- History....1405
- XY-Diagram....1410

### 24.10.1 Zoom

LAYout:ZOOM:ADD....1379
LAYout:ZOOM:ADDCoupled....1379
LAYout:ZOOM:POSCoupling....1380
LAYout:ZOOM:HORZ:MODE....1380
LAYout:ZOOM:HORZ:ABSolute:POSition....1381
LAYout:ZOOM:HORZ:ABSolute:SPAN....1381
LAYout:ZOOM:HORZ:ABSolute:STARt....1381
LAYout:ZOOM:HORZ:ABSolute:STOP....1382
LAYout:ZOOM:HORZ:RELative:POSition....1382
LAYout:ZOOM:HORZ:RELative:SPAN....1383
LAYout:ZOOM:HORZ:RELative:STARt....1383
LAYout:ZOOM:HORZ:RELative:STOP....1383

LAYout:ZOOM:VERTical:MODE....1384
LAYout:ZOOM:VERTical:ABSolute:POSition....1384
LAYout:ZOOM:VERTical:ABSolute:SPAN....1385
LAYout:ZOOM:VERTical:ABSolute:STARt....1385
LAYout:ZOOM:VERTical:ABSolute:STOP....1385
LAYout:ZOOM:VERTical:RELative:POSition....1386
LAYout:ZOOM:VERTical:RELative:SPAN....1386
LAYout:ZOOM:VERTical:RELative:STARt....1387
LAYout:ZOOM:VERTical:RELative:STOP....1387
LAYout:ZOOM:REMove....1387

**LAYout:ZOOM:ADD** <NodeName>, <ParentType>, <InsertBefore>, <XStart>, <XStop>, <YStart>, <YStop>, <NewZoomName>

Adds a new zoom diagram based on the specified waveform.

For numerical values, the input of a unit is not supported for this command.

**Setting parameters:**

<NodeName> String with the name of diagram to be zoomed

<ParentType> VERTical, OFF
The new zoom diagram is displayed below the original one.

<InsertBefore> OFF
Position of the zoom diagram, depending on `ParentType`

<XStart> Defines the x-value at the beginning of the zoom area.

<XStop> Defines the x-value at the end of the zoom area.

<YStart> Defines the y-value at the beginning of the zoom area.

<YStop> Defines the y-value at the end of the zoom area.

<NewZoomName> String with the name of the new zoom diagram.

**Example:** `LAYout:ZOOM:ADD 'Diagram1', VERT, OFF, -10e-9, 20e-9, -0.1, 0.05, 'MyZoom1'`
Creates the zoom diagram 'MyZoom1' for 'Diagram1'.

**Example:** See Chapter 24.3.1.2, "Creating zoom diagrams", on page 1174

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Standard zoom" on page 59

**LAYout:ZOOM:ADDCoupled** <ZoomName>, <XOffset>, <YOffset>, <NewZoomName>

Creates a new zoom diagram based on the settings of an existing zoom area for the same source.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**
<NewZoomName> Defines the name of the new zoom diagram.

**Setting parameters:**
<ZoomName> Defines the name of the zoom diagram to be copied.

<XOffset> Defines an offset to the existing zoom area in x direction.

<YOffset> Defines an offset to the existing zoom area in y direction.

**Usage:** Asynchronous command

**Manual operation:** See "Coupled zoom" on page 59

---

**LAYout:ZOOM:POSCoupling** <DiagramName>, <ZoomName>, <PositionCoupl>
**LAYout:ZOOM:POSCoupling?** <DiagramName>, <ZoomName>

Enables or disables the position coupling of coupled zooms. If position coupling is enabled and one zoom area is moved, the other coupled zoom areas are moved, too, and keep their distance.

**Parameters:**
<PositionCoupl> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<DiagramName> String with the name of the diagram on which the zoom is based

<ZoomName> String with the name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "Position coupling" on page 266

---

**LAYout:ZOOM:HORZ:MODE** <DiagramName>, <ZoomName>,<Mode>
**LAYout:ZOOM:HORZ:MODE?** <DiagramName>, <ZoomName>

Defines whether absolute or relative values are used to specify the x-axis values. Since the zoom area refers to the active signal, relative values ensure that the zoom area remains the same.

**Parameters:**
<Mode> ABS | REL
Mode used to specify the x-axis values of the zoom area.
*RST: ABS

**Parameters for setting and query:**
<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Example:** See Chapter 24.3.1.2, "Creating zoom diagrams", on page 1174

**Usage:** Asynchronous command

**Manual operation:** See "X mode" on page 266

**LAYout:ZOOM:HORZ:ABSolute:POSition** <DiagramName>,
<ZoomName>,<Position>
**LAYout:ZOOM:HORZ:ABSolute:POSition?** <DiagramName>, <ZoomName>

Defines the x-value of the centerpoint of the zoom area.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**

<Position>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**

<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "X position" on page 265

---

**LAYout:ZOOM:HORZ:ABSolute:SPAN** <DiagramName>, <ZoomName>,<Span>
**LAYout:ZOOM:HORZ:ABSolute:SPAN?** <DiagramName>, <ZoomName>

Defines the width of the zoom area.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**

<Span>
Range: 0 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**

<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "X range" on page 266

---

**LAYout:ZOOM:HORZ:ABSolute:STARt** <DiagGroupNme>,
<ZoomDiagramName>,<Start>
**LAYout:ZOOM:HORZ:ABSolute:STARt?** <DiagGroupNme>, <ZoomDiagramName>

Defines the lower limit of the zoom area on the x-axis.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**

<Start>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0

**Parameters for setting and query:**
<DiagGroupNme> Name of the diagram on which the zoom area is based.

<ZoomDiagramName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "X start" on page 265

---

**LAYout:ZOOM:HORZ:ABSolute:STOP** <DiagramName>, <ZoomName>,<Stop>
**LAYout:ZOOM:HORZ:ABSolute:STOP?** <DiagramName>, <ZoomName>

Defines the upper limit of the zoom area on the x-axis.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**
<Stop>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**
<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "X stop" on page 265

---

**LAYout:ZOOM:HORZ:RELative:POSition** <DiagramName>,
<ZoomName>,<RelPosi>
**LAYout:ZOOM:HORZ:RELative:POSition?** <DiagramName>, <ZoomName>

Defines the x-value of the centerpoint of the zoom area.

**Parameters:**
<RelPosi> Relative position of the centerpoint (x-value)
Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Parameters for setting and query:**
<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Example:** See Chapter 24.3.1.2, "Creating zoom diagrams", on page 1174

**Usage:** Asynchronous command

**Manual operation:** See "X position" on page 265

**LAYout:ZOOM:HORZ:RELative:SPAN** <DiagramName>,
<ZoomName>,<RelativeSpan>
**LAYout:ZOOM:HORZ:RELative:SPAN?** <DiagramName>, <ZoomName>

Defines the width of the zoom area.

**Parameters:**
<RelativeSpan> Range: 1E-15 to 100
Increment: 0.1
*RST: 1
Default unit: %

**Parameters for setting and query:**
<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Example:** See Chapter 24.3.1.2, "Creating zoom diagrams", on page 1174

**Usage:** Asynchronous command

**Manual operation:** See "X range" on page 266

---

**LAYout:ZOOM:HORZ:RELative:STARt** <DiagramName>,
<ZoomName>,<RelativeStart>
**LAYout:ZOOM:HORZ:RELative:STARt?** <DiagramName>, <ZoomName>

Defines the lower limit of the zoom area on the x-axis.

**Parameters:**
<RelativeStart> Range: 0 to 100
Increment: 0.1
*RST: 0
Default unit: %

**Parameters for setting and query:**
<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "X start" on page 265

---

**LAYout:ZOOM:HORZ:RELative:STOP** <DiagramName>,
<ZoomName>,<RelativeStop>
**LAYout:ZOOM:HORZ:RELative:STOP?** <DiagramName>, <ZoomName>

Defines the upper limit of the zoom area on the x-axis.

**Parameters:**
<RelativeStop> Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Parameters for setting and query:**

<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "X stop" on page 265

---

**LAYout:ZOOM:VERTical:MODE** <DiagramName>, <ZoomName>,<Mode>
**LAYout:ZOOM:VERTical:MODE?** <DiagramName>, <ZoomName>

Defines whether absolute or relative values are used to specify the y-axis values. Since the zoom area refers to the active signal, relative values ensure that the zoom area remains the same.

**Parameters:**

<Mode> ABS | REL

Mode used to specify the y-axis values of the zoom area.

*RST: ABS

**Parameters for setting and query:**

<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "Y mode" on page 266

---

**LAYout:ZOOM:VERTical:ABSolute:POSition** <DiagramName>, <ZoomName>,<Position>
**LAYout:ZOOM:VERTical:ABSolute:POSition?** <DiagramName>, <ZoomName>

Defines the y-value of the centerpoint of the zoom area.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**

<Position> Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**

<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "Y position" on page 266

**LAYout:ZOOM:VERTical:ABSolute:SPAN** <DiagramName>, <ZoomName>,<Span>
**LAYout:ZOOM:VERTical:ABSolute:SPAN?** <DiagramName>, <ZoomName>

Defines the height of the zoom area.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**

<Span>
Range: 0 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**

<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "Y range" on page 266

**LAYout:ZOOM:VERTical:ABSolute:STARt** <DiagGroupNme>,
<ZoomDiagramName>,<Start>
**LAYout:ZOOM:VERTical:ABSolute:STARt?** <DiagGroupNme>,
<ZoomDiagramName>

Defines the lower limit of the zoom area on the y-axis.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**

<Start>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0

**Parameters for setting and query:**

<DiagGroupNme> Name of the diagram on which the zoom area is based.

<ZoomDiagramName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "Y start" on page 265

**LAYout:ZOOM:VERTical:ABSolute:STOP** <DiagramName>, <ZoomName>,<Stop>
**LAYout:ZOOM:VERTical:ABSolute:STOP?** <DiagramName>, <ZoomName>

Defines the upper limit of the zoom area on the y-axis.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**

<Stop>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**

<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "Y stop" on page 265

---

**LAYout:ZOOM:VERTical:RELative:POSition** <DiagramName>, <ZoomName>,<RelPosi>
**LAYout:ZOOM:VERTical:RELative:POSition?** <DiagramName>, <ZoomName>

Defines the y-value of the centerpoint of the zoom area.

**Parameters:**

<RelPosi> Relative position of the centerpoint (y-value)

Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Parameters for setting and query:**

<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "Y position" on page 266

---

**LAYout:ZOOM:VERTical:RELative:SPAN** <DiagramName>, <ZoomName>,<RelativeSpan>
**LAYout:ZOOM:VERTical:RELative:SPAN?** <DiagramName>, <ZoomName>

Defines the height of the zoom area.

**Parameters:**

<RelativeSpan> Range: 1E-15 to 100
Increment: 0.1
*RST: 1
Default unit: %

**Parameters for setting and query:**

<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "Y range" on page 266

**LAYout:ZOOM:VERTical:RELative:STARt** <DiagramName>,
<ZoomName>,<RelativeStart>
**LAYout:ZOOM:VERTical:RELative:STARt?** <DiagramName>, <ZoomName>

Defines the lower limit of the zoom area on the y-axis.

**Parameters:**
<RelativeStart> Range: 0 to 100
Increment: 0.1
*RST: 0
Default unit: %

**Parameters for setting and query:**
<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "Y start" on page 265

**LAYout:ZOOM:VERTical:RELative:STOP** <DiagramName>,
<ZoomName>,<RelativeStop>
**LAYout:ZOOM:VERTical:RELative:STOP?** <DiagramName>, <ZoomName>

Defines the upper limit of the zoom area on the y-axis.

**Parameters:**
<RelativeStop> Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Parameters for setting and query:**
<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Usage:** Asynchronous command

**Manual operation:** See "Y stop" on page 265

**LAYout:ZOOM:REMove** <DiagramName>, <ZoomName>

Removes the specified zoom diagram.

**Setting parameters:**
<DiagramName> Name of the diagram on which the zoom area is based.

<ZoomName> Name of the zoom diagram

**Example:** See Chapter 24.3.1.2, "Creating zoom diagrams", on page 1174

**Usage:** Setting only
Asynchronous command

## 24.10.2 Reference waveforms


- Reference....1388
- Scaling....1390
- Waveform data export....1395
- Import of multichannel waveform data....1396
- CDR reference files....1398


### 24.10.2.1 Reference


REFCurve<m>:SOURce....1388
REFCurve<m>:STATe....1388
REFCurve<m>:NAME....1389
REFCurve<m>:OPEN....1389
REFCurve<m>:UPDate....1389
REFCurve<m>:SAVE....1389
REFCurve<m>:DELete....1390
REFCurve<m>:CLEar....1390


**REFCurve<m>:SOURce** <Source>

Selects the source waveform to be used as a reference.

**Suffix:**
<m> 1..4
Reference waveform

**Parameters:**
<Source> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | R1 | R2 | R3 | R4 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2

Source of the reference waveform, see Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 274

**REFCurve<m>:STATe** <State>

If enabled, the reference waveform is displayed in the diagram.

**Suffix:**
<m> 1..4
Reference waveform

**Parameters:**

<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show" on page 274

---

**REFCurve<m>:NAME** <Name>

Defines the name of the reference waveform file to be loaded, saved or deleted.

**Suffix:**

<m> 1..4
Reference waveform

**Parameters:**

<Name> Path and name of the file that contains the reference waveform or to which the reference waveform is to be stored (`.xml` or `.bin` format), enclosed in single quotes.

**Usage:** Asynchronous command

---

**REFCurve<m>:OPEN**

Loads the reference waveform file selected by `REFCurve<m>:NAME` on page 1389.

**Suffix:**

<m> 1..4
Reference waveform

**Usage:** Event

**Manual operation:** See "File" on page 275

---

**REFCurve<m>:UPDate**

Copies the selected source waveform (see `REFCurve<m>:SOURce` on page 1388) with all its settings to the memory of the reference waveform.

**Suffix:**

<m> 1..4
Reference waveform

**Usage:** Event

**Manual operation:** See "Update" on page 275

---

**REFCurve<m>:SAVE**

Saves the reference waveform to the file selected by `REFCurve<m>:NAME` on page 1389.

**Suffix:**
<m> 1..4
Reference waveform

**Usage:** Event

**Manual operation:** See "File" on page 275

---

**REFCurve<m>:DELete**

Deletes the reference waveform file selected by `REFCurve<m>:NAME` on page 1389.

**Suffix:**
<m> 1..4
Reference waveform

**Usage:** Event

**Manual operation:** See "File" on page 275

---

**REFCurve<m>:CLEar**

The selected reference waveform is no longer displayed, its memory is deleted.

**Suffix:**
<m> 1..4
Reference waveform

**Usage:** Event
Asynchronous command

**Manual operation:** See "Clear" on page 275

### 24.10.2.2 Scaling

REFCurve<m>:RESTore....1391
REFCurve<m>:VMODe....1391
REFCurve<m>:SCALe....1391
REFCurve<m>:POSition....1392
REFCurve<m>:TOORignal....1392
REFCurve<m>:RESCale:VERTical:STATe....1392
REFCurve<m>:RESCale:VERTical:FACTor....1392
REFCurve<m>:RESCale:VERTical:OFFSet....1393
REFCurve<m>:HMODe....1393
REFCurve<m>:RESCale:HORizontal:STATe....1394
REFCurve<m>:RESCale:HORizontal:FACTor....1394
REFCurve<m>:RESCale:HORizontal:OFFSet....1394
REFCurve<m>:RESCale:HORizontal:POSition....1395

**REFCurve<m>:RESTore**

Resets the time scale and the reference point to the original values of the reference waveform.

**Suffix:**
<m> 1..4
Reference waveform

**Usage:** Event
Asynchronous command

**Manual operation:** See "Restore Settings" on page 276

**REFCurve<m>:VMODe** <VerticalMode>

Selects the coupling of vertical settings.

**Suffix:**
<m> 1..4
Reference waveform

**Parameters:**
<VerticalMode> COUPled | INDependent

**COUPled**
Vertical position and scale of the source are used.

**INDependent**
Scaling and position can be set specific to the reference waveform.

*RST: INDependent

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 276

**REFCurve<m>:SCALe** <VerticalScale>

Sets the scale factor for the reference waveform if vertical scaling is set to "Independent" (see `REFCurve<m>:VMODe` on page 1391).

**Suffix:**
<m> 1..4
Reference waveform

**Parameters:**
<VerticalScale> Range: 1E-15 to 100E+24
Increment: 10E-6
*RST: 0.5
Default unit: V/div

**Usage:** Asynchronous command

**Manual operation:** See "Scale" on page 277

**REFCurve<m>:POSition** <VertPosi>

Moves the reference waveform and its horizontal axis up or down in the diagram, if vertical scaling is set to "Independent" (see `REFCurve<m>:VMODe` on page 1391).

**Suffix:**
<m> 1..4
Reference waveform

**Parameters:**
<VertPosi> Range: -100E+24 to 100E+24
Increment: 0.02
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Position" on page 277

**REFCurve<m>:TOORignal**

Restores the original vertical settings of the reference waveform (vertical scale, position, and offset), if vertical scaling is set to independent (see `REFCurve<m>:VMODe` on page 1391).

**Suffix:**
<m> 1..4

**Usage:** Event
Asynchronous command

**Manual operation:** See "Set to original" on page 277

**REFCurve<m>:RESCale:VERTical:STATe** <State>

Enables and disables the vertical stretching. Stretching changes the display of the waveform independent of the vertical scale and position.

**Suffix:**
<m> 1..4
Reference waveform

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 278

**REFCurve<m>:RESCale:VERTical:FACTor** <ScaleFactor>

Defines the vertical stretching factor. A factor greater than 1 stretches the waveform vertically, a factor lower than 1 compresses the curve.

**Suffix:**
<m> 1..4
Reference waveform

**Parameters:**
<ScaleFactor> Range: -1E+6 to 1E+6
Increment: 0.1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Factor" on page 278

---

**REFCurve<m>:RESCale:VERTical:OFFSet** <Offset>

Moves the reference waveform vertically. Like vertical offset of channel waveforms, the offset of a reference waveform is subtracted from the measured value.

**Suffix:**
<m> 1..4
Reference waveform

**Parameters:**
<Offset> Negative values shift the waveform up, positive values shift it down.
Range: -100E+24 to 100E+24
Increment: 1E-6
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Offset" on page 278

---

**REFCurve<m>:HMODe** <HorizontalMode>

Selects the coupling of horizontal settings.

**Suffix:**
<m> 1..4
Reference waveform

**Parameters:**
<HorizontalMode> ORIGinal | COUPled

**ORIGinal**
Horizontal scaling and reference point of the source waveform are used.

**COUPled**
The current horizontal settings of the diagram are used.

*RST: ORIGinal

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 278

---

**REFCurve<m>:RESCale:HORizontal:STATe** <State>

Enables and disables the horizontal stretching.

Stretching changes the display of the waveform independent of the horizontal settings of the source waveform and of the horizontal diagram settings.

**Suffix:**
<m> 1..4
Reference waveform

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 278

---

**REFCurve<m>:RESCale:HORizontal:FACTor** <ScaleFactor>

A factor greater than 1 stretches the waveform horizontally, a factor lower than 1 compresses the curve.

**Suffix:**
<m> 1..4
Reference waveform

**Parameters:**
<ScaleFactor> Range: 1E-6 to 1E+6
Increment: 0.1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Factor" on page 278

---

**REFCurve<m>:RESCale:HORizontal:OFFSet** <Offset>

Moves the waveform horizontally. Positive values shift the waveform to the right, negative values shift it to the left.

**Suffix:**
<m> 1..4
Reference waveform

**Parameters:**
<Offset> Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0
Default unit: s

**Usage:** Asynchronous command

---

**REFCurve<m>:RESCale:HORizontal:POSition** <Offset>

Moves the waveform horizontally. Enter a value with a time unit suitable for the time scale of the diagram. Positive values shift the waveform to the right, negative values shift it to the left.

**Suffix:**
<m> 1..4

**Parameters:**
<Offset>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Position" on page 279

### 24.10.2.3 Waveform data export

Commands for saving waveform data to file are described in Chapter 24.16.4, "Waveform data export to file", on page 1613.

Commands for reference waveforms are listed below.

REFCurve<m>:DATA:STYPe?....1395
REFCurve<m>:DATA:HEADer?....1395
REFCurve<m>:DATA[:VALues]?....1396

---

**REFCurve<m>:DATA:STYPe?**

Returns the signal type of the source of the reference waveform.

**Suffix:**
<m> 1..4
Reference waveform

**Return values:**
<SignalType> SOUR | SPEC | CORR | NONE
SOURce = normal signal
SPECtrum = FFT spectrum, specific math signal
CORRelation = correlated signal, specific math signal
NONE = undefined

**Usage:** Query only

---

**REFCurve<m>:DATA:HEADer?**

Returns information on the reference waveform.

*Table 24-4: Header data*

| Position | Meaning | Example |
|---|---|---|
| 1 | XStart in s | -9.477E-008 = - 94,77 ns |
| 2 | XStop in s | 9.477E-008 = 94,77 ns |
| 3 | Record length of the waveform in Samples | 200000 |
| 4 | Number of values per sample interval. For reference waveforms the number is always 1. | 1 |

**Suffix:**
<m> 1..4
Reference waveform

**Example:**
```
REFC:DATA:HEAD?
-9.477E-008,9.477E-008,200000,1
```

**Usage:** Query only

---

**REFCurve<m>:DATA[:VALues]?**

Returns the data of the channel waveform points for transmission from the instrument to the controlling computer. The data can be used in MATLAB, for example.

To set the export format, use `FORMat[:DATA]`.

You can retrieve only Y-values, or X- and Y-values. Use `EXPort:WAVeform:INCXvalues` to define this.

**Suffix:**
<m> 1..4
Reference waveform

**Return values:**
<Data> List of values according to the format and content settings.

**Usage:** Query only

### 24.10.2.4 Import of multichannel waveform data

Commands for saving waveform data to file are described in Chapter 24.16.4, "Waveform data export to file", on page 1613.

Commands for reference waveforms are listed below.

REFCurve<m>:MULTichannel:NAME ..... 1397
REFCurve<m>:MULTichannel:IMPort ..... 1397
REFCurve<m>:MULTichannel:OPEN ..... 1397

**REFCurve<m>:MULTichannel:NAME** <MltChImptPath>

Defines the path and the file to be imported. If not path is given, the default path `C:\Users\Public\Public Documents\Rohde-Schwarz\RTx\RefWaveforms` is used.

**Suffix:**
<m> 1..4
The suffix is irrelevant.

**Parameters:**
<MltChImptPath> String with path and filename

**Example:** See `REFCurve<m>:MULTichannel:IMPort` on page 1397.

**Usage:** Asynchronous command

**REFCurve<m>:MULTichannel:IMPort** <MltChImptWfmSel>

Assigns a waveform from the multichannel file to a reference waveform. To import all waveforms that are in the file, you must assign each waveform to another reference waveform.

**Suffix:**
<m> 1..4
Selects the reference waveform.

**Parameters:**
<MltChImptWfmSel> NONE | NONE | WF1 | WAVeform1 | WF2 | WAVeform2 | WF3 | WAVeform3 | WF4 | WAVeform4

WF1 = WAVeform1, WF2 = WAVeform2, WF3 = WAVeform3, WF4 = WAVeform4
Selects the waveform in the export file.

*RST: NONE

**Example:** Prerequisite: file `RefCurve_2016-02-16_01.bin` is a multi-channel export with at least WF1, WF3 and WF4.

```
REFCurve:MULTichannel:NAME 'RefCurve_2016-02-16_01.bin'
REFCurve1:MULTichannel:IMPort WF1
REFCurve2:MULTichannel:IMPort WF3
REFCurve3:MULTichannel:IMPort WF4
REFCurve:MULTichannel:OPEN
```

**Usage:** Asynchronous command

**Firmware/software:** FW 3.20

**REFCurve<m>:MULTichannel:OPEN**

Loads the waveform data to the reference waveforms.

**Suffix:**
<m> 1..4
The suffix is irrelevant.

**Example:** See REFCurve<m>:MULTichannel:IMPort on page 1397.

**Usage:** Event
Asynchronous command

### 24.10.2.5 CDR reference files

The following commands query the settings of the reference waveform, after a RefCurve.CDR.* file has been recalled.

REFCurve<m>:CDR:ALGorithm? ..... 1398
REFCurve<m>:CDR:BITRate? ..... 1398
REFCurve<m>:CDR:BWIDth? ..... 1399
REFCurve<m>:CDR:RELBwidth? ..... 1399
REFCurve<m>:CDR:SAMTime? ..... 1399
REFCurve<m>:CDR:SOURce? ..... 1400

---

**REFCurve<m>:CDR:ALGorithm?**

Queries the algorithm of the CDR reference waveform.

**Suffix:**
<m> 1..4
Reference waveform

**Return values:**
<Algorithm> CFRequency | FF
CFRequency: constant frequency
FF: feed forward
*RST: FF

**Usage:** Query only
Asynchronous command

**Manual operation:** See "CDR" on page 927

---

**REFCurve<m>:CDR:BITRate?**

Queries the bit rate of the CDR reference waveform.

**Suffix:**
<m> 1..4
Reference waveform

**Return values:**
<Bitrate> Range: 1000 to 40E+9
Increment: 10
*RST: 1E+9
Default unit: bps

**Usage:** Query only
Asynchronous command

**Manual operation:** See "CDR" on page 927

---

**REFCurve<m>:CDR:BWIDth?**

Queries the bandwidth of the CDR reference waveform.

**Suffix:**
<m> 1..4
Reference waveform

**Return values:**
<Bandwidth> Range: 20E+3 to 10E+6
Increment: 10
*RST: 599.88E+3
Default unit: Hz

**Usage:** Query only
Asynchronous command

**Manual operation:** See "CDR" on page 927

---

**REFCurve<m>:CDR:RELBwidth?**

Queries the relative bandwidth of the CDR reference waveform.

**Suffix:**
<m> 1..4
Reference waveform

**Return values:**
<RelBw> Range: 500 to 3000
Increment: 1
*RST: 1667

**Usage:** Query only
Asynchronous command

**Manual operation:** See "CDR" on page 927

---

**REFCurve<m>:CDR:SAMTime?**

Queries the sampling time of the CDR reference waveform.

**Suffix:**
<m> 1..4
Reference waveform

**Return values:**

<SamplingTime> Range: 0 to 1
Increment: 0.01
*RST: 0.5
Default unit: UI

**Usage:** Query only
Asynchronous command

**Manual operation:** See "CDR" on page 927

---

**REFCurve<m>:CDR:SOURce?**

Queries the analog source of the CDR refernce waveform.

**Suffix:**

<m> 1..4
Reference waveform

**Return values:**

<AnalogSource> C1 | C2 | C3 | C4 | DIFF1 | DIFF2 | COMMON1 | COMMON2
*RST: C1

**Usage:** Query only
Asynchronous command

### 24.10.3 Mathematics

CALCulate:MATH<m>[:EXPRession][:DEFine] ..... 1400
CALCulate:MATH<m>:STATe ..... 1401
CALCulate:MATH<m>:ENVSelection ..... 1401
CALCulate:MATH<m>:ARIThmetics ..... 1401
CALCulate:MATH<m>:VERTical:OFFSet ..... 1402
CALCulate:MATH<m>:VERTical:RANGe ..... 1403
CALCulate:MATH<m>:VERTical:SCALe ..... 1403
CALCulate:MATH<m>:VERTical:MAXimum ..... 1403
CALCulate:MATH<m>:DATA:STYPe? ..... 1404
CALCulate:MATH<m>:DATA:HEADer? ..... 1404
CALCulate:MATH<m>:DATA[:VALues]? ..... 1405

---

**CALCulate:MATH<m>[:EXPRession][:DEFine]** <RemComplExpr>

Defines the math expression to be calculated for the specified math channel.

For an overview of corresponding expressions for the available keys in the formula editor, see Chapter 7.3.4, "Equation", on page 284.

**Suffix:**

<m> 1..8
Selects the math waveform.

**Parameters:**
<RemComplExpr> String with regular expression for calculation

**Example:** `CALC:MATH 'Ch1Wfm1*Ch2Wfm1'`
Defines the multiplication of waveforms Ch1Wfm1 and Ch2Wfm1.

**Usage:** Asynchronous command

---

**CALCulate:MATH<m>:STATe** <State>

Enables the math waveform display.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Display" on page 281

---

**CALCulate:MATH<m>:ENVSelection** <EnvelopeCurve>

Selects the upper or lower part of the waveform for mathematic calculation, or a combination of both.

The setting is relevant for waveforms with waveform arithmetic mode "Envelope" or with "Peak detect" decimation. All mathematic operations - except for derivation - can be applied to envelope waveforms and waveforms with "Peak detect" decimation.

**Suffix:**
<m> 1..8
Selects the math waveform.

**Parameters:**
<EnvelopeCurve> MIN | MAX | BOTH
*RST: BOTH

**Usage:** Asynchronous command

**Firmware/software:** FW 2.25

**Manual operation:** See "Envelope selection" on page 281

---

**CALCulate:MATH<m>:ARIThmetics** <Arithmetics>

Selects the method to build the resulting math waveform from consecutive acquisitions. The processing is similar to the waveform arithmetics - instead of the acquired waveforms the results of the mathematic formula are used to create envelope, average and RMS.

To define the reset condition for arithmetics calculation, use `ACQuire:ARESet:MODE`.

**Suffix:**
<m> 1..8
Selects the math waveform.

**Parameters:**
<Arithmetics> OFF | ENVelope | AVERage | RMS | MINHold | MAXHold

Waveform arithmetic mode

**OFF**
The math waveform is built according to the mathematic formula.

**ENVelope**
Detects the minimum and maximum math values in a sample interval over a number of acquisitions.

**AVERage**
Calculates the average from the math data of the current acquisition and a number of acquisitions before. To define the number of acquisitions, use `ACQuire:COUNt`.

**RMS**
The resulting math waveform is the root mean square of the current acquisition and a number of acquisitions before. The result is the average power spectrum. Number of acquisitions: `ACQuire:COUNt`

**MAXHold**
Determines the maximum result for each input value from the math data of the current acquisition and a number of acquisitions before. To define the number of acquisitions, use `ACQuire:COUNt`.

**MINHold**
Determines the minimum result for each input value from the math data of the current acquisition and a number of acquisitions before. To define the number of acquisitions, use `ACQuire:COUNt`.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 297

---

**CALCulate:MATH<m>:VERTical:OFFSet** <VerticalOffset>

Sets a voltage offset to adjust the vertical position of the math function on the screen.

**Suffix:**
<m> 1..8
Math waveform

**Parameters:**

<VerticalOffset> Negative values move the waveform au, positive values move it down.

Range: -100E+12 to 100E+12
Increment: 0.01
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Vertical Offset" on page 296

---

**CALCulate:MATH<m>:VERTical:RANGe** <VerticalRange>

Defines the range of FFT values to be displayed.

**Suffix:**

<m> 1..8
Math waveform

**Parameters:**

<VerticalRange> Range: 0 to 1E+15
Increment: 0.01
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Vertical range" on page 418

---

**CALCulate:MATH<m>:VERTical:SCALe** <VerticalScale>

Defines the scale of the y-axis in the math function diagram. The value is defined as "V per division", e.g. *50V/div*. In this case, the horizontal grid lines are displayed in intervals of 50 V.

**Suffix:**

<m> 1..8
Math waveform

**Parameters:**

<VerticalScale> Range: 1E-12 to 100E+12
Increment: 10E-6
*RST: 0.5
Default unit: V/div

**Usage:** Asynchronous command

**Manual operation:** See " Vertical Scale" on page 296

---

**CALCulate:MATH<m>:VERTical:MAXimum** <VerticalMax>

Defines the maximum value on y-axis for spectrum displays.

**Suffix:**
<m> 1..8

**Parameters:**
<VerticalMax> Range: -1E+15 to 1E+15
Increment: 0.01
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Vertical maximum" on page 418

---

### CALCulate:MATH<m>:DATA:STYPe?

Returns the signal type of the source of the math waveform.

**Suffix:**
<m> 1..8
Selects the math waveform.

**Return values:**
<SignalType> SOUR | SPEC | CORR | MEAS | NONE
SOURce = normal signal
SPECtrum = FFT spectrum, specific math signal
CORRelation = correlated signal, specific math signal
MEAsurement = result of a measurement
NONE = undefined

**Usage:** Query only
Asynchronous command

---

### CALCulate:MATH<m>:DATA:HEADer?

Returns the header of math waveform data. The header contains attributes of the waveform.

***Table 24-5: Header data***

| Position | Meaning | Example |
|---|---|---|
| 1 | XStart in s | -9.477E-008 = - 94,77 ns |
| 2 | XStop in s | 9.477E-008 = 94,77 ns |
| 3 | Record length of the waveform in Samples | 200000 |
| 4 | Number of values per sample interval. For most waveforms the result is 1, for peak detect and envelope waveforms it is 2. If the number is 2, the number of returned values is twice the number of samples (record length). | 1 |

**Suffix:**
<m> 1..8
Selects the math waveform.

**Example:**
```
CALC:MATH1:DATA:HEAD?
-9.477E-008,9.477E-008,200000,1
```

**Usage:** Query only

---

**CALCulate:MATH<m>:DATA[:VALues]?**

Returns the data of the math waveform points for transmission from the instrument to the controlling computer. The data can be used in MATLAB, for example.

To set the export format, use `FORMat[:DATA]`.

You can retrieve only Y-values, or X- and Y-values. Use `EXPort:WAVeform:INCXvalues` to define this.

**Suffix:**
<m> 1..8
Selects the math waveform.

**Return values:**
<Data> List of values according to the format and content settings.

**Usage:** Query only

## 24.10.4 History

CHANnel<m>[:WAVeform<n>]:HISTory[:STATe] ..... 1405
ACQuire:AVAilable? ..... 1406
CHANnel<m>[:WAVeform<n>]:HISTory:CURRent ..... 1406
CHANnel<m>[:WAVeform<n>]:HISTory:STARt ..... 1406
CHANnel<m>[:WAVeform<n>]:HISTory:STOP ..... 1407
CHANnel<m>[:WAVeform<n>]:HISTory:TPACq ..... 1407
CHANnel<m>[:WAVeform<n>]:HISTory:PLAY ..... 1408
CHANnel<m>[:WAVeform<n>]:HISTory:REPLay ..... 1408
CHANnel<m>[:WAVeform<n>]:HISTory:TSDate? ..... 1408
CHANnel<m>[:WAVeform<n>]:HISTory:TSABsolute? ..... 1409
CHANnel<m>[:WAVeform<n>]:HISTory:TSRelative? ..... 1409
CHANnel<m>[:WAVeform<n>]:HISTory:TSRReference? ..... 1410

---

**CHANnel<m>[:WAVeform<n>]:HISTory[:STATe]** <State>

Enables or disables the history display.

**Suffix:**
<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Parameters:**

<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show history / Export history" on page 300

---

**ACQuire:AVAilable?**

Returns the number of acquisitions that is saved in the memory. This number of acquisitions is available for history viewing. It is also the number of acquisitions in a fast segmentation acquisition series.

**Return values:**

<AcquisitionCount> Range: 0 to 4294967295

**Usage:** Query only
Asynchronous command

---

**CHANnel<m>[:WAVeform<n>]:HISTory:CURRent** <CurrAcqIdx>

Accesses a particular acquisition in the memory to display it. The query returns the index of the segment that is shown.

**Suffix:**

<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Parameters:**

<CurrAcqIdx> History index: the newest segment has the index "0", older segments have a negative index: -(n-1), .... -1 , 0
where n is the number of acquired segments.
Range: 0 to -(n-1)
Increment: 1

**Example:**

```
CHAN2:HIST:STAT ON
CHAN2:HIST:CURR -1
*OPC
```

Displays the acquisition before last from the history.

**Usage:** Asynchronous command

**Manual operation:** See "Current acquisition" on page 301

---

**CHANnel<m>[:WAVeform<n>]:HISTory:STARt** <StrtAcqIdx>

Sets the index of the oldest history acquisition for the history viewing.

**Suffix:**

<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Parameters:**

<StrtAcqIdx> The start index is always negative.

Range: 0 to -(n-1)
Increment: 1

**Usage:** Asynchronous command

**Manual operation:** See "Start acquisition" on page 301

---

**CHANnel<m>[:WAVeform<n>]:HISTory:STOP** <StpAcqIdx>

Sets the index of the latest segment to be displayed in the history viewer.

**Suffix:**

<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Parameters:**

<StpAcqIdx> Index of the stop acquisition. The newest acquisition always has the index "0".

Range: 0 to -(n-1)
Increment: 1

**Usage:** Asynchronous command

**Manual operation:** See "Stop acquisition" on page 301

---

**CHANnel<m>[:WAVeform<n>]:HISTory:TPACq** <TimePerAcq>

Sets the display time for one acquisition. The shorter the time, the faster is the replay.

**Suffix:**

<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Parameters:**

<TimePerAcq>
Range: 40E-6 to 10
Increment: 1
*RST: 0.05
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Display time" on page 301

---

**CHANnel<m>[:WAVeform<n>]:HISTory:PLAY**

Starts and stops the replay of the history waveforms.

**Suffix:**

<m>
1..4
Selects the input channel.

<n>
1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Start export" on page 495

---

**CHANnel<m>[:WAVeform<n>]:HISTory:REPLay** <AutoRepeat>

If ON, the replay of the history waveform sequence repeats automatically. Otherwise, the replay stops at the stop index set with `CHANnel<m>[:WAVeform<n>]:HISTory:STOP`.

**Suffix:**

<m>
1..4
Selects the input channel.

<n>
1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Parameters:**

<AutoRepeat>
ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Auto repeat" on page 301

---

**CHANnel<m>[:WAVeform<n>]:HISTory:TSDate?**

Returns the date of the current acquisition that is shown in the history viewer (`CHANnel<m>[:WAVeform<n>]:HISTory:CURRent`).

**Suffix:**
<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Return values:**
<DateAbsString> String with date of the current acquisition (absolute time)

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Time stamp" on page 302

---

**CHANnel<m>[:WAVeform<n>]:HISTory:TSABsolute?**

Returns the abolsute daytime of the current acquisition that is shown in the history viewer (`CHANnel<m>[:WAVeform<n>]:HISTory:CURRent`).

**Suffix:**
<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Return values:**
<TimeAbsString> String containing the time and unit

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Time stamp" on page 302

---

**CHANnel<m>[:WAVeform<n>]:HISTory:TSRelative?**

Returns the relative time of the current acquisition - the time difference to the newest acquisition (index = 0).

See also: (`CHANnel<m>[:WAVeform<n>]:HISTory:CURRent`).

**Suffix:**
<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Return values:**
<TimeRelativ> Range: -100E+24 to 100E+24
Default unit: s

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Time stamp" on page 302

---

**CHANnel<m>[:WAVeform<n>]:HISTory:TSRReference?**

Returns the relative time of the currently selected acquisition and the internal reference time (horizontal alignment) in history view with respect to the acquisition with index 0.

**Suffix:**
<m> 1..4
Selects the input channel.

<n> 1..3
Selects the waveform. If [:WAVeform<n>] is omitted, waveform 1 is adressed.

**Return values:**
<TimeRelIntRef> Range: -100E+24 to 100E+24
Increment: 1
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

**Firmware/software:** Version 2.70

**Manual operation:** See "Time stamp" on page 302

### 24.10.5 XY-Diagram

WAVeform<m>:XYCurve:RATio ..... 1410
WAVeform<m>:XYCurve:STATe ..... 1411
WAVeform<m>:XYCurve:SWAP ..... 1411
LAYout:SIGNal:AXIS ..... 1411
WAVeform<m>:XYCurve:XSOurce ..... 1412
WAVeform<m>:XYCurve:YSOurce ..... 1412

---

**WAVeform<m>:XYCurve:RATio** <ConstantXYRatio>

If enabled, the x- and y-axes maintain a constant ratio in the diagram.

**Suffix:**
<m> 1..4
XY-diagram

**Parameters:**
<ConstantXYRatio> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Constant XY-ratio" on page 306

---

**WAVeform<m>:XYCurve:STATe** <State>

Activates an XY-waveform.

**Suffix:**
<m> 1..4
XY-diagram

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 305

---

**WAVeform<m>:XYCurve:SWAP**

Replaces the source of the x-axis with the source of the y-axis and vice versa.

**Suffix:**
<m> 1..4
XY-diagram

**Usage:** Event
Asynchronous command

**Manual operation:** See "Swap XY" on page 306

---

**LAYout:SIGNal:AXIS** <DiagramName>, <Source>, <XSource>

Creates an XY-diagram by adding a second waveform to a diagram.

**Setting parameters:**
<DiagramName> String with the name of the diagram where the waveform is added.

<Source> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | SG1TL1 | SG1TL2 | SG2TL1 | SG2TL2 | SG3TL1 | SG3TL2 | SG4TL1 | SG4TL2 | SG5TL1 | SG5TL2 | SG6TL1 | SG6TL2 | SG7TL1 | SG7TL2 | SG8TL1 | SG8TL2 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | SRTRack1 | SDTRack1 | TJTRack1 | DJTRack1 | DDJTrack1 | PJTRack1 | RJOTrack1 | SRTRack2 | SDTRack2 | TJTRack2 | DJTRack2 | DDJTrack2 | PJTRack2 | RJOTrack2 | TNTRack1 | DNTRack1 | DDNTrack1 | PNTRack1 | RNOTrack1

Waveform to be added, see Chapter 24.4.2, "Waveform parameter", on page 1193

<XSource> ON | OFF

If on, the added waveform is assigned to the x-axis.
If off, it is assigned to the y-axis.

**Usage:** Setting only
Asynchronous command

---

**WAVeform<m>:XYCurve:XSOurce** <XSource>

Defines the signal source that supplies the x-values of the XY-diagram.

**Suffix:**
<m> 1..4
XY-diagram

**Parameters:**
<XSource> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | SG1TL1 | SG1TL2 | SG2TL1 | SG2TL2 | SG3TL1 | SG3TL2 | SG4TL1 | SG5TL1 | SG5TL2 | SG6TL1 | SG6TL2 | SG7TL1 | SG7TL2 | SG8TL1 | SG8TL2 | SG4TL2 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2

Source of x-values, see Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "X-source" on page 305

---

**WAVeform<m>:XYCurve:YSOurce** <YSource>

Defines the signal source that supplies the y-values of the XY-diagram.

**Suffix:**
<m> 1..4
XY-diagram

**Parameters:**
<YSource> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | SG1TL1 | SG1TL2 | SG2TL1 | SG2TL2 | SG3TL1 | SG3TL2 | SG4TL1 | SG4TL2 | SG5TL2 | SG6TL1 | SG6TL2 | SG7TL1 | SG7TL2 | SG8TL1 | SG8TL2 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2

Source of y-values, see Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C2W1

**Usage:** Asynchronous command

**Manual operation:** See "Y-source" on page 306

## 24.11 Cursor measurements

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

24.11.1 Cursor setup....1413
24.11.2 Cursor results....1423
24.11.3 Peak search using cursors....1424

### 24.11.1 Cursor setup

CURSor<m>:STATe....1414
CURSor<m>:AOFF....1414
CURSor<m>:FUNCtion....1414
CURSor<m>:SOURce....1415
CURSor<m>:SSOurce....1416
CURSor<m>:USSource....1417

CURSor<m>:TRACking[:STATe] 1418
CURSor<m>:X1Position 1418
CURSor<m>:X2Position 1418
CURSor<m>:XCOupling 1419
CURSor<m>:Y1Position 1419
CURSor<m>:Y2Position 1420
CURSor<m>:YCOupling 1420
CURSor<m>:X1ENvelope 1420
CURSor<m>:X2ENvelope 1421
CURSor<m>:LABel[:LABel] 1421
CURSor<m>:LABel:X<1..2> 1422
CURSor<m>:LABel:Y<1..2> 1422
CURSor<m>:SIAD 1422

**CURSor<m>:STATe** <State>

Switches the indicated cursor on or off.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Cursor On/Off" on page 314

**CURSor<m>:AOFF**

This command switches all cursors off.

**Suffix:**
<m> The numeric suffix is irrelevant.

**Usage:** Event
Asynchronous command

**CURSor<m>:FUNCtion** <Type>

Defines the type of the indicated cursor set.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<Type> HORizontal | VERTical | PAIRed

**HORizontal**
A pair of horizontal cursor lines.

**VERTical**
A pair of vertical cursor lines.

**PAIRed**
Both vertical and horizontal cursor line pairs.

*RST: PAIRed

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 314

---

**CURSor<m>:SOURce** <Source>

Defines the source of the cursor measurement.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**

<Source> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | XY1 | XY2 | XY3 | XY4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | MSOB1 | MSOB2 | MSOB3 | MSOB4 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | SG1 | SG2 | SG3 | SG4 | SG5 | SG6 | SG7 | SG8 | SG1TL1 | SG1TL2 | SG2TL1 | SG2TL2 | SG3TL1 | SG3TL2 | SG4TL1 | SG4TL2 | SG5TL1 | SG5TL2 | SG6TL1 | SG6TL2 | SG7TL1 | SG7TL2 | SG8TL1 | SG8TL2 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | EYE1 | EYE2 | EYE3 | EYE4 | TDRZ | TDRH | TDRR | BATHtub1 | BMEas1 | SRESponse1 | SRTRack1 | SDTRack1 | TJTRack1 | DJTRack1 | DDJTrack1 | PJTRack1 | RJTRack1 | TJSPectrum1 | DDJSpectrum1 | PJSPectrum1 | RJSPectrum1 | BATHtub2 | BMEas2 | SRESponse2 | SRTRack2 | SDTRack2 | TJTRack2 | DJTRack2 | DDJTrack2 | PJTRack2 | RJOTRack2 | TJSPectrum2 | DDJSpectrum2 | PJSPectrum2 | RJOSpectrum2 | SRHist1 | SDHist1 | TJHist1 | DJHist1 | DDJHist1 | PJHist1 | RJOHist1 | SRHist2 | SDHist2 | TJHist2 | DJHist2 | DDJHist2 | PJHist2 | RJOHist2 | SRTRack2 | SDTRack2 | TJTRack2 | DJTRack2 | DDJTrack2 | PJTRack2 | RJOTrack2 | TJSPectrum2 | DDJSpectrum2 | PJSPectrum2 | RJOSpectrum2 | SEYE2 | NBAThtub1 | NBMeas1 | TNHist1 | DNHist1 | DDNHist1 | PNHist1 | RNOHist1 | TNTRack1 | DNTRack1 | DDNTrack1 | PNTRack1 | RNOTRack1 | TNSPectrum1 | DDNSpectrum1 | PNSPectrum1 | RNOSPectrum1 | NBAThtub2 | NBMeas2 | TNHist2 | DNHist2 | DDNHist2 | PNHist2 | RNOHist2 | TNTRack2 | DNTRack2 | DDNTrack2 | PNTRack2 | RNOTrack2 | TNSPectrum2 | DDNSpectrum2 | PNSPectrum2 | RNSPectrum2

Source of the cursor measurement, see Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 314

---

**CURSor<m>:SSOurce** <Source2>

Selects the second cursor source.

**Suffix:**

<m> 1..4
Selects the cursor set.

**Parameters:**

<Source2> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | XY1 | XY2 | XY3 | XY4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | MSOB1 | MSOB2 | MSOB3 | MSOB4 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | SG1 | SG2 | SG3 | SG4 | SG5 | SG6 | SG7 | SG8 | SG1TL1 | SG1TL2 | SG2TL1 | SG2TL2 | SG3TL1 | SG3TL2 | SG4TL1 | SG4TL2 | SG5TL1 | SG5TL2 | SG6TL1 | SG6TL2 | SG7TL1 | SG7TL2 | SG8TL1 | SG8TL2 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | EYE1 | EYE2 | EYE3 | EYE4 | TDRZ | TDRH | TDRR | BATHtub1 | BMEas1 | SRESponse1 | SRTRack1 | SDTRack1 | TJTRack1 | DJTRack1 | DDJTrack1 | PJTRack1 | RJOTRack1 | TJSPectrum1 | DDJSpectrum1 | PJOSPectrum1 | RJSPectrum1 | BATHtub2 | BMEas2 | SRESponse2 | SRTRack2 | SDTRack2 | TJTRack2 | DJTRack2 | DDJTrack2 | PJTRack2 | RJTORack2 | TJSPectrum2 | DDJSpectrum2 | PJSPectrum2 | RJOSPectrum2 | SEYE1 | SEYE2 | SDHist1 | TJHist1 | DJHist1 | DDJHist1 | PJHist1 | RJOHist1 | SRHist2 | SDHist2 | TJHist2 | DJHist2 | DDJHist2 | PJHist2 | RJOHist2 | NBAThtub1 | NBMeas1 | TNHist1 | DNHist1 | DDNHist1 | PNHist1 | RNOHist1 | TNTRack1 | DNTRack1 | DDNTrack1 | PNTRack1 | RNOTRack1 | TNSPectrum1 | DDNSpectrum1 | PNSPectrum1 | RNSPectrum1 | NBAThtub2 | NBMeas2 | TNHist2 | DNHist2 | DDNHist2 | PNHist2 | RNHist2 | TNTRack2 | DNTRack2 | DDNTrack2 | PNTRack2 | RNOTRack2 | TNSPectrum2 | DDNSpectrum2 | PNSPectrum2 | RNOSPectrum2

Second source of the cursor measurement, see Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C2W1

**Usage:** Asynchronous command

**Firmware/software:** FW 3.60

**Manual operation:** See "Second source, Source 2" on page 314

---

**CURSor<m>:USSource** <UseSource2>

Enables the second cursor source. To select the second source, use `CURSor<m>:SSOurce` on page 1416.

If enabled, the second corsor lines Cx.2 measure on the second source. Using a second source, you can measure differences between two channels with cursors.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<UseSource2> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Firmware/software:** FW 3.60

**Manual operation:** See "Second source, Source 2" on page 314

---

**CURSor<m>:TRACking[:STATe]** <TrackCurve>

If set to ON, the horizontal cursor lines follow the waveform.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<TrackCurve> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Track waveform" on page 315

---

**CURSor<m>:X1Position** <XPosition1>

Defines the position of the left vertical cursor line.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<XPosition1> Range: 0 to 500
Increment: 0.1
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "X1 position, X2 position" on page 314

---

**CURSor<m>:X2Position** <XPosition2>

Defines the position of the right vertical cursor line.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<XPosition2> Range: 0 to 500
Increment: 0.1
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "X1 position, X2 position" on page 314

---

**CURSor<m>:XCOupling** <Coupling>

Defines the positioning mode of the vertical cursor.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<Coupling> ON | OFF

**ON**
Moving one cursor line moves the other cursor line too. The cursor lines always remain a fixed distance.

**OFF**
Each cursor line is positioned independently.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Coupling (2 follows 1)" on page 316

---

**CURSor<m>:Y1Position** <YPosition1>

Defines the position of the lower horizontal cursor line.

If `CURSor<m>:TRACking[:STATe]` is enabled, the query returns the measurement result - the lower vertical value of the waveform.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<YPosition1> Range: -50 to 50
Increment: 0.01
*RST: 0
Default unit: The unit depends on the type of the waveform.

**Usage:** Asynchronous command

**Manual operation:** See "Y1 position, Y2 position" on page 314

---

**CURSor<m>:Y2Position** <YPosition2>

Defines the position of the upper horizontal cursor line.

If `CURSor<m>:TRACking[:STATe]` is enabled, the query returns the measurement result - the upper vertical value of the waveform.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<YPosition2> Range: -50 to 50
Increment: 0.01
*RST: 0
Default unit: The unit depends on the type of the waveform.

**Usage:** Asynchronous command

**Manual operation:** See "Y1 position, Y2 position" on page 314

---

**CURSor<m>:YCOupling** <Coupling>

Defines the positioning mode of the horizontal cursor. If the horizontal cursor lines track the waveform, the y-coupling is irrelevant (`CURSor<m>:MODE TRACk`).

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<Coupling> ON | OFF

**ON**
Moving one cursor line moves the other cursor line too. The cursor lines always remain a fixed distance.

**OFF**
Each cursor line is positioned independently.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Coupling (2 follows 1)" on page 316

---

**CURSor<m>:X1ENvelope** <EnvelopeCurve1>

If the waveform arithmetics are set to envelope curve (see `CHANnel<m>[:WAVeform<n>]:ARIThmetics` on page 1244) and `CURSor<m>:TRACking[:STATe]` is set to "ON", this setting defines how the first horizontal cursor is positioned.

**Suffix:**
<m> 1..*
Selects the cursor set. 4 cursor sets are available.

**Parameters:**
<EnvelopeCurve1> MIN | MAX

**MIN**
The horizontal cursor is set to the crossing point of the vertical cursor with the minimum waveform envelope.

**MAX**
The horizontal cursor is set to the crossing point of the vertical cursor with the maximum waveform envelope.

*RST: MIN

**Usage:** Asynchronous command

**Manual operation:** See "Envelope selection 1|2" on page 315

---

**CURSor<m>:X2ENvelope** <EnvelopeCurve2>

If the waveform arithmetics are set to envelope curve (see `CHANnel<m>[:WAVeform<n>]:ARIThmetics` on page 1244) and `CURSor<m>:TRACking[:STATe]` is set to "ON", this setting defines how the second horizontal cursor is positioned.

**Suffix:**
<m> 1..*
Selects the cursor set. 4 cursor sets are available.

**Parameters:**
<EnvelopeCurve2> MIN | MAX

**MIN**
The horizontal cursor is set to the crossing point of the vertical cursor with the minimum waveform envelope.

**MAX**
The horizontal cursor is set to the crossing point of the vertical cursor with the maximum waveform envelope.

*RST: MAX

**Usage:** Asynchronous command

**Manual operation:** See "Envelope selection 1|2" on page 315

---

**CURSor<m>:LABel[:LABel]** <ShowLabel>

Shows the cursor labels in the diagram.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<ShowLabel> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Show label" on page 316

---

**CURSor<m>:LABel:X<1..2>** <Name>

Defines a label to be displayed with the vertical cursors. X1 is the left cursor line, and X2 the right one. By default, the cursors are labeled as Cu1.X1, Cu1.X2, Cu2.X1, Cu2.X2, Cu3.X1, Cu3.X2, Cu4.X1, Cu4.X2.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<Name> String with the cursor label

**Usage:** Asynchronous command

**Manual operation:** See "X1 label, X2 label" on page 317

---

**CURSor<m>:LABel:Y<1..2>** <Name>

Defines a label to be displayed with the vertical cursors. Y1 is the lower cursor line, and Y2 the upper one. By default, the cursors are labeled as Cu1.Y1, Cu1.Y2, Cu2.Y1, Cu2.Y2, Cu3.Y1, Cu3.Y2, Cu4.Y1, Cu4.Y2.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<Name> String with the cursor label

**Usage:** Asynchronous command

**Manual operation:** See "Y1 label, Y2 label" on page 317

---

**CURSor<m>:SIAD** <ShwInAllDiags>

Shows the enabled cursor measurements in all active diagrams of the same domain (time/ spectrum).

**Suffix:**
<m> 1..*
The suffix is irrelevant. All cursor measurements are affected.

**Parameters:**
<ShowInAllDiagr> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Show in all diagrams" on page 316

### 24.11.2 Cursor results

CURSor<m>:XDELta[:VALue]?....1423
CURSor<m>:XDELta:INVerse?....1423
CURSor<m>:YDELta[:VALue]?....1423
CURSor<m>:YDELta:SLOPe....1424

**CURSor<m>:XDELta[:VALue]?**

Queries the delta value (distance) of two vertical cursor lines.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Return values:**
<Delta> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

**CURSor<m>:XDELta:INVerse?**

Queries the inverse value of the delta value (distance) of the two vertical cursor lines.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Return values:**
<DeltaInverse> Range: -100E+24 to 100E+24
*RST: 0
Default unit: Hz

**Usage:** Query only
Asynchronous command

**CURSor<m>:YDELta[:VALue]?**

Queries the delta value (distance) of the two horizontal cursor lines.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Return values:**
<Delta> Range: -100E+24 to 100E+24
*RST: 0

**Usage:** Query only
Asynchronous command

---

**CURSor<m>:YDELta:SLOPe** <DeltaSlope>

Returns the inverse value of the voltage difference - the reciprocal of the vertical distance of two horizontal cursor lines: *1/ΔV*.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Parameters:**
<DeltaSlope> Range: -100E+24 to 100E+24
Increment: 0
*RST: 0

**Usage:** Asynchronous command

### 24.11.3 Peak search using cursors

CURSor<m>:FFT:SETCenter ..... 1424
CURSor<m>:FFT:TOCenter ..... 1424
CURSor<m>:MAXimum[:PEAK] ..... 1425
CURSor<m>:MAXimum:LEFT ..... 1425
CURSor<m>:MAXimum:RIGHt ..... 1425
CURSor<m>:MAXimum:NEXT ..... 1425

---

**CURSor<m>:FFT:SETCenter**

Sets the center frequency to the frequency value that is measured at cursor line c1.

**Suffix:**
<m> The suffix is irrelevant.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Set center frequency to Cursor" on page 318

---

**CURSor<m>:FFT:TOCenter**

Sets the vertical cursor line c1 to the center frequency.

**Suffix:**
<m> The suffix is irrelevant.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Center frequency" on page 318

---

**CURSor<m>:MAXimum[:PEAK]**

Sets both cursors to the absolute peak value.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Usage:** Event

**Manual operation:** See "Absolute" on page 318

---

**CURSor<m>:MAXimum:LEFT**

Sets cursor 2 to the next maximum to the left of the current position.

**Suffix:**
<m> 1..4
Selects the cursor set.

**Usage:** Event

**Manual operation:** See "Next left" on page 318

---

**CURSor<m>:MAXimum:RIGHt**

Sets cursor 2 to the next peak to the right (from the current position).

**Suffix:**
<m> 1..4
Selects the cursor set.

**Usage:** Event

**Manual operation:** See "Next right" on page 318

---

**CURSor<m>:MAXimum:NEXT**

Sets cursor 2 to the next smaller peak (from the current position).

**Suffix:**
<m> 1..4
Selects the cursor set.

**Usage:** Event

**Manual operation:** See "Next absolute" on page 318

## 24.12 Automatic measurements

This chapter contains all remote commands to set up automatic measurements and to analyze the measurement results.

**Selection of the measurement group: MEASurement<m>**

With R&S RTP, you can configure up to eight simultaneous measurement groups. In manual operation, these eight measurement groups are represented by subtabs "MG1" to "MG8" in the "Measurements" dialog box. For remote operation, the measurement group is indicated by the suffix MEAS<m>, containing the number of the measurement group.

| Remote control: measurement suffix <m> | Manual operation: "MG" subtab |
|---|---|
| 1 to 8 correspond to | "MG1" to "MG8" |
| 9, 10: do not use | Not available |

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

- General settings....1427
- Results....1432
- Amplitude/time measurements....1436
- Arithmetic for amplitude/time and spectrum measurements....1447
- Eye diagram measurements....1456
- Spectrum....1466
- Optical measurements....1473
- Histograms....1476
- Display....1487
- Statistics and long-term measurements....1488
- Protocol....1495
- Track....1502
- Gating....1503
- Limit check and event actions....1506
- Reference levels....1509

### 24.12.1 General settings

MEASurement<m>[:ENABle] ..... 1427
MEASurement<m>:SOURce ..... 1427
MEASurement<m>:FSRC ..... 1429
MEASurement<m>:SSRC ..... 1429
MEASurement<m>:CATegory ..... 1429
MEASurement<m>:MAIN ..... 1430
MEASurement<m>:ADDitional ..... 1431
MEASurement<m>:AON ..... 1432
MEASurement<m>:AOFF ..... 1432

**MEASurement<m>[:ENABle]** <State>

Switches the indicated measurement on or off.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<State> ON | OFF
*RST: OFF

**Example:** See "Simple frequency and amplitude measurement" on page 1175.

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 324

**MEASurement<m>:SOURce** <SignalSource>, [<SignalSource2>]

Defines the source of the measurement. Availability of sources depends on the selected category and installed options.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**

<SignalSource> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | SBUS1 | SBUS2 | SBUS3 | SBUS4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | SG1TL1 | SG1TL2 | SG2TL1 | SG2TL2 | SG3TL1 | SG3TL2 | SG4TL1 | SG4TL2 | SG5TL1 | SG5TL2 | SG6TL1 | SG6TL2 | SG7TL1 | SG7TL2 | SG8TL1 | SG8TL2 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | | SRESponse1 | SRTRack1 | SDTRack1 | TJTRack1 | DJTRack1 | DDJTrack1 | PJTRack1 | RJTRack1 | SRESponse2 | SRTRack2 | SDTRack2 | TJTRack2 | DJTRack2 | DDJTrack2 | PJTRack2 | RJTRack2 | SEYE1 | SEYE2 | TNTRack1 | DNTRack1 | DDNTrack1 | PNTRack1 | RNOTrack1 | TNTRack2 | DNTRack2 | DDNTrack2 | PNTRack2 | RNOTrack2 | PJDTrack1 | PJDTrack2 | PNDTrack1 | PNDTrack2

First source of the measurement, see Chapter 24.4.2, "Waveform parameter", on page 1193

<SignalSource2> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | | SRESponse1 | SRTRack1 | SDTRack1 | TJTRack1 | DJTRack1 | DDJTrack1 | PJTRack1 | RJTRack1 | SRESponse2 | SRTRack2 | SDTRack2 | TJTRack2 | DJTRack2 | DDJTrack2 | PJTRack2 | RJTRack2 | SEYE1 | SEYE2 | TNTRack1 | DNTRack1 | DDNTrack1 | PNTRack1 | RNOTrack1 | TNTRack2 | DNTRack2 | DDNTrack2 | PNTRack2 | RNOTrack2 | PJDTrack1 | PJDTrack2 | PNDTrack1 | PNDTrack2

Second measurement source. The 2nd source is required for amplitude/time measurements that are performed on two waveforms (e.g. delay, phase).
Digital channels are only available if <SignalSource> is a digital channel.
Step response, advanced jitter track components and synthetic eye sources are available for amplitude/time measurements only.

*RST: C1W1,C2W1

**Example:** See "Simple frequency and amplitude measurement" on page 1175.

**Manual operation:** See "Source, Source 2" on page 324

**MEASurement<m>:FSRC** <Source>

Defines the first measurement source.

The command is an alternative to `MEASurement<m>:SOURce`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Source> Source of the measurement, see `MEASurement<m>:SOURce` on page 1427.
*RST: C1W1

**Usage:** Asynchronous command

**MEASurement<m>:SSRC** <Source2>

Defines the second measurement source.

The command is an alternative to `MEASurement<m>:SOURce`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Source2> Second source of the measurement, see `MEASurement<m>:SOURce` on page 1427.
Digital channels are only available if a digital channel is set as first measurement source using `MEASurement<m>:FSRC`.
*RST: C2W1

**Usage:** Asynchronous command

**MEASurement<m>:CATegory** <Category>

Defines the measurement category.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Category> AMPTime | JITTer | EYEJitter | SPECtrum | HISTogram | PROTocol | OPTical

**AMPTime**
Amplitude and time measurements

**JITTer**
Jitter measurements, available if one of the jitter options is installed.

**EYEJitter**
Eye diagram measurements

**SPECtrum**
Spectrum measurements

**HISTogram**
Histogram measurements

**PROTocol**
Protocol measurements

**OPTical**
Optical measurements

*RST: AMPTime

**Example:** See "Creating and reading histograms" on page 1176

**Usage:** Asynchronous command

**Manual operation:** See "Category" on page 324

---

**MEASurement<m>:MAIN** <MeasType>

Defines the measurement that is used as a source for math calculations and result analysis. This measurement is mandatory in the measurement group. Further measurements are added to the measurement group with `MEASurement<m>:ADDitional`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<MeasType> **Amplitude/time measurements**
HIGH | LOW | AMPLitude | MAXimum | MINimum | PDELta | MEAN | RMS | STDDev | POVershoot | NOVershoot | AREA | RTIMe | FTIMe | PPULse | NPULse | PERiod | FREQuency | PDCYcle | NDCYcle | CYCarea | CYCMean | CYCRms | CYCStddev | PULCnt | DELay | PHASe | BWIDth | PSWitching | NSWitching | PULSetrain | EDGecount | SHT | SHR | DTOTrigger | PROBemeter | SLERising | SLEFalling | ATAR
See Chapter 24.12.3, "Amplitude/time measurements", on page 1436.
*RST value for amplitude/time measurements: `AMPLitude`.

**Jitter measurements**
CCJitter | NCJitter | CCWidth | CCDutycycle | TIE | UINTerval | DRATe | SKWDelay | SKWPhase
Only available if option R&S RTP-K12 is installed. See Chapter 24.22.1, "Jitter measurements (option R&S RTP-K12)", on page 2374.

**Eye diagram measurements**
ERPercent | ERDB | EHEight | EWIDth | ETOP | EBASe | QFACtor | RMSNoise | SNRatio | DCDistortion | ERTime | EFTime | EBRate | EAMPlitude | PPJitter | STDJitter | RMSJitter
See Chapter 24.12.5, "Eye diagram measurements", on page 1456
*RST value for eye/jitter measurements: `ERPercent`.

**Spectrum measurements**
CPOWer | OBWidth | SBWidth | THD | THDPCT | THDA | THDU | THDR | PLISt | HAR | THDF | SPAR
See Chapter 24.12.6, "Spectrum", on page 1466.
*RST value for spectrum measurements: `CPOWer`.

**Histogram measurements**
WCOunt | WSAMples | HSAMples | HPEak | PEAK | UPEakvalue | LPEakvalue | HMAXimum | HMINimum | MEDian | MAXMin | HMEan | HSTDdev | M1STddev | M2STddev | M3STddev | MKPositive | MKNegative
See Chapter 24.12.8.2, "Histogram measurement", on page 1482.
*RST value for histogram measurements: `WCOunt`.

**Protocol measurements**
NONE | AUDio | F2F | F2A | FLDValue | GAP | MBITrate | SBITrate | BIDLe | FCNT | FEC | FER | CFER
Requires option R&S RTP-K35
See Chapter 24.12.11, "Protocol", on page 1495

**Optical measurements**
OAP | OMA
See Chapter 24.12.7, "Optical measurements", on page 1473.

**Example:** See "Simple frequency and amplitude measurement" on page 1175.
See "Creating and reading histograms" on page 1176

**Usage:** Asynchronous command

**Manual operation:** See "Active measurements, Add / Remove" on page 324

---

**MEASurement<m>:ADDitional** <MeasType>, <State>
**MEASurement<m>:ADDitional?** <MeasType>

Adds additional measurements to the indicated measurement group. Only one measurement can be enabled or disabled per command. The query returns the state of the specified measurement.

Note that each measurement group must contain measurements from the same category. For example, if the main measurement of measurement group 1 is amplitude, then you cannot enable an eye width measurement in the same measurement group.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<State> ON | OFF
Enables or disables the measurement.

**Parameters for setting and query:**
<MeasType> See `MEASurement<m>:MAIN` on page 1430.

**Usage:** Asynchronous command

**Manual operation:** See "Active measurements, Add / Remove" on page 324

---

**MEASurement<m>:AON**

Enables all additional measurements in all categories of the indicated measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Usage:** Event
Asynchronous command

---

**MEASurement<m>:AOFF**

Disables all additional measurements in all categories of the indicated measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Usage:** Event
Asynchronous command

## 24.12.2 Results

MEASurement<m>:ARES? 1433
MEASurement<m>:ARNames 1434
MEASurement<m>:RESult[:ACTual]? 1434
MEASurement<m>:RESult:AVG? 1434
MEASurement<m>:RESult:EVTCount? 1434
MEASurement<m>:RESult:NPEak? 1434

MEASurement<m>:RESult:PPEak? ..... 1434
MEASurement<m>:RESult:RELiability? ..... 1434
MEASurement<m>:RESult:RMS? ..... 1434
MEASurement<m>:RESult:WFMCount? ..... 1434
MEASurement<m>:RESult:STDDev? ..... 1434
MEASurement<m>:RESult:STARt? ..... 1436
MEASurement<m>:RESult:STOP? ..... 1436
MEASurement<m>:RESult:COUNt? ..... 1436

### MEASurement<m>:ARES?

Returns the results of all active measurements for the selected measurement group. If statistics are enabled, the instrument returns also statistical results.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Return values:**

<sData> Result string

**Example:**

```
:MEASurement:ARES?
2.351778656126e-001,1.343873517787e-001,
-1.343873517787e-001
```

**Example:**

```
:MEASurement:ARNames 1
:MEASurement:ARES?
Amplitude: 2.351778656126e-001,Max:
1.343873517787e-001,Min: -1.343873517787e-001
```

**Example:**

```
:MEASurement:ARNames 1
:MEASurement:STATistics:ENABle 1
:MEASurement:ARES?
Amplitude:
2.312252964427e-001,2.608695652174e-001,
2.292490118577e-001,2.391822576775e-001,
2.392462568683e-001,5.533663458383e-003,12447,
12447,Max: 1.343873517787e-001,
1.343873517787e-001,1.324110671937e-001,
1.340297640714e-001, 1.340319233732e-001,
7.608360495310e-004,12447,12447, Min:
-1.324110671937e-001,-1.324110671937e-001,
-1.343873517787e-001,-1.340169449387e-001,
1.340191641476e-001,7.712812700636e-004,12447,
12447
Results: current, peak+, peak-, average, RMS,
standard deviation, event count, waveform count
```

**Usage:**

Query only
Asynchronous command

**MEASurement<m>:ARNames** <Identifier>

Enables a prefix that indicates the measurement in the result string of the `MEASurement<m>:ARES?` command.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Identifier> ON | OFF
*RST: OFF

**Example:**
```
:MEASurement:ARNames 1
:MEASurement:ARES?
Low: -1.185770750988e-001,Amplitude:
2.371541501976e-001,Max: 1.343873517787e-001
```

**Usage:** Asynchronous command

**MEASurement<m>:RESult[:ACTual]?** [<MeasType>]
**MEASurement<m>:RESult:AVG?** [<MeasType>]
**MEASurement<m>:RESult:EVTCount?** [<MeasType>]
**MEASurement<m>:RESult:NPEak?** [<MeasType>]
**MEASurement<m>:RESult:PPEak?** [<MeasType>]
**MEASurement<m>:RESult:RELiability?** [<MeasType>]
**MEASurement<m>:RESult:RMS?** [<MeasType>]
**MEASurement<m>:RESult:WFMCount?** [<MeasType>]
**MEASurement<m>:RESult:STDDev?** [<MeasType>]

Return the statistic results of the specified measurement. If no parameter is specified, the result of the main measurement is returned. The main measurement is defined using `MEASurement<m>:MAIN`.

- [:ACTual]: current measurement result
- AVG: average of the long-term measurement results
- EVTCount: number of measurement results in the long-term measurement
- NPEak: negative peak value of the long-term measurement results
- PPEak: positive peak value of the long-term measurement results
- RELiability: reliability of the measurement result
- RMS: RMS value of the long-term measurement results
- STDDev: standard deviation of the long-term measurement results

For a detailed description of the results, see "Selection of the measurement group: MEASurement<m>" on page 1426.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Query parameters:**

<MeasType>

**Amplitude/time measurements**
HIGH | LOW | AMPLitude | MAXimum | MINimum | PDELta | MEAN | RMS | STDDev | POVershoot | NOVershoot | AREA | RTIMe | FTIMe | PPULse | NPULse | PERiod | FREQuency | PDCYcle | NDCYcle | CYCarea | CYCMean | CYCRms | CYCStddev | PULCnt | DELay | PHASe | BWIDth | PSWitching | NSWitching | PULSetrain | EDGecount | SETup | HOLD | SHR | DTOTrigger | PROBemeter | SLERising | SLEFalling | ATAR
See Chapter 24.12.3, "Amplitude/time measurements", on page 1436.

**Jitter measurements (option R&S RTP-K12)**
CCJitter | NCJitter | CCWidth | CCDutycycle | TIE | UINTerval | DRATe | SKWDelay | SKWPhase
See Chapter 24.22.1, "Jitter measurements (option R&S RTP-K12)", on page 2374

**Eye diagram measurements**
ERPercent | ERDB | EHEight | EWIDth | ETOP | EBASe | QFACtor | RMSNoise | SNRatio | DCDistortion | ERTime | EFTime | EBRate | EAMPlitude | PPJitter | STDJitter | RMSJitter | CPPercent
See Chapter 24.12.5, "Eye diagram measurements", on page 1456.

**Spectrum measurements**
CPOWer | OBWidth | SBWidth | THD | THDPCT | THDA | THDU | THDR | PLISt | HAR | THDF | SPAR
See Chapter 24.12.6, "Spectrum", on page 1466.

**Histogram measurements**
WCOunt | WSAMples | HSAMples | HPEak | PEAK | UPEakvalue | LPEakvalue | HMAXimum | HMINimum | MEDian | MAXMin | HMEan | HSTDdev | M1STddev | M2STddev | M3STddev | MKPositive | MKNegative
See Chapter 24.12.8.2, "Histogram measurement", on page 1482.

**Protocol measurements**
NONE | AUDio | F2F | T2F | FLDValue | GAP | MBITrate | SBITrate | BIDLe | FCNT | FEC | FER | CFER
Requires option R&S RTP-K5 for type AUDio, or option R&S RTP-K35.

**Optical measurements**
OAP | OMA
See Chapter 24.12.7, "Optical measurements", on page 1473.

**Example:** See "Simple frequency and amplitude measurement" on page 1175.
See "Creating and reading histograms" on page 1176

**Usage:** Query only

---

**MEASurement<m>:RESult:STARt?** [<MeasType>]
**MEASurement<m>:RESult:STOP?** [<MeasType>]

Return the start and stop times of the specified measurement. The parameter defines the measurement. If no parameter is specified, the result of the main measurement is returned. The main measurement is defined using `MEASurement<m>:MAIN`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Query parameters:**
<MeasType> See `MEASurement<m>:RESult:STDDev?` on page 1434.

**Usage:** Query only

---

**MEASurement<m>:RESult:COUNt?**

Returns the number of result groups that are returned by `MEASurement:RESult:ACTual? HAR`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Query parameters:**
<MeasType> HAR
The command is only relevant for harmonic search.

**Return values:**
Number Number of result groups

**Example:**
```
:MEASurement:RESult:ACTual? HAR
99.9;-6.000139236;199.8;-80.701713562;299.7;-15.528377533;
:MEASurement:RESult:COUNt? HAR
4
```

**Usage:** Query only

**Firmware/software:** Version 2.70

### 24.12.3 Amplitude/time measurements

The following table lists the <MeasType> parameter values with a short description.

In addition, the suffixes <n> for limit-check remote commands are listed.

For a detailed description, see Chapter 8.2.5.1, "Overview of amplitude/time measurements", on page 341.

***Table 24-6: Amplitude and time measurements***

| <MeasType> parameter value (remote control) | Measurement (manual operation) | Description, result | Suffix <n> |
|---|---|---|---|
| HIGH | High | High signal level | 1 |
| LOW | Low | Low signal level | 2 |
| AMPLitude | Amplitude | Amplitude of the signal | 3 |
| MAXimum | Max | Maximum value of the waveform | 4 |
| MINimum | Min | Minimum value of the waveform | 5 |
| PDELta | Peak to peak | Peak-to-peak value of the waveform | 6 |
| MEAN | Mean | Mean value of the waveform | 7 |
| RMS | RMS | RMS (Root Mean Square) value of the voltage | 8 |
| STDDev | σ (S-dev) | Standard deviation of the waveform | 9 |
| POVershoot | Pos. overshoot | Positive overshoot of a square wave | 10 |
| NOVershoot | Neg. overshoot | Negative overshoot of a square wave | 11 |
| AREA | Area | Area beneath the waveform (integral) | 12 |
| RTIMe | Rise time | Rise time of the left-most rising edge of the waveform. | 13 |
| FTIMe | Fall time | Falling time of the left-most falling edge of the waveform. | 14 |
| PPULse | Pos. pulse | Width of a positive pulse – a rising edge followed by a falling edge. The measurement requires at least one complete period of a triggered signal. | 15 |
| NPULse | Neg. pulse | Width of a negative pulse – a falling edge followed by a rising edge. The measurement requires at least one complete period of a triggered signal. | 16 |
| PERiod | Period | Length of the left-most signal period of the waveform | 17 |
| FREQuency | Frequency | Frequency of the signal. The result is based on the period measurement. | 18 |
| PDCYcle | Pos. duty cycle | Positive duty cycle. The measurement requires at least one complete period of a triggered signal. | 19 |
| NDCYcle | Neg. duty cycle | Negative duty cycle. The measurement requires at least one complete period of a triggered signal. | 20 |
| CYCarea | Cycle area | Area (integral) beneath one cycle | 21 |
| CYCMean | Cycle mean | Mean value of one cycle | 22 |
| CYCRms | Cycle RMS | The RMS (Root Mean Square) value of one cycle | 23 |
| CYCStddev | Cycle σ (S-dev) | Standard deviation of one cycle | 24 |

| <MeasType> parameter value (remote control) | Measurement (manual operation) | Description, result | Suffix <n> |
|---|---|---|---|
| PULCnt | Pulse count | Number of positive or negative pulses of the waveform, or both | 25 |
| DELay | Delay | Time difference between any edges of two measurement sources at any reference level. The measurement result is negative if the edge of the second source comes before the edge of the first source. | 26 |
| PHASe | Phase | Phase difference between two waveforms | 27 |
| BWIDth | Burst width | Duration of one burst, measured from the first edge to the last | 28 |
| PSWitching | Pos. switching | Settling time at rising edges | 29 |
| NSWitching | Neg. switching | Settling time at falling edges | 30 |
| PULSetrain | Pulse train | Duration of N positive pulses, measured from the rising edge of the first pulse to the falling edge of the N-th pulse. N has to be configured. | 31 |
| EDGecount | Edge count | Number of positive or negative edges of the waveform, or both | 32 |
| SETup | Setup time | Parameters to query the setup and hold times.<br>Use these parameters only in following queries:<br>• `MEASurement<m>:ARES?`<br>• `MEASurement<n>:RESult:..` commands | 33 |
| HOLD | Hold time | | |
| SHT | Setup/Hold time | Setting parameter to enable Setup/Hold time measurements.<br>Use this parameter only as setting in:<br>• `MEASurement<m>:MAIN` on page 1430<br>• `MEASurement<m>:ADDitional` on page 1431 | 35 |
| SHR | Setup/Hold ratio | Setup/Hold ratio measurement. Setup/Hold ratio is the ratio of the setup time to the sum of hold and setup time:<br>$T_{Setup} / (T_{Setup} + T_{Hold})$<br>Use this parameter as setting to activate the Setup/Hold ratio measurement in:<br>• `MEASurement<m>:MAIN` on page 1430<br>• `MEASurement<m>:ADDitional` on page 1431<br>It is also used in the following queries:<br>• `MEASurement<m>:ARES?`<br>• `MEASurement<n>:RESult:..`<br>Used also in queries with and commands. | 36 |
| Used for jitter measurements (limit checks) see Chapter 24.22.1, "Jitter measurements (option R&S RTP-K12)", on page 2374. | | | 37 to 45 |
| DTOTrigger | Delay to trigger | Time between the trigger event and a following signal slope. High accuracy even if the trigger event is outside the acquisition data. | 46 |
| PROBemeter | Trig. ProbeMeter | DC voltage measured by the connected active R&S probe | 47 |
| SLERising | Slew rate on rising edge | Steepness of the rising edge: voltage difference between the lower and higher reference level, divided by the rise time. | 48 |

| <MeasType> parameter value (remote control) | Measurement (manual operation) | Description, result | Suffix <n> |
|---|---|---|---|
| SLEFalling | Slew rate on falling edge | Steepness of the falling edge: voltage difference between the higher and lower reference level, divided by the fall time. | 49 |
| ATAR | Arithmetic for amplitude/time measurements | Basic arithmetic calculations with measurement values and scalars. | 50 |

MEASurement<m>:ENVSelect ..... 1439
MEASurement<m>:DETThreshold ..... 1440
MEASurement<m>:AMPTime:ALEVel ..... 1440
MEASurement<m>:AMPTime:PFSLope ..... 1440
MEASurement<m>:AMPTime:PSLope ..... 1441
MEASurement<m>:AMPTime:DELay<n>:DIRection ..... 1441
MEASurement<m>:AMPTime:DELay<n>:ECOunt ..... 1442
MEASurement<m>:AMPTime:DELay<n>:LSELect ..... 1442
MEASurement<m>:AMPTime:DELay<n>:SLOPe ..... 1443
MEASurement<m>:AMPTime:PTCount ..... 1443
MEASurement<m>:AMPTime:ESLope ..... 1444
MEASurement<m>:AMPTime:CSLope ..... 1444
MEASurement<m>:AMPTime:CLCK<n>:LSELect ..... 1444
MEASurement<m>:AMPTime:DATA<n>:LSELect ..... 1445
MEASurement<m>:AMPTime:DTOTrigger<n>:SLOPe ..... 1445
MEASurement<m>:AMPTime:DTOTrigger<n>:LSELect ..... 1445
MEASurement<m>:AMPTime:LCHeck<n>:VALid ..... 1446
MEASurement<m>:AMPTime:LCHeck<n>:LOWer:LIMit ..... 1446
MEASurement<m>:AMPTime:LCHeck<n>:UPPer:LIMit ..... 1446
MEASurement<m>:AMPTime:LCHeck<n>:LOWer:MARGin ..... 1447
MEASurement<m>:AMPTime:LCHeck<n>:UPPer:MARGin ..... 1447

**MEASurement<m>:ENVSelect** <EnvelopeCurve>

The command is only relevant for measurements on envelope waveforms. It selects the envelope to be used for measurement.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**

<EnvelopeCurve> MIN | MAX | BOTH

MIN: measures on the lower envelope
MAX: measures on the upper envelope
BOTH: the envelope is ignored and the waveform measured as usual

*RST: BOTH

**Usage:** Asynchronous command

**Manual operation:** See "Envelope" on page 325

---

**MEASurement<m>:DETThreshold** <SignDetectThres>

The measured signal value must reach a certain voltage or bit to be included in the measurement algorithm. This minimum value is defined in percent of the signal amplitude (difference of high and low signal levels). Values below this value are considered to be noise and they are ignored. The setting is relevant for area, time, and counting measurements.

**Suffix:**
<m> 1..10
irrelevant

**Parameters:**
<SignDetectThres> Range: 0 to 50
Increment: 1
*RST: 5
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Signal threshold" on page 347

---

**MEASurement<m>:AMPTime:ALEVel** <AreaLevel>

Defines the reference level used to integrate the waveform.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<AreaLevel> Range: -100E+24 to 100E+24
Increment: 0
*RST: 0
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Area level" on page 347

---

**MEASurement<m>:AMPTime:PFSLope** <PeriodSlope>

Selects the slope direction for frequency and period measurements.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<PeriodSlope> FIRSt | POSitive | NEGative | EITHer

**POSitive | NEGative**
Measures the time between rising or falling edges, respectively.

**EITHer**
In multiple measurements, the time is measured both between rising edges and between falling edges.
In single measurements. the first edge is taken for the measurement.

**FIRSt**
Time is measured either between rising edges or between falling edges. The first edge is taken for the measurement. In single measurements, it works the same way as "Either".
Only available for analog measurement sources.

*RST: FIRSt (analog source), POSitive (digital source)

**Usage:** Asynchronous command

**Manual operation:** See "Measured slope" on page 348

---

**MEASurement<m>:AMPTime:PSLope** <PulsesSlope>

Sets the first slope of the pulses to be counted. The setting is only relevant for pulse count measurement (`MEASurement<m>:MAIN PULCnt` or `MEASurement<m>:ADDitional PULCnt,ON`).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<PulsesSlope> POSitive | NEGative | EITHer

Count either positive or negative pulses, or both.

*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Pulses slope" on page 348

---

**MEASurement<m>:AMPTime:DELay<n>:DIRection** <EdgeCntDirct>

Selects the direction for counting slopes for each source: from the beginning of the waveform, or from the end.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
Selects the source number.

**Parameters:**
<EdgeCntDirct> FRFI | FRLA

FRFI - FRom FIrst, counting starts with the first edge of the waveform.
FRLA - FRom LAst, counting starts with the last edge of the waveform.

*RST: FRFI

**Usage:** Asynchronous command

**Manual operation:** See "Settings for delay and phase measurements (analog sources)" on page 348

---

**MEASurement<m>:AMPTime:DELay<n>:ECOunt** <EdgeIndex>

Sets the number of the edge that is relevant for delay measurement for each source.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
Selects the source number.

**Parameters:**
<EdgeIndex> Edge number

Range: 1 to 100000
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Settings for delay and phase measurements (analog sources)" on page 348

---

**MEASurement<m>:AMPTime:DELay<n>:LSELect** <DlyLevSel>

Selects the reference level on which the time is measured for each source.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
Selects the source number.

**Parameters:**
<DlyLevSel> UPPer | MIDDle | LOWer
*RST: MIDDle

**Usage:** Asynchronous command

**Manual operation:** See "Settings for delay and phase measurements (analog sources)" on page 348

---

**MEASurement<m>:AMPTime:DELay<n>:SLOPe** <Slope>

Sets the edge of each source, between which the delay is measured.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
Selects the source number.

**Parameters:**
<Slope> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Settings for delay and phase measurements (analog sources)" on page 348

---

**MEASurement<m>:AMPTime:PTCount** <PulseCount>

Sets the number of positive pulses for the pulse train measurement. It measures the duration of N positive pulses from the rising edge of the first pulse to the falling edge of the N-th pulse.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<PulseCount> Range: 1 to 2147483647
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Pulse count" on page 351

**MEASurement<m>:AMPTime:ESLope** <EdgesSlope>

Sets the edge direction to be counted: rising edges, falling edges, or both. The setting is only relevant for edge count measurement (`MEASurement<m>:MAIN EDGecount` or `MEASurement<m>:ADDitional EDGecount,ON`).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<EdgesSlope> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Edges slope" on page 351

**MEASurement<m>:AMPTime:CSLope** <SetHoldClkSlp>

Sets the edge of the clock from which the setup and hold times are measured.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<SetHoldClkSlp> POSitive | NEGative | EITHer
**EITHer**
The clock edges next to the data edge are considered regardless of the clock slope.
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Setup/Hold measurement settings" on page 349

**MEASurement<m>:AMPTime:CLCK<n>:LSELect** <ClkLevSel>

Selects the reference level of the clock on which the time is measured. Reference level and clock slope define the time point for setup and hold measurements.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
The suffix is irrelevant, omit it.

**Parameters:**
<ClkLevSel> UPPer | MIDDle | LOWer
*RST: MIDDle

**Usage:** Asynchronous command

**Manual operation:** See "Clock ref level" on page 350

---

**MEASurement<m>:AMPTime:DATA<n>:LSELect** <DatLevSel>

Selects the reference level of the data on which the setup and hold time are measured.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
The suffix is irrelevant, omit it.

**Parameters:**
<DatLevSel> UPPer | MIDDle | LOWer
*RST: MIDDle

**Usage:** Asynchronous command

**Manual operation:** See "Data ref level / Reference level" on page 351

---

**MEASurement<m>:AMPTime:DTOTrigger<n>:SLOPe** <DlyTrigSlp>

Sets the edge direction to be used for delay measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
The suffix is irrelevant.

**Parameters:**
<DlyTrigSlp> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Delay to trigger measurement settings" on page 351

---

**MEASurement<m>:AMPTime:DTOTrigger<n>:LSELect** <DlyTrigLevSel>

Selects the reference level of the measurement source on which the delay is measured.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
The suffix is irrelevant.

**Parameters:**
<DlyTrigLevSel> UPPer | MIDDle | LOWer
*RST: MIDDle

**Usage:** Asynchronous command

**Manual operation:** See "Delay to trigger measurement settings" on page 351

---

**MEASurement<m>:AMPTime:LCHeck<n>:VALid** <ValidRange>

Enables or disables limit checking for amplitude vs. time measurements in the specified measurement channel.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..50
Number of the amplitude/time measurement, see Table 24-6.

**Parameters:**
<ValidRange> ILIMit | ULIMit | LLIMit | OLIMit

**ILIMit**
Inside (within) limit; between the upper and lower limit values

**ULIMit**
Upper limit; above the upper limit value

**LLIMit**
Lower limit; below the lower limit value

**OLIMit**
Outside limit; above the upper limit or below the lower limit values

*RST: ILIMit

**Usage:** Asynchronous command

**Manual operation:** See "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394

---

**MEASurement<m>:AMPTime:LCHeck<n>:LOWer:LIMit** <LowerLimit>
**MEASurement<m>:AMPTime:LCHeck<n>:UPPer:LIMit** <UpperLimit>

Define the lower and upper limit for limit checking, respectiveley. The valid range is defined using the `MEASurement<m>:AMPTime:LCHeck<n>:VALid` command.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..50
Number of the amplitude/time measurement, see Table 24-6.

**Parameters:**
<UpperLimit> Range: -100 to 100
Increment: 10E-6
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394

---

**MEASurement<m>:AMPTime:LCHeck<n>:LOWer:MARGin** <LowerMargin>
**MEASurement<m>:AMPTime:LCHeck<n>:UPPer:MARGin** <UpperMargin>

Define the lower and upper margins for the limit check, respectively. Margins are not as strict as limits and must be within the valid value range. The valid range is defined using the `MEASurement<m>:AMPTime:LCHeck<n>:VALid` command.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..50
Number of the amplitude/time measurement type, see Table 24-6.

**Parameters:**
<UpperMargin> Range: -100 to 100
Increment: 10E-6
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394

### 24.12.4 Arithmetic for amplitude/time and spectrum measurements

MEASurement<m>:AMPTime:ARIThmetic:CONStant ..... 1448
MEASurement<m>:AMPTime:ARIThmetic:COUNit ..... 1448
MEASurement<m>:AMPTime:ARIThmetic:CUNState ..... 1449
MEASurement<m>:AMPTime:ARIThmetic:CUUNit ..... 1449
MEASurement<m>:AMPTime:ARIThmetic:CUUState ..... 1450
MEASurement<m>:AMPTime:ARIThmetic:OPERator ..... 1450
MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:AMPTime ..... 1450

MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:GROup....1451
MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:SPECtrum....1451
MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:TYPE....1452
MEASurement<m>:SPECtrum:ARIThmetic:CONStant....1452
MEASurement<m>:SPECtrum:ARIThmetic:COUNit....1453
MEASurement<m>:SPECtrum:ARIThmetic:CUNState....1453
MEASurement<m>:SPECtrum:ARIThmetic:CUUNit....1454
MEASurement<m>:SPECtrum:ARIThmetic:CUUState....1454
MEASurement<m>:SPECtrum:ARIThmetic:OPERator....1454
MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:AMPTime....1454
MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:GROup....1455
MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:SPECtrum....1455
MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:TYPE....1456

**MEASurement<m>:AMPTime:ARIThmetic:CONStant** <ArithConstant>

Sets a constant numeric value as the second operand of the amplitude/time arithmetic measurement.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**

<ArithConstant>
Range: 0 to 0
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Constant" on page 354

**MEASurement<m>:AMPTime:ARIThmetic:COUNit** <ArithCstUnt>

Selects the operation type to be performed on the selected operands for the amplitude/time arithmetic measurement.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**

<ArithCstUnt> DBM | DB | DB_HZ | DB_GHZ | DBC | DBC_HZ | DBPW_HZ | DBW | DBPW | DBM_HZ | DBMW | W | MW | W_HZ | J | V_S | A_S | DBV | DBMV | DBMV_HZ | DBMV_MHZ | DBUV | DBUV_HZ | DBUV_SQRT_HZ | DBUV_MHZ | DBUV_M | DBMV_M_HZ | DBUV_M_MHZ | V | MV | UV | VPP | VV | UV_HZ | UV_M_HZ | V_SQRT_HZ | DBA | DBMA | DBMA_HZ | DBMA_MHZ | DBUA | DBUA_HZ | DBUA_SQRT_HZ | DBUA_MHZ | DBUA_M | DBUA_M_HZ | DBUA_M_MHZ | A | UA_HZ | UA_M_HZ | A_SQRT_HZ | PCT | DBPT | DBPT_HZ | H | PH_GHZ | F | FF_GHZ | OHM | SIEMENS | S | DBPS | DBNS | DBUS | DBMS | DBS | HZ | DBHZ | DBKHZ | DBMHZ | DBGHZ | M | IN | DEG | RAD | PCT | SYMB | VA_LIN | VA_LOG | NONE | IRE | MILES | SYMB_S | MSYMB_S | MBIT_S | BIT_S | BIT | WORD | PPM | DB | PER_SEC | S_DIV | HZ_DIV | V_DIV | A_DIV | OHM_DIV | NONE_DIV | DIV | BER | S | S_S | BAUD | TDEG | VS | USER | UINV | W_DIV | V_V | A_V | W_V | BYTS | V_A | V_W | VA | VAR | UI | LN

*RST: NONE

**Usage:** Asynchronous command

**Manual operation:** See "Constant unit" on page 354

---

**MEASurement<m>:AMPTime:ARIThmetic:CUNState** <ArithCustomNmeSt>

Enables the use of user-defined custom name for the amplitude/time arithmetic measurement.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**

<ArithCustomNmeSt> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Custom" on page 354

---

**MEASurement<m>:AMPTime:ARIThmetic:CUUNit** <ArithCustomUnit>

Sets a user-defined value of the unit for the amplitude/time arithmetic measurement.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ArithCustomUnit>

**Usage:** Asynchronous command

---

**MEASurement<m>:AMPTime:ARIThmetic:CUUState** <ArithCustomUntSt>

Enables the use of user-defined custom unit for the amplitude/time arithmetic measurement. You can set the unit with `MEASurement<m>:AMPTime:ARIThmetic:CUUNit`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ArithCustomUntSt> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

---

**MEASurement<m>:AMPTime:ARIThmetic:OPERator** <ArithOperator>

Selects the operation type to be performed on the selected operands for the amplitude/time arithmetic measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ArithOperator> ADD | SUB | MUL | DIV
*RST: ADD

**Usage:** Asynchronous command

**Manual operation:** See "Operator" on page 354

---

**MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:AMPTime**
<ArithOp1MeasAmplitudeTime>

Selects the amplitude/time arithmetics measurements for the measurment group.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
Selects the operand.

**Parameters:**

<Op1MeasAmpTime> HIGH | LOW | AMPLitude | MAXimum | MINimum | PDELta | MEAN | RMS | STDDev | POVershoot | NOVershoot | AREA | RTIMe | FTIMe | PPULse | NPULse | PERiod | FREQuency | PDCYcle | NDCYcle | CYCarea | CYCMean | CYCRms | CYCStddev | PULCnt | DELay | PHASe | BWIDth | PSWitching | NSWitching | PULSetrain | EDGecount | SETup | HOLD | SHT | SHR | CCJitter | NCJitter | CCWidth | CCDutycycle | TIE | UINTerval | DRATe | SKWDelay | SKWPhase | DTOTrigger | PROBemeter | SLERising | SLEFalling | ATAR

See Table 24-6.

*RST: HIGH

**Usage:** Asynchronous command

**Manual operation:** See "Operand 1" on page 354

---

**MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:GROup**
<ArithOp2MeasGroup>

Sets the measurement group for the second operand of the amplitude/time arithmetic measurement.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 2

**Parameters:**

<Op2MeasGroup> MEAS1 | MEAS2 | MEAS3 | MEAS4 | MEAS5 | MEAS6 | MEAS7 | MEAS8 | IMEAS | IMEas | QUICK | QUICk

*RST: MEAS1

**Usage:** Asynchronous command

**Manual operation:** See "Operand 2 meas group / Operand 2" on page 354

---

**MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:SPECtrum**
<ArithOp1MeasSpectrum>

Selects the spectrum arithmetics measurements for the measurment group.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
Selects the operand.

**Parameters:**
<Op1MeasSpectrum> CPOWer | ACPower | OBWidth | SBWidth | TOI | AMMod | THD | THDPCT | THDPct | THDA | THDU | THDR | PLISt | HAR | THDF | SPAR

*RST: CPOWer

**Usage:** Asynchronous command

**Manual operation:** See "Operand 1" on page 354

---

**MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:TYPE** <ArithOp2Type>

Sets the type of amplitude arithmetic measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 2

**Parameters:**
<ArithOp2Type> MEASurement | CONStant

**MEASurement**
Performs an arithmetic operation between two measurement results.

**CONStant**
Performs an operation between a measurement result and a user defined fixed constant.

*RST: MEASurement

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 353

---

**MEASurement<m>:SPECtrum:ARIThmetic:CONStant** <ArithConstant>

Sets a constant numeric value as the second operand of the spectrum arithmetic measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ArithConstant> Range: - 3.4 e38 to + 3.4 e38
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Constant" on page 354

**MEASurement<m>:SPECtrum:ARIThmetic:COUNit** <ArithCstUnt>

Sets the unit for the constant, set with `MEASurement<m>:SPECtrum:ARIThmetic:CONStant`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ArithCstUnt> DBM | DB | DB_HZ | DB_GHZ | DBC | DBC_HZ | DBPW_HZ | DBW | DBPW | DBM_HZ | DBMW | W | MW | W_HZ | J | V_S | A_S | DBV | DBMV | DBMV_HZ | DBMV_MHZ | DBUV | DBUV_HZ | DBUV_SQRT_HZ | DBUV_MHZ | DBUV_M | DBMV_M_HZ | DBUV_M_MHZ | V | MV | UV | VPP | VV | UV_HZ | UV_M_HZ | V_SQRT_HZ | DBA | DBMA | DBMA_HZ | DBMA_MHZ | DBUA | DBUA_HZ | DBUA_SQRT_HZ | DBUA_MHZ | DBUA_M | DBUA_M_HZ | DBUA_M_MHZ | A | UA_HZ | UA_M_HZ | A_SQRT_HZ | PCT | DBPT | DBPT_HZ | H | PH_GHZ | F | FF_GHZ | OHM | SIEMENS | S | DBPS | DBNS | DBUS | DBMS | DBS | HZ | DBHZ | DBKHZ | DBMHZ | DBGHZ | M | IN | DEG | RAD | PCT | SYMB | VA_LIN | VA_LOG | NONE | IRE | MILES | SYMB_S | MSYMB_S | MBIT_S | BIT_S | BIT | WORD | PPM | DB | PER_SEC | S_DIV | HZ_DIV | V_DIV | A_DIV | OHM_DIV | NONE_DIV | DIV | BER | S | S_S | BAUD | TDEG | VS | USER | UINV | W_DIV | V_V | A_V | W_V | BYTS | V_A | V_W | VA | VAR | UI | LN
*RST: NONE

**Usage:** Asynchronous command

**Manual operation:** See "Constant unit" on page 354

**MEASurement<m>:SPECtrum:ARIThmetic:CUNState** <ArithCustomNmeSt>

Enables the use of user-defined custom name for the spectrum arithmetic measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ArithCustomNmeSt> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Custom" on page 354

**MEASurement<m>:SPECtrum:ARIThmetic:CUUNit** <ArithCustomUnit>

Sets a user-defined value of the unit for the spectrum arithmetic measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ArithCustomUnit>

**Usage:** Asynchronous command

**MEASurement<m>:SPECtrum:ARIThmetic:CUUState** <ArithCustomUntSt>

Enables the use of user-defined custom unit for the amplitude/time arithmetic measurement. You can set the unit with `MEASurement<m>:AMPTime:ARIThmetic:CUUNit`.

**Suffix:**
<m> 1..10

**Parameters:**
<ArithCustomUntSt> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**MEASurement<m>:SPECtrum:ARIThmetic:OPERator** <ArithOperator>

Selects the operation type to be performed on the selected operands for the spectrum arithmetic measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ArithOperator> ADD | SUB | MUL | DIV
*RST: ADD

**Usage:** Asynchronous command

**MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:AMPTime** <ArithOp1MeasAmplitudeTime>

Selects the amplitude/time arithmetics measurements for the measurment group.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
Selects the operand.

**Parameters:**
<Op1MeasAmpTime> HIGH | LOW | AMPLitude | MAXimum | MINimum | PDELta | MEAN | RMS | STDDev | POVershoot | NOVershoot | AREA | RTIMe | FTIMe | PPULse | NPULse | PERiod | FREQuency | PDCYcle | NDCYcle | CYCarea | CYCMean | CYCRms | CYCStddev | PULCnt | DELay | PHASe | BWIDth | PSWitching | NSWitching | PULSetrain | EDGecount | SETup | HOLD | SHT | SHR | CCJitter | NCJitter | CCWidth | CCDutycycle | TIE | UINTerval | DRATe | SKWDelay | SKWPhase | DTOTrigger | PROBemeter | SLERising | SLEFalling | ATAR
*RST: HIGH

**Usage:** Asynchronous command

**Manual operation:** See "Operand 1" on page 354

---

**MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:GROup**
<ArithOp2MeasGroup>

Sets the measurement group for the second operand of the spectrum arithmetic measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 2

**Parameters:**
<ArithOp2MeasGroup> MEAS1 | MEAS2 | MEAS3 | MEAS4 | MEAS5 | MEAS6 | MEAS7 | MEAS8 | IMEAS | IMEas | QUICK | QUICk
*RST: MEAS1

**Usage:** Asynchronous command

**Manual operation:** See "Operand 2 meas group / Operand 2" on page 354

---

**MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:SPECtrum**
<ArithOp1MeasSpectrum>

Selects the first operand for the sepctrum arithmetics measurements.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
Selects the operand.

**Parameters:**

<Op1MeasSpectrum> CPOWer | ACPower | OBWidth | SBWidth | TOI | AMMod | THD | THDPCT | THDPct | THDA | THDU | THDR | PLISt | HAR | THDF | SPAR

See Table 24-8.

*RST: CPOWer

**Usage:** Asynchronous command

**Manual operation:** See "Operand 1" on page 354

---

**MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:TYPE** <ArithOp2Type>

Sets the type of spectrum arithmetic measurement.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 2

**Parameters:**

<ArithOp2Type> MEASurement | CONStant

**MEASurement**
Performs an arithmetic operation between two measurement results.

**CONStant**
Performs an operation between a measurement result and a user defined fixed constant.

*RST: MEASurement

**Usage:** Asynchronous command

## 24.12.5 Eye diagram measurements

The following table lists the <MeasType> parameter values with a short description.

In addition, the suffixes <n> for limit-check remote commands are listed.

For a detailed description, see Chapter 8.2.6.1, "Overview of eye diagram measurements", on page 357.

***Table 24-7: Eye diagram measurements***

| <MeasType> parameter value (remote control) | Measurement (manual operation) | Description/Result | Suffix <n> |
|---|---|---|---|
| | | not used | 1 |
| ERPercent | Extinction ratio (%) | Eye base / Eye top *100<br>Prerequisite: Eye base > 0 and Eye top > 0 | 2 |

| <MeasType> parameter value (remote control) | Measurement (manual operation) | Description/Result | Suffix <n> |
|---|---|---|---|
| ERDB | Extinction ratio (dB) | 10*log (Eye top / Eye base) | 3 |
| EHEight | Eye height | Vertical eye opening | 4 |
| EWIDth | Eye width | Horizontal eye opening | 5 |
| ETOP | Eye top | Mean of the upper vertical histogram | 6 |
| EBASe | Eye base | Mean of the lower vertical histogram | 7 |
| | | not used | 8 |
| CPPercent | | Crossing points percentage | 9 |
| QFACtor | Q factor | (Eye top – Eye base) / (σtop + σbase) | 10 |
| | | not used | 11...13 |
| RMSNoise | Noise (RMS) | Quadratic mean of the noise of eye top and eye base | 14 |
| SNRatio | S/N ratio | Signal-to-noise ratio<br>10 * log (Eye amplitude / Noise RMS) | 15 |
| DCDistortion | Duty cycle distortion | 20 * log (Eye amplitude / Noise RMS) | 16 |
| ERTime | Eye rise time | Duration for signal to rise from 10% to 90% of the high signal level | 17 |
| EFTime | Eye fall time | Duration for signal to fall from 90% to 10% of the high signal level | 18 |
| EBRate | Eye bit rate | Frequency between two crossings | 19 |
| EAMPlitude | Eye amplitude | Eye top - Eye base | 20 |
| | | not used | 21...27 |
| PPJitter | Jitter (peak to peak) | Average of the jitter for both crossing points<br>(σcrossing1 + σcrossing2) / 2 | 28 |
| STDJitter | Jitter (6*σ) | Jitter *6 | 29 |
| RMSJitter | Jitter (RMS) | Quadratic mean of the jitter at both crossing points | 30 |

MEASurement<m>:EYEJitter:AUToscale ..... 1458
MEASurement<m>:EYEJitter:LCHeck<n>:VALid ..... 1458
MEASurement<m>:EYEJitter:LCHeck<n>:LOWer:LIMit ..... 1459
MEASurement<m>:EYEJitter:LCHeck<n>:UPPer:LIMit ..... 1459
MEASurement<m>:EYEJitter:LCHeck<n>:LOWer:MARGin ..... 1459
MEASurement<m>:EYEJitter:LCHeck<n>:UPPer:MARGin ..... 1459
MEASurement<m>:EYEJitter:GENSettings:MLEHistogram:ALGorithm ..... 1460
MEASurement<m>:EYEJitter:GENSettings:MLEHistogram:THReshold ..... 1460
MEASurement<m>:EYEJitter:GENSettings:RELative:STARt ..... 1460
MEASurement<m>:EYEJitter:GENSettings:RELative:STOP ..... 1461
MEASurement<m>:EYEJitter:DCDistortion:RELative:LEVel ..... 1461
MEASurement<m>:EYEJitter:EYEWidth:ALGorithm ..... 1461

MEASurement<m>:EYEJitter:EYEWidth:MEASwindow....1462
MEASurement<m>:EYEJitter:EYEWidth:RELative:STARt....1462
MEASurement<m>:EYEJitter:EYEWidth:RELative:STOP....1463
MEASurement<m>:EYEJitter:EYEHeight:ALGorithm....1463
MEASurement<m>:EYEJitter:EYEHeight:RELative:STARt....1463
MEASurement<m>:EYEJitter:EYEHeight:RELative:STOP....1464
MEASurement<m>:EYEJitter:JITTer:MEASlevel....1464
MEASurement<m>:EYEJitter:JITTer:RELative:LEVel....1465
MEASurement<m>:EYEJitter:RTFTime:RELative:STOP....1465
MEASurement<m>:EYEJitter:RTFTime:RELative:STARt....1465

**MEASurement<m>:EYEJitter:AUToscale**

Defines optimized settings to perform an eye diagram measurement for the selected source.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Autoset" on page 359

**MEASurement<m>:EYEJitter:LCHeck<n>:VALid** <ValidRange>

Enables or disables limit checking for eye/jitter measurements in the specified measurement channel.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..30
Number of the eye/jitter measurement, see Table 24-7.

**Parameters:**

<ValidRange> ILIMit | ULIMit | LLIMit | OLIMit

**ILIMit**
Inside (within) limit; between the upper and lower limit values

**ULIMit**
Upper limit; above the upper limit value

**LLIMit**
Lower limit; below the lower limit value

**OLIMit**
Outside limit; above the upper limit or below the lower limit values

*RST: ILIMit

**Usage:** Asynchronous command

**Manual operation:** See "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394

---

**MEASurement<m>:EYEJitter:LCHeck<n>:LOWer:LIMit** <LowerLimit>
**MEASurement<m>:EYEJitter:LCHeck<n>:UPPer:LIMit** <UpperLimit>

Define the lower and upper limit for the limit check, respectively. The valid range is defined using the `MEASurement<m>:EYEJitter:LCHeck<n>:VALid` command.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..30
Number of the eye/jitter measurement, see Table 24-7.

**Parameters:**
<UpperLimit>
Range: -100 to 100
Increment: 10E-6
*RST: 0
Default unit: The unit depends on the selected measurement.

**Usage:** Asynchronous command

**Manual operation:** See "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394

---

**MEASurement<m>:EYEJitter:LCHeck<n>:LOWer:MARGin** <LowerMargin>
**MEASurement<m>:EYEJitter:LCHeck<n>:UPPer:MARGin** <UpperMargin>

Defines the upper margin for the limit check. Margins are not as strict as limits and must be within the valid value range. The valid range is defined using the `MEASurement<m>:EYEJitter:LCHeck<n>:VALid` command.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..30
Number of the eye/jitter measurement, see Table 24-7.

**Parameters:**
<UpperMargin>
Range: -100 to 100
Increment: 10E-6
*RST: 0
Default unit: The unit depends on the selected measurement.

**Usage:** Asynchronous command

**Manual operation:** See "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394

**MEASurement<m>:EYEJitter:GENSettings:MLEHistogram:ALGorithm**
<Algorithm>

Defines how to compute the top/base level and the corresponding sigma values.

**Suffix:**
<m> 1..10

**Parameters:**
<Algorithm> MEAN | PEAK | INNPeak | OUTPeak
*RST: MEAN

**Usage:** Asynchronous command

**Manual operation:** See "General" on page 359

**MEASurement<m>:EYEJitter:GENSettings:MLEHistogram:THReshold**
<Threshold>

Sets the threshold used to separate peaks: two peaks are separated if they are split by a valley of height *Threshold * Maximal_histogram_value*.

**Suffix:**
<m> 1..10

**Parameters:**
<Threshold> Range: 1 to 99
Increment: 0.1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "General" on page 359

**MEASurement<m>:EYEJitter:GENSettings:RELative:STARt** <Start>

Defines the start of the measurement window in relative values.

This settings is relevant, if `MEASurement<m>:MAIN` is set to one of the following: `ERPercent, ERDB, EHEight, QFACtor, RMSNoise, SNRatio, EAMPlitude`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Start> Range: 1 to 99
Increment: 0.1
*RST: 40
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "General" on page 359

---

**MEASurement<m>:EYEJitter:GENSettings:RELative:STOP** <Stop>

Defines the stop of the measurement window in relative values.

This settings is relevant, if `MEASurement<m>:MAIN` is set to one of the following: `ERPercent, ERDB, EHEight, QFACtor, RMSNoise, SNRatio, EAMPlitude` .

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Stop>
Range: 1 to 99
Increment: 0.1
*RST: 60
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "General" on page 359

---

**MEASurement<m>:EYEJitter:DCDistortion:RELative:LEVel** <RelLevel>

Defines the level of the measurement window in relative values.

This settings is relevant, if `MEASurement<m>:MAIN` is set to `DCDistortion`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<RelLevel>
Range: 1 to 99
Increment: 0.1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Duty cycle disortion" on page 362

---

**MEASurement<m>:EYEJitter:EYEWidth:ALGorithm** <Algorithm>

Selects the algorithm for the measurement of the width.

This settings is relevant, if `MEASurement<m>:MAIN` is set to `EWIDth`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Algorithm> ACTWidth | TSIGma

**ACTWidth**
Measure actual width. You can define a measuerement window for the measurement.

**TSIGma**
Extrapolate using three sigma. The eye width is measured at the crossing level.

*RST: TSIGma

**Usage:** Asynchronous command

**Manual operation:** See "Eye width" on page 361

---

**MEASurement<m>:EYEJitter:EYEWidth:MEASwindow** <UseWindow>

If enabled, you can define a user defined start and stop values for the eye width measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<UseWindow> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Eye width" on page 361

---

**MEASurement<m>:EYEJitter:EYEWidth:RELative:STARt** <Start>

Sets the start of the measurement window in relative values.

This settings is relevant, if `MEASurement<m>:MAIN` is set to `EWIDth`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Start> Range: 1 to 99
Increment: 0.1
*RST: 40
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Eye width" on page 361

---

**MEASurement<m>:EYEJitter:EYEWidth:RELative:STOP** <Stop>

Sets the stop of the measurement window in relative values.

This settings is relevant, if `MEASurement<m>:MAIN` is set to `EWIDth`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Stop>
Range: 1 to 99
Increment: 0.1
*RST: 60
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Eye width" on page 361

---

**MEASurement<m>:EYEJitter:EYEHeight:ALGorithm** <Algorithm>

Selects the algorithm for the measurement of the height.

This settings is relevant, if `MEASurement<m>:MAIN` is set to `EHEight`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Algorithm> ACTHeight | TSIGma

**ACTHeight**
Measure actual height. You can define a measuerement window for the measurement.

**TSIGma**
Extrapolate using three sigma. The eye height is measured with the settings defined with the general commands.

*RST: TSIGma

**Usage:** Asynchronous command

**Manual operation:** See "Eye height" on page 361

---

**MEASurement<m>:EYEJitter:EYEHeight:RELative:STARt** <Start>

Defines the start of the measurement window in relative values.

This settings is relevant, if MEASurement<m>:MAIN is set to EHEight.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Start> Range: 1 to 99
Increment: 0.1
*RST: 40
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Eye height" on page 361

---

**MEASurement<m>:EYEJitter:EYEHeight:RELative:STOP** <Stop>

Defines the start of the measurement window in relative values.

This settings is relevant, if MEASurement<m>:MAIN is set to EHEight.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Stop> Range: 1 to 99
Increment: 0.1
*RST: 60
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Eye height" on page 361

---

**MEASurement<m>:EYEJitter:JITTer:MEASlevel** <UseLevel>

If enabled, you can define a user defined level for the eye measurement jitter.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<UseLevel> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Jitter" on page 363

**MEASurement<m>:EYEJitter:JITTer:RELative:LEVel** <RelLevel>

Defines the level of the measurement window in relative values.

This settings is relevant, if `MEASurement<m>:MAIN` is set to one of the following measuerements: `PPJitter, STDJitter, RMSJitter`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<RelLevel>
Range: 1 to 99
Increment: 0.1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Jitter" on page 363

**MEASurement<m>:EYEJitter:RTFTime:RELative:STOP** <Stop>

Sets the local stop level of range for eye measurements rise/fall time.

**Suffix:**
<m> 1..10

**Parameters:**
<Stop>
Range: 1 to 99
Increment: 0.1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Eye rise time,Eye fall time" on page 362

**MEASurement<m>:EYEJitter:RTFTime:RELative:STARt** <Start>

Sets the local start level of range for eye measurements rise/fall time.

**Suffix:**
<m> 1..10

**Parameters:**
<Start>
Range: 1 to 99
Increment: 0.1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Eye rise time,Eye fall time" on page 362

### 24.12.6 Spectrum

The following table lists the <MeasType> parameter values with a short description.

In addition, the suffixes <n> for limit-check remote commands are listed.

For a detailed description, see Chapter 8.2.7.1, "Overview of spectrum measurements", on page 363.

***Table 24-8: Spectrum measurements***

| <MeasType> parameter value (remote control) | Measurement (manual operation) | Description/Result | Suffix <n> |
|---|---|---|---|
| CPOWer | Channel power | Power integrated over the sample values defined by a center frequency and a bandwidth; based on a defined impedance; the result is given in mW | 1 |
| | | not used | 2 |
| OBWidth | Occupied bandwidth | From the defined center frequency, symmetric sample value pairs to the left and right are integrated until a user-defined percentage of the total power is reached | 3 |
| SBWidth | Bandwidth | n dB down bandwidth. The samples to the left and right of the peak value are analyzed until the n dB threshold is exceeded. Frequencies at which the threshold is exceeded define the limits of the requested bandwidth. | 4 |
| | | not used | 5, 6 |
| THD | THD[dB] | Total harmonic distortion in dB | 7 |
| THDPCT | THD[%] | Total harmonic distortion in % | 8 |
| THDA | THD_a | Is equivalent to THD. Requires option R&S RTP-K37. | 9 |
| THDU | THD_u | Requires option R&S RTP-K37. | 10 |
| THDR | THD_r | Distortion factor. Requires option R&S RTP-K37. | 11 |
| PLISt | Peak list | List of frequency and peak power value pairs. | 12 |
| HAR | Harmonic search | Returns the measured harmonics. For each harmonic, the frequency and the value are listed. To get the number of result pairs (= harmonics), use `MEASurement<m>:RESult:COUNt?`. | 13 |
| THDF | THD_f | Root mean square of the sum of all amplitudes of the harmonic waves in relation to the amplitude at the fundamental frequency (first harmonic). Requires option R&S RTP-K37. | 14 |
| SPAR | Arithmetic for spectrum measurements | Basic arithmetic calculations with measurement values and scalars. | 15 |

MEASurement<m>:SPECtrum:CPOWer:BANDwidth ..... 1467
MEASurement<m>:SPECtrum:OBANdwidth ..... 1467
MEASurement<m>:SPECtrum:CPOWer:CFRequency ..... 1468

MEASurement<m>:SPECtrum:NDBDown ..... 1468
MEASurement<m>:SPECtrum:PEXCursion ..... 1468
MEASurement<m>:SPECtrum:ATHReshold ..... 1469
MEASurement<m>:SPECtrum:RESult<n>:COUNt ..... 1469
MEASurement<m>:SPECtrum:RESult<n>:MODE ..... 1469
MEASurement<m>:RESult:MAXCount ..... 1470
MEASurement<m>:RESult:SHFRequency ..... 1470
MEASurement<m>:RESult:SHLabels ..... 1471
MEASurement<m>:SPECtrum:LCHeck<n>:VALid ..... 1471
MEASurement<m>:SPECtrum:LCHeck<n>:LOWer:LIMit ..... 1472
MEASurement<m>:SPECtrum:LCHeck<n>:UPPer:LIMit ..... 1472
MEASurement<m>:SPECtrum:LCHeck<n>:LOWer:MARGin ..... 1473
MEASurement<m>:SPECtrum:LCHeck<n>:UPPer:MARGin ..... 1473

**MEASurement<m>:SPECtrum:CPOWer:BANDwidth** <ChPowBw>

Defines the bandwidth over which the channel power is calculated.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ChPowBw>
Range: 0 to 4E+9
Increment: 1
*RST: 0
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Channel BW" on page 366

**MEASurement<m>:SPECtrum:OBANdwidth** <OccupiedBW>

Defines the percentage of the total power used to determine the occupied bandwidth.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<OccupiedBW>
Range: 0.1 to 99.9
Increment: 1
*RST: 20
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Occup. BW" on page 366

**MEASurement<m>:SPECtrum:CPOWer:CFRequency** <ChPowCtrFreq>

Defines the center frequency from which the channel power is calculated over the specified bandwidth.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ChPowCtrFreq>
Range: 0 to 6E+9
Increment: 1
*RST: 0
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Channel CF" on page 366

**MEASurement<m>:SPECtrum:NDBDown** <NDbDown>

Defines the threshold until which the samples to the left and right of the peak value are analyzed in order to determine the "N dB down bandwidth".

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<NDbDown>
Range: 0 to 100
Increment: 1
*RST: 20
Default unit: dB

**Usage:** Asynchronous command

**Manual operation:** See "N db down" on page 366

**MEASurement<m>:SPECtrum:PEXCursion** <Value>

Defines a relative threshold, the minimum level value by which the waveform must rise or fall to be considered as a peak. To avoid identifying noise peaks, enter a peak excursion value that is higher than the noise levels.

**Suffix:**
<m> 1..10
The suffix is irrelevant.

**Parameters:**
<Value>

**Usage:** Asynchronous command

**Firmware/software:** Version 2.70

**Manual operation:** See "Peak excursion" on page 319

---

**MEASurement<m>:SPECtrum:ATHReshold** <Value>

Defines an absolute threshold as an additional condition for the peak search. Only peaks that exceed the threshold are detected.

This setting is only available for spectrum waveforms. It is valid for cursor measurements, spectrum measurements and peak search.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Value> numeric value
Default unit: dBm

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 318

---

**MEASurement<m>:SPECtrum:RESult<n>:COUNt** <MaxNoOfResults>

Sets the maximum number of measurement results that are listed in the result table. Available for peak list and harmonic search measurements (`MEASurement:MAIN` is set to `PLISt` or `HAR`).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 12 | 13
12: Peak list measurements
13: Harmonic search measurements

**Parameters:**
<MaxNoOfResults> Range: 1 to 1000
Increment: 1
*RST: 10

**Usage:** Asynchronous command

**Manual operation:** See "Max results" on page 367

---

**MEASurement<m>:SPECtrum:RESult<n>:MODE** <ResultMode>

Sets the way how the measurement results are displayed. Available only if `MEASurement:MAIN` is set to `PLISt` or `HAR`.

**Suffix:**
<m> 1..10
<n> 12 | 13
12: Peak list measurements
13: Harmonic search measurements

**Parameters:**
<ResultMode> ABS | REL

**ABS**
The harmonics/peaks are shown in absolute value, dBm.

**REL**
The level of the carrier is shown in absolute value dBm. The values the harmonics/peaks are shown relatively to the carrier in dBc.

*RST: ABS

**Example:** `:MEASurement1:SPECtrum:RESult13:MODE ABS`
The harmonic measurements are shown in absolute value, dBm.

**Usage:** Asynchronous command

**Manual operation:** See "Result mode" on page 368

---

**MEASurement<m>:RESult:MAXCount** <Number>
**MEASurement<m>:RESult:MAXCount?** <MeasType>

Defines the maximum number of peaks that are labeled in the diagram.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Number> Maximum number of peaks

**Query parameters:**
<MeasType> PLISt

Only for peaklist measurements. For other measurement, this command returns an error.

**Manual operation:** See "Result labels" on page 368

---

**MEASurement<m>:RESult:SHFRequency** <MeasType>, <State>
**MEASurement<m>:RESult:SHFRequency?** <MeasType>

Includes the frequency of the detected peak in the diagram labels (if `MEASurement<m>:RESult:LABorder=ON`).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<State> ON | OFF

**Parameters for setting and query:**
<MeasType> PLISt
Only for peaklist measurements. For other measurements, this command returns an error.

**Example:** Display labels.
```
MEAS:RES:SHL PLIS,ON
```
Show frequency in labels.
```
MEAS:RES:SHFR PLIS,ON
```
Query the type of labels for peak lists.
```
MEAS:RES:SHFR? PLIS
//Result: ON
```

**Manual operation:** See "Result labels" on page 368

---

**MEASurement<m>:RESult:SHLabels** <MeasType>, <State>
**MEASurement<m>:RESult:SHLabels?** <MeasType>

Defines whether a description (label) is displayed for each detected peak in the spectrum diagram.

The layout of the label is defined by `MEASurement<m>:RESult:LABorder`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<State> ON | OFF

**Parameters for setting and query:**
<MeasType> PLISt
Only for peaklist measurements. For other measurements, this command returns an error.

**Example:** Display labels.
```
MEAS:RES:SHL PLIS,ON
```

**Manual operation:** See "Result labels" on page 368

---

**MEASurement<m>:SPECtrum:LCHeck<n>:VALid** <ValidRange>

Enables or disables limit checking for spectrum measurements in the specified measurement channel.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..15
Number of the spectrum measurement, see Table 24-8.

**Parameters:**
<ValidRange> ILIMit | ULIMit | LLIMit | OLIMit

**ILIMit**
Inside (within) limit; between the upper and lower limit values

**ULIMit**
Upper limit; above the upper limit value

**LLIMit**
Lower limit; below the lower limit value

**OLIMit**
Outside limit; above the upper limit or below the lower limit values

*RST: ILIMit

**Usage:** Asynchronous command

**Manual operation:** See "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394

---

**MEASurement<m>:SPECtrum:LCHeck<n>:LOWer:LIMit** <LowerLimit>
**MEASurement<m>:SPECtrum:LCHeck<n>:UPPer:LIMit** <UpperLimit>

Define the lower and upper limits for the limit check, respectively. The valid range is defined using the `MEASurement<m>:SPECtrum:LCHeck<n>:VALid` command.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..15
Number of the spectrum measurement, see Table 24-8.

**Parameters:**
<UpperLimit> Range: -100 to 100
Increment: 10E-6
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394

**MEASurement<m>:SPECtrum:LCHeck<n>:LOWer:MARGin** <LowerMargin>
**MEASurement<m>:SPECtrum:LCHeck<n>:UPPer:MARGin** <UpperMargin>

Define the lower and upper margins for the limit check, respectively. Margins are not as strict as limits and must be within the valid value range. The valid range is defined using the `MEASurement<m>:SPECtrum:LCHeck<n>:VALid` command.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..15
Number of the spectrummeasurement, see Table 24-8.

**Parameters:**
<UpperMargin> Range: -100 to 100
Increment: 10E-6
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394

## 24.12.7 Optical measurements

The following table lists the <MeasType> parameter values with a short description.

In addition, the suffixes <n> for limit-check remote commands are listed.

*Table 24-9: Optical measurements*

| <MeasType> parameter value (remote control) | Measurement (manual operation) | Description/Result | Suffix <n> |
|---|---|---|---|
| OAP | Average power | True average component of an optical signal. $X_{Average} = \frac{1}{N_{Eval}} * \sum_{i=1}^{N_{Eval}} x(i)$ | 1 |
| OMA | Modulation amplitude | Measure of the difference between the optical power of a non-return-to-zero (NRZ) one pulse and the optical power of an NRZ zero pulse.<br>OMA requires an NRZ pattern. | 2 |

MEASurement<m>:OPTical:OMAMplitude:RELative:STARt....1474
MEASurement<m>:OPTical:OMAMplitude:RELative:STOP....1474
MEASurement<m>:OPTical:UNIT....1474
MEASurement<m>:OPTical:LCHeck<n>:LOWer:LIMit....1475
MEASurement<m>:OPTical:LCHeck<n>:UPPer:LIMit....1475
MEASurement<m>:OPTical:LCHeck<n>:LOWer:MARGin....1475
MEASurement<m>:OPTical:LCHeck<n>:UPPer:MARGin....1475
MEASurement<m>:OPTical:LCHeck<n>:VALid....1475

**MEASurement<m>:OPTical:OMAMplitude:RELative:STARt** <Start>

Sets the start value of the range for the optical modulation amplitude.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Start>
Range: 1 to 99
Increment: 0.1
*RST: 40
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Start" on page 378

**MEASurement<m>:OPTical:OMAMplitude:RELative:STOP** <Stop>

Sets the end value of the range for the optical modulation amplitude.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Stop>
Range: 1 to 99
Increment: 0.1
*RST: 60
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Stop" on page 378

**MEASurement<m>:OPTical:UNIT** <Unit>

Selects the unit for the optical measurements.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Unit>
WATT | DBM
*RST: WATT

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 378

**MEASurement<m>:OPTical:LCHeck<n>:LOWer:LIMit** <LowerLimit>
**MEASurement<m>:OPTical:LCHeck<n>:UPPer:LIMit** <UpperLimit>

Define the lower/ upper limit for limit checking of the optical measurements.

The range is defined with `MEASurement<m>:OPTical:LCHeck<n>:VALid`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2

**Parameters:**
<UpperLimit> Range: -100 to 100
Increment: 10E-6
*RST: 0

**Usage:** Asynchronous command

**MEASurement<m>:OPTical:LCHeck<n>:LOWer:MARGin** <LowerMargin>
**MEASurement<m>:OPTical:LCHeck<n>:UPPer:MARGin** <UpperMargin>

Define the lower/ upper margin for limit checking of the optical measurements.

The range is defined with `MEASurement<m>:OPTical:LCHeck<n>:VALid`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2

**Parameters:**
<UpperMargin> Range: -100 to 100
Increment: 10E-6
*RST: 0

**Usage:** Asynchronous command

**MEASurement<m>:OPTical:LCHeck<n>:VALid** <ValidRange>

Selects the condition for the limit check of the optical measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2

**Parameters:**
<ValidRange> ILIMit | ULIMit | LLIMit | OLIMit

ILIMit: Inside (within) limit; between the upper and lower limit values
ULIMit: Upper limit; above the upper limit value
LLIMit : Lower limit; below the lower limit value
OLIMit: Outside limit; above the upper limit or below the lower limit values
*RST: ILIMit

**Usage:** Asynchronous command

## 24.12.8 Histograms

See also: Chapter 24.16.5, "Waveform histogram export to file", on page 1622.


- Histogram display....1476
- Histogram measurement....1482


### 24.12.8.1 Histogram display


LAYout:HISTogram:ADD.... 1476
LAYout:HISTogram:ENABle....1477
LAYout:HISTogram:SOURce.... 1478
LAYout:HISTogram:MODE.... 1478
LAYout:HISTogram:HORZ:MODE.... 1479
LAYout:HISTogram:HORZ:ABSolute:STARt.... 1479
LAYout:HISTogram:HORZ:ABSolute:STOP....1479
LAYout:HISTogram:HORZ:RELative:STARt....1480
LAYout:HISTogram:HORZ:RELative:STOP.... 1480
LAYout:HISTogram:VERTical:MODE.... 1480
LAYout:HISTogram:VERTical:ABSolute:STARt.... 1481
LAYout:HISTogram:VERTical:ABSolute:STOP....1481
LAYout:HISTogram:VERTical:RELative:STARt....1481
LAYout:HISTogram:VERTical:RELative:STOP.... 1482
LAYout:HISTogram:RESet....1482
LAYout:HISTogram:REMove....1482


---

**LAYout:HISTogram:ADD** <HistogramName>, <Source>, <XStart>, <XStop>, <YStart>, <YStop>, <Relative>, <Orientation>

Defines and displays a new histogram for the specified source.

Note: To define the mode of the histogram (vertical or horizontal), use the `LAYout:HISTogram:MODE` command.

For numerical values, the input of a unit is not supported for this command.

**Setting parameters:**

<HistogramName> String defining the histogram name which is used to refer to the histogram by other functions.

<Source> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | MRESult1 | MRESult2 | MRESult3 | MRESult4 | MRESult5 | MRESult6 | MRESult7 | MRESult8 | SG1TL1 | SG1TL2 | SG2TL1 | SG2TL2 | SG3TL1 | SG3TL2 | SG4TL1 | SG4TL2 | SG5TL2 | SG6TL1 | SG6TL2 | SG7TL1 | SG7TL2 | SG8TL1 | SG8TL2 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | EYE1 | EYE2 | EYE3 | EYE4 | SRHist1 | SDHist1 | TJHist1 | DJHist1 | DDJHist1 | PJOHist1 | RJOHist1 | SEYE1 | SRHist2 | SDHist2 | TJHist2 | DJHist2 | DDJHist2 | PJHOist2 | RJOHist2 | SEYE2 | TNHist1 | DNHist1 | DDNHist1 | PNHist1 | RNOHist1 | TNHist2 | DNHist2 | DDNHist2 | PNHist2 | RNOHist2 | PJDHist1 | BUJHist1 | RJHist1 | PJDHist2 | BUJHist2 | RJHist2 | PNDHist1 | BUNHist1 | RNHist1 | PNDHist2 | BUNHist2 | RNHist2

Data source of the histogram, see Chapter 24.4.2, "Waveform parameter", on page 1193

<XStart> Defines the start value of the x-value range.

<XStop> Defines the stop value of the x-value range.

<YStart> Defines the start value of the y-value range.

<YStop> Defines the stop value of the y-value range.

<Relative> ON | OFF

Defines whether relative or absolute values are used for the value range definition.

<Orientation> VERTical | HORizontal

**Example:** See "Creating and reading histograms" on page 1176

**Usage:** Setting only
Asynchronous command

---

**LAYout:HISTogram:ENABle** <HistogramName>,<State>
**LAYout:HISTogram:ENABle?** <HistogramName>

Enables or disables the histogram evaluation and display.

**Parameters:**

<State> ON | OFF

*RST: ON

**Parameters for setting and query:**

<HistogramName> String parameter, name of the histogram as defined with `LAYout:HISTogram:ADD`.

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 374

**LAYout:HISTogram:SOURce** <HistogramName>,<Source>
**LAYout:HISTogram:SOURce?** <HistogramName>

Defines the waveform which is the source of the histogram.

**Parameters:**

<Source> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | MRESult1 | MRESult2 | MRESult3 | MRESult4 | MRESult5 | MRESult6 | MRESult7 | MRESult8 | SG1TL1 | SG1TL2 | SG2TL1 | SG2TL2 | SG3TL1 | SG3TL2 | SG4TL1 | SG4TL2 | SG5TL2 | SG6TL1 | SG6TL2 | SG7TL1 | SG7TL2 | SG8TL1 | SG8TL2 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | EYE1 | EYE2 | EYE3 | EYE4 | SRHist1 | SDHist1 | TJHist1 | DJHist1 | DDJHist1 | PJOHist1 | RJOHist1 | SEYE1 | SRHist2 | SDHist2 | TJHist2 | DJHist2 | DDJHist2 | PJOHist2 | RJOHist2 | SEYE2 | TNHist1 | DNHist1 | DDNHist1 | PNHist1 | RNOHist1 | TNHist2 | DNHist2 | DDNHist2 | PNHist2 | RNOHist2 | PJDHist1 | BUJHist1 | RJHist1 | RJHist2 | PNDHist1 | BUNHist1 | RNHist1 | PNDHist2 | BUNHist2 | RNHist2

Waveform source of the histogram, see Chapter 24.4.2, "Waveform parameter", on page 1193.
If the histogram source is a measurement, the histogram shows the density distribution of the results for the selected measurement (`MEAS<m>:MAIN`).

**Parameters for setting and query:**

<HistogramName> String parameter

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 374

**LAYout:HISTogram:MODE** <HistogramName>,<Mode>
**LAYout:HISTogram:MODE?** <HistogramName>

Defines or queries the type of histogram.

**Parameters:**

<Mode> VERTical | HORizontal

**VERTical**
Amplitude histogram (horizontal bars across amplitude)

**HORizontal**
Time or frequency histogram (vertical bars over time/frequencies)

*RST: VERTical

**Parameters for setting and query:**
<HistogramName> The name of the histogram as defined using `LAYout:HISTogram:ADD` on page 1476.

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 374

---

**LAYout:HISTogram:HORZ:MODE** <HistogramName>,<Mode>
**LAYout:HISTogram:HORZ:MODE?** <HistogramName>

Defines or queries whether the value range limits are entered as absolute or relative values.

**Parameters:**
<Mode> ABS | REL
*RST: ABS

**Parameters for setting and query:**
<HistogramName>

**Usage:** Asynchronous command

**Manual operation:** See "Mode (Horizontal / Vertical)" on page 375

---

**LAYout:HISTogram:HORZ:ABSolute:STARt** <HistogramName>,<Start>
**LAYout:HISTogram:HORZ:ABSolute:STARt?** <HistogramName>

Defines the horizontal start value of the histogram.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**
<Start> Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0

**Parameters for setting and query:**
<HistogramName>

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop (Horizontal)" on page 375

---

**LAYout:HISTogram:HORZ:ABSolute:STOP** <HistogramName>,<Stop>
**LAYout:HISTogram:HORZ:ABSolute:STOP?** <HistogramName>

Defines the horizontal stop value of the histogram.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**
<Stop> Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**
<HistogramName>

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop (Horizontal)" on page 375

---

**LAYout:HISTogram:HORZ:RELative:STARt** <HistogramName>,<RelativeStart>
**LAYout:HISTogram:HORZ:RELative:STARt?** <HistogramName>

Defines the horizontal start value of the histogram.

**Parameters:**
<RelativeStart>
Range: 0 to 100
Increment: 0.1
*RST: 0
Default unit: %

**Parameters for setting and query:**
<HistogramName>

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop (Horizontal)" on page 375

---

**LAYout:HISTogram:HORZ:RELative:STOP** <HistogramName>,<RelativeStop>
**LAYout:HISTogram:HORZ:RELative:STOP?** <HistogramName>

Defines the horizontal stop value of the histogram.

**Parameters:**
<RelativeStop>
Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Parameters for setting and query:**
<HistogramName>

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop (Horizontal)" on page 375

---

**LAYout:HISTogram:VERTical:MODE** <HistogramName>,<Mode>
**LAYout:HISTogram:VERTical:MODE?** <HistogramName>

Defines or queries whether the value range limits are entered as absolute or relative values.

**Parameters:**
<Mode>
ABS | REL
*RST: ABS

**Parameters for setting and query:**
<HistogramName>

**Usage:** Asynchronous command

**Manual operation:** See "Mode (Horizontal / Vertical)" on page 375

---

**LAYout:HISTogram:VERTical:ABSolute:STARt** <HistogramName>,<Start>
**LAYout:HISTogram:VERTical:ABSolute:STARt?** <HistogramName>

Defines the vertical start value of the histogram.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**
<Start>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0

**Parameters for setting and query:**
<HistogramName>

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop (Vertical)" on page 375

---

**LAYout:HISTogram:VERTical:ABSolute:STOP** <HistogramName>,<Stop>
**LAYout:HISTogram:VERTical:ABSolute:STOP?** <HistogramName>

Defines the vertical stop value of the histogram.

For numerical values, the input of a unit is not supported for this command.

**Parameters:**
<Stop>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**
<HistogramName>

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop (Vertical)" on page 375

---

**LAYout:HISTogram:VERTical:RELative:STARt** <HistogramName>,<RelativeStart>
**LAYout:HISTogram:VERTical:RELative:STARt?** <HistogramName>

Defines the vertical start value of the histogram.

**Parameters:**
<RelativeStart>
Range: 0 to 100
Increment: 0.1
*RST: 0
Default unit: %

**Parameters for setting and query:**
<HistogramName>

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop (Vertical)" on page 375

---

**LAYout:HISTogram:VERTical:RELative:STOP** <HistogramName>,<RelativeStop>
**LAYout:HISTogram:VERTical:RELative:STOP?** <HistogramName>

Defines the vertical stop value of the histogram.

**Parameters:**
<RelativeStop>
Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Parameters for setting and query:**
<HistogramName>

**Usage:** Asynchronous command

**Manual operation:** See "Start, Stop (Vertical)" on page 375

---

**LAYout:HISTogram:RESet** <HistogramName>

Resets the values to begin a new histogram.

**Setting parameters:**
<HistogramName>

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Reset" on page 374

---

**LAYout:HISTogram:REMove** <Name>

Removes the specified histogram.

**Setting parameters:**
<Name>

**Usage:** Setting only
Asynchronous command

### 24.12.8.2 Histogram measurement

This chapter lists commands to set up measurements on histograms.

Note that the R&S RTP performs histogram measurements on the measurement that is defined with `MEASurement<m>:MAIN`. Other measurements are not considered.

See also: "Creating and reading histograms" on page 1176.

The following table lists the <MeasType> parameter values with a short description.

In addition, the suffixes <n> for limit-check remote commands are listed.

For a detailed description, see Table 8-8.

***Table 24-10: Histogram measurements***

| **<MeasType> parameter value (remote control)** | **Measurement (manual operation)** | **Description/Result** | **Suffix <n>** |
|---|---|---|---|
| WCOunt | Waveform count | Number of acquisitions (waveforms) the histogram is based on | 1 |
| WSAMples | Waveform samples | Number of samples from the most recent acquisition included in the current histogram | 2 |
| HSAMples | Histogram samples | Number of samples from all acquisitions included in the current histogram | 3 |
| HPEak | Histogram peak | Maximum sample value in the histogram | 4 |
| PEAK | Peak value | Signal value at the histogram peak | 5 |
| UPEakvalue | Upper peak value | Signal value at the maximum sample value in the upper half of the histogram | 6 |
| LPEakvalue | Lower peak value | Signal value at the maximum sample value in the lower half of the histogram | 7 |
| HMAXimum | Maximum | Highest signal value with a probability > 0 | 8 |
| HMINimum | Minimum | Lowest signal value with a probability > 0 | 9 |
| MEDian | Median | Signal value for which half the samples lie above, the other half below in the histogram | 10 |
| MAXMin | Max - Min | Range of signal values with a probability > 0 | 11 |
| HMEan | Mean | Weighted arithmetic average of the histogram | 12 |
| HSTDdev | σ (S-dev) | Standard deviation of the sample numbers | 13 |
| M1STddev | Mean ±σ | Range between (mean value + standard deviation) and (mean value - standard deviation) | 14 |
| M2STddev | Mean ±2*σ | Range between (mean value + 3 * standard deviation) and (mean value - 2 * standard deviation) | 15 |
| M3STddev | Mean ±3*σ | Range between (mean value + 3 * standard deviation) and (mean value - 2 * standard deviation) | 16 |
| MKPositive | Marker + Probability % | Marker value (according to the selected probability domain marker type) plus the defined limit.<br>Note that the value is restricted to the histogram range. | 17 |
| MKNegative | Marker - Probability % | Marker value (according to the selected probability domain marker type) minus the defined limit.<br>Note that the value is restricted to the histogram range. | 18 |

MEASurement<m>:HISTogram:SELect ..... 1484
MEASurement<m>:HISTogram:PROBability:TYPE ..... 1484
MEASurement<m>:HISTogram:PROBability:LIMit ..... 1485
MEASurement<m>:HISTogram:LCHeck<n>:VALid ..... 1485
MEASurement<m>:HISTogram:LCHeck<n>:LOWer:LIMit ..... 1486
MEASurement<m>:HISTogram:LCHeck<n>:UPPer:LIMit ..... 1486
MEASurement<m>:HISTogram:LCHeck<n>:LOWer:MARGin ..... 1486
MEASurement<m>:HISTogram:LCHeck<n>:UPPer:MARGin ..... 1486

**MEASurement<m>:HISTogram:SELect** <HistgRef>

Selects the histogram on which the measurement is based.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<HistgRef> String with the name of the histogram

**Example:** See "Creating and reading histograms" on page 1176

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 376

**MEASurement<m>:HISTogram:PROBability:TYPE** <HistgPrbDomMark>

Defines the marker reference in the probability domain.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<HistgPrbDomMark> PEAK | UPPK | LWPK | MAXimum | MINimum | MEDian | MEAN

**PEAK**
The y-value with the maximum sample value in the histogram

**UPPK**
The y-value at the maximum sample value in the upper half of the histogram

**LWPK**
The y-value at the maximum sample value in the lower half of the histogram

**MAXimum**
The highest y-value with a probability > 0

**MINimum**
The lowest y-value with a probability > 0

**MEDian**
The y-value for which half the samples lie above, the other half below in the histogram

**MEAN**
The weighted arithmetic average of the histogram

*RST: PEAK

**Usage:** Asynchronous command

**Manual operation:** See "Probability domain marker, Reference" on page 376

---

**MEASurement<m>:HISTogram:PROBability:LIMit** <HistgPrbDomLim>

Defines a range around the probability marker.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<HistgPrbDomLim> Range: 0 to 100
Increment: 10
*RST: 10
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Probability domain marker, Delta" on page 376

---

**MEASurement<m>:HISTogram:LCHeck<n>:VALid** <ValidRange>

Enables or disables limit checking for histogram measurements in the specified measurement channel.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..18
Number of the histogram measurement, see Table 24-10.

**Parameters:**
<ValidRange> ILIMit | ULIMit | LLIMit | OLIMit

**ILIMit**
Inside (within) limit; between the upper and lower limit values

**ULIMit**
Upper limit; above the upper limit value

**LLIMit**
Lower limit; below the lower limit value

**OLIMit**
Outside limit; above the upper limit or below the lower limit values

*RST: ILIMit

**Usage:** Asynchronous command

**Manual operation:** See "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394

---

**MEASurement<m>:HISTogram:LCHeck<n>:LOWer:LIMit** <LowerLimit>
**MEASurement<m>:HISTogram:LCHeck<n>:UPPer:LIMit** <UpperLimit>

Define the lower and upper limits for the limit check, respectively. The valid range is defined using the `MEASurement<m>:HISTogram:LCHeck<n>:VALid` command.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..18
Number of the histogram measurement, see Table 24-10.

**Parameters:**
<UpperLimit> Range: -100 to 100
Increment: 10E-6
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394

---

**MEASurement<m>:HISTogram:LCHeck<n>:LOWer:MARGin** <LowerMargin>
**MEASurement<m>:HISTogram:LCHeck<n>:UPPer:MARGin** <UpperMargin>

Define the lower and upper margins for the limit check, respectively. Margins are not as strict as limits and must be within the valid value range. The valid range is defined using the `MEASurement<m>:HISTogram:LCHeck<n>:VALid` command.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..18
Number of the histogram measurement, see Table 24-10.

**Parameters:**
<UpperMargin> Range: -100 to 100
Increment: 10E-6
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Upper limit, Lower limit, Upper margin, Lower margin, Valid range" on page 394

### 24.12.9 Display

MEASurement<m>:DISPlay:LEVels ..... 1487
MEASurement<m>:DISPlay:RESults ..... 1487
MEASurement<m>:DISPlay:STYLe ..... 1488
MEASurement<m>:DISPlay:HISTogram ..... 1488

**MEASurement<m>:DISPlay:LEVels** <DisplayLevels>

If enabled, the reference levels used for the measurement are displayed in the diagram.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<DisplayLevels> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Reference levels" on page 328

**MEASurement<m>:DISPlay:RESults** <DisplayResult>

If enabled, the intermediate result linesare displayed in the measurement diagram.

Result lines mark the samples in the waveform that are used to obtain the measurement result. These are, for example, maximum and minimum values, mean, cycle start and cycle end, and others.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<DisplayResult> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Result lines" on page 328

**MEASurement<m>:DISPlay:STYLe** <DisplayStyle>

Selects the style in which the measurement waveform is displayed.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<DisplayStyle> LINE | MARKer

**LINE**
The individual data points are connected by a line.

**MARKer**
Only the individual data points are displayed as markers.

*RST: LINE

**Usage:** Asynchronous command

**MEASurement<m>:DISPlay:HISTogram** <DispHistg>

Displays a histogram for the source of the selected measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<DispHistg> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

## 24.12.10 Statistics and long-term measurements

See also: Chapter 24.16.7, "Long term measurement results and measurement histogram export to file", on page 1625.

MEASurement<m>:STATistics[:ENABle] 1489
MEASurement<m>:CLEar 1489
MEASurement<m>:MULTiple 1489
MEASurement<m>:MNOMeas 1490
MEASurement<m>:STATistics:HISTogram 1490
MEASurement<m>:STATistics:HBINs 1490
MEASurement<m>:STATistics:MODE 1491
MEASurement<m>:STATistics:RCOunt 1491
MEASurement<m>:STATistics:RMEascount 1492
MEASurement<m>:STATistics:RTIMe 1492
MEASurement<m>:STATistics:RESet 1492
MEASurement<m>:VERTical:CONT 1493

MEASurement<m>:VERTical:AUTO....1493
MEASurement<m>:VERTical:OFFSet....1493
MEASurement<m>:VERTical:SCALe....1494
MEASurement<m>:LTMeas[:STATe]....1494
MEASurement<m>:LTMeas:COUNt....1494
MEASurement<m>:LTMeas:TIME....1495

**MEASurement<m>:STATistics[:ENABle]** <StatisticsState>

Enables statistics calculation for the measurement.

For details on the statistics results, see Chapter 8.2.12.1, "Statistics", on page 384.

**Suffix:**
<m>

**Parameters:**
<StatisticsState> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Statistics" on page 325

**MEASurement<m>:CLEar**

Deletes the statistic results of the indicated mesurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Usage:** Event
Asynchronous command

**MEASurement<m>:MULTiple** <MultiMeas>

The measurement is performed repeatedly if the measured parameter occurs several times inside the acquisition or defined gate.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<MultiMeas> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Measure all events in each acquisition" on page 391

**MEASurement<m>:MNOMeas** <MaxMeasPerAcq>

Sets the maximum number of measurements per acquisition if repeated measurement is enabled (`MEASurement<m>:MULTiple` is `ON`).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<MaxMeasPerAcq> Range: 2 to 1000000
Increment: 1
*RST: 1000

**Usage:** Asynchronous command

**Manual operation:** See "Measure all events in each acquisition" on page 391

**MEASurement<m>:STATistics:HISTogram** <ShowHistogram>

Displays a histogram of the statistical results. Enabling the histogram enables also the calculation and display of statistics for the measurement results if statistics were disabled. the histogram shows the cumulative occurence distribution of mean measurement results in a graphic.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ShowHistogram> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable (Histogram)" on page 389

**MEASurement<m>:STATistics:HBINs** <HistogramBins>

Sets the number of bins - the number of vertical bars that build the histogram.

If `MEASurement<m>:VERTical:CONT` is ON, the instrument determines the number of bins automatically based on the time base, the current measurements, and other settings.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<HistogramBins> Range: 2 to 1000
Increment: 10
*RST: 1000

**Usage:** Asynchronous command

**Manual operation:** See "Number of bins" on page 389

---

**MEASurement<m>:STATistics:MODE** <ResetMode>

Defines when the statistics for long term measurements are reset.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ResetMode> TIME | WFMS | MEAS

**TIME**
Sets one long term measurement point after the time defined using `MEASurement<m>:STATistics:RTIMe`.

**WFMS - Waveforms**
Sets one long term measurement point after a number of acquired waveforms defined using `MEASurement<m>:STATistics:RCOunt`.

**MEAS**
Sets one long term measurement point after a number of measurement results.

*RST: TIME

**Usage:** Asynchronous command

**Manual operation:** See "Scale mode" on page 391

---

**MEASurement<m>:STATistics:RCOunt** <RstWfmCnt>

Defines the number of measured waveforms from which one point of the long term measurement is created (reset of statistics).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<RstWfmCnt> Range: 1 to 65535
Increment: 10
*RST: 1000

**Usage:** Asynchronous command

**Manual operation:** See "Wfms / point" on page 392

---

**MEASurement<m>:STATistics:RMEascount** <RstMeasCnt>

Defines the number of measurement results from which one point of the long term measurement is created.

This setting is only available if `MEASurement<m>:STATistics:MODE` is set to `MEAS`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<RstMeasCnt>
Range: 1 to 65535
Increment: 10
*RST: 1000

**Usage:** Asynchronous command

**Manual operation:** See "Measurements / point" on page 392

---

**MEASurement<m>:STATistics:RTIMe** <ResetTime>

Defines the time or period after which the statistics are reset.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ResetTime>
Range: 0.1 to 2.14748E+9
Increment: 1E-3
*RST: 0.2
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Time / point" on page 392

---

**MEASurement<m>:STATistics:RESet**

Resets the histogram, the long term measurement and the statistics.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Reset" on page 390

---

**MEASurement<m>:VERTical:CONT** <ContAutoScl>

If enabled, automatic vertical scaling is performed whenever the waveform does not fit in the diagram during the long term measurement period.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ContAutoScl> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Analysis scale" on page 390

---

**MEASurement<m>:VERTical:AUTO**

Performs an automatic scaling once so that the scaling is adapted to the current measurement results. Available only for long term measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Analysis scale" on page 390

---

**MEASurement<m>:VERTical:OFFSet** <VerticalOffset>

Defines a vertical offset for the long term measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<VerticalOffset> Range: -100E+12 to 100E+12
Increment: 1E-6
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Analysis scale" on page 390

**MEASurement<m>:VERTical:SCALe** <VerticalScale>

Defines the vertical scaling per division, so that the scaling can be adapted automatically during the long term measurement period.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<VerticalScale> Range: 1E-12 to 100E+12
Increment: 10E-6
*RST: 0.5
Default unit: V/div

**Usage:** Asynchronous command

**Manual operation:** See "Analysis scale" on page 390

**MEASurement<m>:LTMeas[:STATe]** <ShowDiagram>

Enables long term measurement for a defined number of measurement points (see `MEASurement<m>:LTMeas:COUNt` on page 1494) or a specified time (see`MEASurement<m>:LTMeas:TIME` on page 1495).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ShowDiagram> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable (Long term)" on page 388

**MEASurement<m>:LTMeas:COUNt** <MeasCnt>

Defines the total number of points to be measured during the long term measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<MeasCnt> Range: 1000 to 200000
Increment: 10
*RST: 1000

**Usage:** Asynchronous command

**Manual operation:** See "Number of points" on page 391

---

**MEASurement<m>:LTMeas:TIME** <MeasurementTime>

Defines the total duration of the long term measurement.

This setting is only available if `MEASurement<m>:STATistics:MODE` is set to "Time".

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<MeasurementTime> Range: 0.01 to 2.14748E+9
Increment: 1
*RST: 200
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Measurement time" on page 392

## 24.12.11 Protocol

Before starting the protocol measurement, a bus type has to be selected (`BUS<m>:TYPE`) and the bus has to be activated (`BUS<m>[:STATe]`).

The following table lists the <MeasType> parameter values with a short description.

***Table 24-11: Protocol measurements***

| <MeasType> parameter value (remote control) | Measurement (manual operation) | Description/Result |
|---|---|---|
| AUDio | Audio | |
| FLDValue | Field value | Value of a field over time |
| F2F | Frame to frame | The distance between two frame types |
| | Trigger to frame | |
| GAP | Gap | Measures a gap, periods at which the bus is idle. The distance of a gap can only be measured between two identified frames. |
| MBITrate | Main bit rate | Bit rate of the protocol |
| SBITrate | 2nd bit rate | Additional bit rate, protocol dependent. |
| BIDLe | Bus idle | Calculates the bus idle time |
| FCNT | Frame count | Number of all frames within the acquisition window |
| FEC | Frame error count | Sum of all frames with errors within the acquisition window |

| <MeasType> parameter value (remote control) | Measurement (manual operation) | Description/Result |
|---|---|---|
| FER | Frame error rate | Sum of all frames with errors divided by all frames within the acquisition window |
| CFER | Consecutive frame error rate | Measures the frame error rate |

MEASurement<m>:PROTocol:F2FRame:FLDFrom ... 1496
MEASurement<m>:PROTocol:F2FRame:FLDTo ... 1496
MEASurement<m>:PROTocol:F2FRame:FRMFrom ... 1497
MEASurement<m>:PROTocol:F2FRame:FRMTo ... 1497
MEASurement<m>:PROTocol:F2FRame:VALFrom ... 1497
MEASurement<m>:PROTocol:F2FRame:VALTo ... 1497
MEASurement<m>:PROTocol:FLDValue:FLD ... 1498
MEASurement<m>:PROTocol:FLDValue:FRM ... 1498
MEASurement<m>:PROTocol:FLDValue:TRCK ... 1498
MEASurement<m>:PROTocol:FLDValue:VAL ... 1499
MEASurement<m>:PROTocol:MBITrate:FLD ... 1499
MEASurement<m>:PROTocol:MBITrate:FRM ... 1499
MEASurement<m>:PROTocol:MBITrate:VAL ... 1499
MEASurement<m>:PROTocol:SBITrate:FLD ... 1500
MEASurement<m>:PROTocol:SBITrate:FRM ... 1500
MEASurement<m>:PROTocol:SBITrate:VAL ... 1500
MEASurement<m>:PROTocol:T2FRame:DIRection ... 1500
MEASurement<m>:PROTocol:T2FRame:FLD ... 1501
MEASurement<m>:PROTocol:T2FRame:FRM ... 1501
MEASurement<m>:PROTocol:T2FRame:VALue ... 1501

---

**MEASurement<m>:PROTocol:F2FRame:FLDFrom** <FieldName>

Sets the type of field for the start frame of the frame to frame protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FieldName>

**Manual operation:** See "Frame to frame" on page 380

---

**MEASurement<m>:PROTocol:F2FRame:FLDTo** <FieldName>

Sets the type of field for the end frame of the frame to frame protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FieldName>

**Manual operation:** See "Frame to frame" on page 380

---

**MEASurement<m>:PROTocol:F2FRame:FRMFrom** <FrameName>

Sets the type of start frame for the frame to frame protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FrameName>

**Manual operation:** See "Frame to frame" on page 380

---

**MEASurement<m>:PROTocol:F2FRame:FRMTo** <FrameName>

Sets the type of end frame for the frame to frame protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FrameName>

**Manual operation:** See "Frame to frame" on page 380

---

**MEASurement<m>:PROTocol:F2FRame:VALFrom** <FrmFrmFromVal>

Sets the field value of the start frame for the frame to frame protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FrmFrmFromVal>

**Usage:** Asynchronous command

**Manual operation:** See "Frame to frame" on page 380

---

**MEASurement<m>:PROTocol:F2FRame:VALTo** <FrmFrmVal>

Sets the field value of the end frame for the frame to frame protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FrmFrmVal>

**Usage:** Asynchronous command

**Manual operation:** See "Frame to frame" on page 380

---

**MEASurement<m>:PROTocol:FLDValue:FLD** <FieldName>

Sets the type of field for the frame of the field value protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FieldName>

**Manual operation:** See "Field value" on page 383

---

**MEASurement<m>:PROTocol:FLDValue:FRM** <FrameName>

Sets the type frame for the field value protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FrameName>

**Manual operation:** See "Field value" on page 383

---

**MEASurement<m>:PROTocol:FLDValue:TRCK** <FieldName>

Sets the field to be tracked value of the frame for the field value protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FieldName>

**Manual operation:** See "Field value" on page 383

**MEASurement<m>:PROTocol:FLDValue:VAL** <FieldValIdVal>

Sets the field value of the frame for the field value protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FieldValIdVal>

**Usage:** Asynchronous command

**Manual operation:** See "Field value" on page 383

**MEASurement<m>:PROTocol:MBITrate:FLD** <FieldName>

Sets the type of field for the frame of the main bit rate protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FieldName>

**Manual operation:** See "Main bit rate / 2nd bit rate" on page 383

**MEASurement<m>:PROTocol:MBITrate:FRM** <FrameName>

Sets the type frame for the main bit rate protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FrameName>

**Manual operation:** See "Main bit rate / 2nd bit rate" on page 383

**MEASurement<m>:PROTocol:MBITrate:VAL** <MainBtRateIdVal>

Sets the field value of the frame for the main bit rate protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<MainBtRateIdVal>

**Usage:** Asynchronous command

**Manual operation:** See "Main bit rate / 2nd bit rate" on page 383

---

**MEASurement<m>:PROTocol:SBITrate:FLD** <FieldName>

Sets the type of field for the frame of the secondary bit rate protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FieldName>

**Manual operation:** See "Main bit rate / 2nd bit rate" on page 383

---

**MEASurement<m>:PROTocol:SBITrate:FRM** <FrameName>

Sets the type frame for the secondary bit rate protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<FrameName>

**Manual operation:** See "Main bit rate / 2nd bit rate" on page 383

---

**MEASurement<m>:PROTocol:SBITrate:VAL** <SecondaryBtRateIdVal>

Sets the field value of the frame for the secondary bit rate protocol measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<SecondaryBtRateIdVal>

**Usage:** Asynchronous command

**Manual operation:** See "Main bit rate / 2nd bit rate" on page 383

---

**MEASurement<m>:PROTocol:T2FRame:DIRection** <TrigFrmDirct>

Sets the direction of the field for trigger to frame protocol measurement.

**Suffix:**
<m> 1..10

**Parameters:**
<TrigFrmDirct> T2F | F2T
`T2F`: trigger to frame
`F2T`: frame to trigger
*RST: T2F

**Usage:** Asynchronous command

**Manual operation:** See "Trigger to frame" on page 382

---

**MEASurement<m>:PROTocol:T2FRame:FLD** <FieldName>

Sets the type of the field for trigger to frame protocol measurement.

**Suffix:**
<m> 1..10

**Parameters:**
<FieldName>

**Manual operation:** See "Trigger to frame" on page 382

---

**MEASurement<m>:PROTocol:T2FRame:FRM** <FrameName>

Sets the value of the frame for trigger to frame protocol measurement.

**Suffix:**
<m> 1..10

**Parameters:**
<FrameName>

**Manual operation:** See "Trigger to frame" on page 382

---

**MEASurement<m>:PROTocol:T2FRame:VALue** <TrigFrmVal>

Sets the value of the field for trigger to frame protocol measurement.

**Suffix:**
<m> 1..10

**Parameters:**
<TrigFrmVal>

**Usage:** Asynchronous command

**Manual operation:** See "Trigger to frame" on page 382

## 24.12.12 Track

MEASurement<m>:TRACk[:STATe] ..... 1502
MEASurement<m>:TRACk:DATA:HEADer? ..... 1502
MEASurement<m>:TRACk:DATA:STYPe? ..... 1502
MEASurement<m>:TRACk:DATA[:VALues]? ..... 1502

**MEASurement<m>:TRACk[:STATe]** <State>

Enables the track functionality and displays the track.

The track functionality requires at least one option, see "Enable (Track)" on page 390.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable (Track)" on page 390

**MEASurement<m>:TRACk:DATA:HEADer?**

Returns the header of the track.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Usage:** Query only

**MEASurement<m>:TRACk:DATA:STYPe?**

Returns the data type: TRK (track).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Usage:** Query only

**MEASurement<m>:TRACk:DATA[:VALues]?**

Returns the data of track points for transmission from the instrument to the controlling computer. The data can be used in MATLAB, for example.

To set the export format, use `FORMat[:DATA]`.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Usage:** Query only

### 24.12.13 Gating

MEASurement<m>:GATE[:STATe] ..... 1503
MEASurement<m>:GATE:MODE ..... 1503
MEASurement<m>:GATE:ABSolute:STARt ..... 1504
MEASurement<m>:GATE:ABSolute:STOP ..... 1504
MEASurement<m>:GATE:RELative:STARt ..... 1504
MEASurement<m>:GATE:RELative:STOP ..... 1504
MEASurement<m>:GATE:CURSor ..... 1504
MEASurement<m>:GATE:CCOupling ..... 1505
MEASurement<m>:GATE:ZCOupling ..... 1505
MEASurement<m>:GATE:ZDIagram ..... 1506
MEASurement<m>:GATE:GCOupling ..... 1506

---

**MEASurement<m>:GATE[:STATe]** <State>

Considers the gating settings of the source waveform for the measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Use gate" on page 330

---

**MEASurement<m>:GATE:MODE** <Mode>

Defines whether the gate settings are configured using absolute or relative values.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Mode> ABS | REL
*RST: ABS

**Usage:** Asynchronous command

**Manual operation:** See "Gate description" on page 330

---

**MEASurement<m>:GATE:ABSolute:STARt** <Start>
**MEASurement<m>:GATE:ABSolute:STOP** <Stop>

Define the absolute start and end values for the gate.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Stop>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Usage:** Asynchronous command

**Manual operation:** See "(Relative) Stop" on page 331

---

**MEASurement<m>:GATE:RELative:STARt** <RelativeStart>
**MEASurement<m>:GATE:RELative:STOP** <RelativeStop>

Define the relative start and end values for the gate, respectively.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<RelativeStop>
Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "(Relative) Stop" on page 331

---

**MEASurement<m>:GATE:CURSor** <Cursorset>

Selects the cursor set to be used for measurement gating. The gate area is defined by the cursor lines.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**

<Cursorset> CURSOR1 | CURSor1 | CURSOR2 | CURSor2 | CURSOR3 | CURSor3 | CURSOR4 | CURSor4

CURSOR1 = CURSor1, CURSOR2 = CURSor2, CURSOR3 = CURSor3, CURSOR4 = CURSor4

*RST: CURSOR1

**Usage:** Asynchronous command

**Manual operation:** See "Cursor" on page 331

---

**MEASurement<m>:GATE:CCOupling** <CursorCoupling>

Enables the cursor coupling for automatic measurements.

Select the cursor set to be used with `MEASurement<m>:GATE:CURSor`.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**

<CursorCoupling> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Cursor" on page 331

---

**MEASurement<m>:GATE:ZCOupling** <ZoomCoupling>

If enabled, the gate area is defined identically to the zoom area for the zoom diagram.

If enabled, define the zoom area to be used as gate with `MEASurement<m>:GATE:ZDIagram`.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**

<ZoomCoupling> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Zoom" on page 331

**MEASurement<m>:GATE:ZDIagram** <DiagramName>, <ZoomName>

If `MEASurement<m>:GATE:ZCOupling` is enabled, the gate area is defined identically to the zoom area for the selected zoom diagram.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<ZoomDiagram> String with the name of the diagram on which the zoom is based

**Example:** Prerequisite: Zoom 1 is available
```
MEASurement:GATE:ZCOupling ON
MEASurement:GATE:ZDIagram "Diagram1","Zoom1"
```

**Usage:** Asynchronous command

**Manual operation:** See "Zoom" on page 331

**MEASurement<m>:GATE:GCOupling** <GateCoupling>

If you enable the gate coupling, the gate settings of the selected measurement are copied to all other measurements. If zoom or cursor coupling is active in a measurement, the zoom size and cursor positions are adjusted.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<GateCoupling> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Apply to all meas" on page 330

## 24.12.14 Limit check and event actions

MEASurement<m>:LCHeck ..... 1506
MEASurement<m>:ONViolation:BEEP ..... 1507
MEASurement<m>:ONViolation:ACQStop ..... 1507
MEASurement<m>:ONViolation:WFMSave ..... 1508
MEASurement<m>:ONViolation:REPort ..... 1508
MEASurement<m>:ONViolation:TRIGgerout ..... 1508
MEASurement<m>:ONViolation:RUNexec ..... 1509

**MEASurement<m>:LCHeck** <LimitCheckState>

Defines the type of the limit check that can run together with the measurement.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<LimitCheckState> OFF | LIMit | LMARgin

**OFF**
No limit check.

**LIMit**
Only limits are checked.

**LMARgin**
Limits and margins are checked.

*RST: OFF

**Usage:** Asynchronous command

---

**MEASurement<m>:ONViolation:BEEP** <Beep>

Generates a beep sound for the specified event.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Beep> NOACtion | SUCCess | VIOLation
See Chapter 24.4.5, "Event parameter", on page 1195
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Beep" on page 395

---

**MEASurement<m>:ONViolation:ACQStop** <StopAcq>

Stops data acquisition for the specified event.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<StopAcq> NOACtion | SUCCess | VIOLation
See Chapter 24.4.5, "Event parameter", on page 1195
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Stop acq" on page 396

**MEASurement<m>:ONViolation:WFMSave** <SaveWfm>

Saves the waveform data using the default RefCurve filename.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<SaveWfm> NOACtion | SUCCess | VIOLation
See Chapter 24.4.5, "Event parameter", on page 1195
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Save Wfm" on page 396

**MEASurement<m>:ONViolation:REPort** <Report>

Creates and saves a report of the current settings and results.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<Report> NOACtion | SUCCess | VIOLation
See Chapter 24.4.5, "Event parameter", on page 1195
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Report" on page 396

**MEASurement<m>:ONViolation:TRIGgerout** <TriggerOutPulse>

Creates a trigger out pulse on limit or margin violation or on successful completion of the measurement.

**Suffix:**
<m> 1..10

**Parameters:**
<TriggerOutPulse> NOACtion | SUCCess | VIOLation
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Trigger Out Pulse" on page 396

**MEASurement<m>:ONViolation:RUNexec** <RunExecutable>

Starts an external application if the command is set to `VIOLation` or `SUCCess`.

Use the following commands to set up the application:

- `EXECutable:NAME` on page 1635
- `EXECutable:PARameter` on page 1635
- `EXECutable:WDIRectory` on page 1636

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<RunExecutable> NOACtion | SUCCess | VIOLation
See Chapter 24.4.5, "Event parameter", on page 1195
*RST: NOACtion

**Usage:** Asynchronous command

**Manual operation:** See "Start Executable" on page 396

## 24.12.15 Reference levels


- General reference level settings ..... 1509
- Automatic configuration ..... 1512
- Manual configuration ..... 1514
- Hysteresis ..... 1519
- Tube ..... 1519
- Results ..... 1521


### 24.12.15.1 General reference level settings


REFLevel<m>:LDETection ..... 1509
REFLevel<m>:LMODe ..... 1510
REFLevel<m>:RELative:MODE ..... 1510
REFLevel<m>:USRLevel ..... 1511


**REFLevel<m>:LDETection** <Mode>

Defines whether the reference level is configured manually or automatically.

For automatic configuration, select the signal level to be used (see `REFLevel<m>:AUTO:MODE` on page 1512).

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Mode> AUTO | MANual

**Example:** `REFLevel2:LDETection MANual`
Sets manual level configuration for Ch1Wfm1. C1W1 corresponds to suffix number 2.

**Example:** See: Chapter 24.3.2.2, "Setting reference levels", on page 1175

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 336

---

**REFLevel<m>:LMODe** <Mode>

Defines whether the reference is configured using absolute or relative values.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Mode> ABS | REL

**Example:** `REFLevel2:LMODe ABS`
Sets definition of reference levels to absolute values for Ch1Wfm1. C1W1 corresponds to suffix number 2.

**Example:** See: "Manual reference level definition using relative values" on page 1175

**Usage:** Asynchronous command

**Manual operation:** See "Level definition" on page 337

---

**REFLevel<m>:RELative:MODE** <Mode>

The lower, middle and upper reference levels, defined as percentages of the high signal level.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Mode> FIVE | TEN | TWENty | USER

**FIVE**
5/50/95

**TEN**
10/50/90

**TWENty**
20/50/80

**USER**
Set the reference levels to individual values with `REFLevel<m>:RELative:LOWer`, `REFLevel<m>:RELative:MIDDle`, and `REFLevel<m>:RELative:UPPer`.

**Example:** `REFL2:REL:MODE FIVE`
Reference levels for Ch1Wfm1: Lower reference level = 5% of high signal level, middle reference level = 50% of high signal level, upper reference level = 95% of high signal level

**Example:** See: "Manual reference level definition using relative values" on page 1175

**Usage:** Asynchronous command

**Manual operation:** See "Relative levels" on page 338

---

**REFLevel<m>:USRLevel** <Mode>

Defines whether the user-defined signal levels or user-defined reference levels are used for the measurements.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Mode> USIGnal | UREF

**USIGnal**
The high and low signal levels are defined by the user.

**UREF**
The reference levels are defined by the user.

**Example:** `REFLevel2:USRLevel UREF`
Sets user-defined reference levels to be used for Ch1Wfm1. C1W1 corresponds to suffix number 2.

**Example:** See: "Manual reference level definition using absolute values" on page 1176

**Usage:** Asynchronous command

**Manual operation:** See "User level" on page 337

### 24.12.15.2 Automatic configuration

REFLevel<m>:AUTO:MODE ..... 1512
REFLevel<m>:AUTO[:STATe] ..... 1513
REFLevel<m>:AUTO:COUNt ..... 1513

---

**REFLevel<m>:AUTO:MODE** <Mode>

Defines the high and low signal levels from which the reference levels are derived.

This setting is only available for automatic reference level mode (see `REFLevel<m>:LDETection` on page 1509).

**Suffix:**

<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**

<Mode> AUTO | PPRobability | MPRobability | ABSolutepeak | UPLM | UMLP | UALM | UMLA

**AUTO**
Auto select absolute probability: most suitable signal levels for the selected measurement

**PPRobability**
Peak probability: signal levels with the highest probability value

**MPRobability**
Mean probability: signal levels with mean probability

**ABSolutepeak**
Absolute peak: absolute peak signal levels

**UPLM**
Upper absolute peak, lower mean probability: high signal level is the upper absolute peak, low signal level is the level with the mean probability in the lower half of the histogram.

**UMLP**
Upper mean probability, lower absolute peak: high signal level is the level with mean probability in the upper half of the histogram, low signal level is the lower absolute peak.

**UALM**
Upper absolute peak, lower manual value: high signal level is the maximum result value of the amplitude measurement; low signal level is manually set using `REFLevel<m>:RELative:LOWer`.

**UMLA**
Upper manual, lower absolute peak: The upper signal level is set manually using `REFLevel<m>:RELative:UPPer`. The lower level is the minimum result value of the amplitude measurement.

**Example:** `REFLevel5:AUTO:MODE PPRobability`
Sets the automatic reference level mode for Ch2Wfm1 to "Peak probability". C2W1 corresponds to suffix number 5.

**Example:** See: "Automatic level detection, peak probability" on page 1176

**Usage:** Asynchronous command

**Manual operation:** See "Signal level" on page 337

---

**REFLevel<m>:AUTO[:STATe]** <Value>

Enables averaging over several histograms to determine the reference levels. The number of histograms to consider is defined using `REFLevel<m>:AUTO:COUNt`.

This function is only available in automatic reference level mode (see `REFLevel<m>:LDETection` on page 1509).

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value> ON | OFF

**Usage:** Asynchronous command

**Manual operation:** See "Histogram averaging" on page 339

---

**REFLevel<m>:AUTO:COUNt** <Value>

Defines the number of histograms from which the average is calculated.

Prerequisites:

- `REFLevel<m>:AUTO[:STATe]` is set to `ON`
- `REFLevel<m>:LDETection` on page 1509 is set to `AUTO`

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Usage:** Asynchronous command

**Manual operation:** See "Average Count" on page 339

### 24.12.15.3 Manual configuration

REFLevel<m>:ABSolute:HIGH ..... 1514
REFLevel<m>:ABSolute:LOW ..... 1514
REFLevel<m>:ABSolute:TDIStance ..... 1515
REFLevel<m>:ABSolute:BDIStance ..... 1515
REFLevel<m>:ABSolute:MLEVel ..... 1516
REFLevel<m>:ABSolute:ULEVel ..... 1516
REFLevel<m>:ABSolute:LLEVel ..... 1517
REFLevel<m>:RELative:UPPer ..... 1517
REFLevel<m>:RELative:MIDDle ..... 1518
REFLevel<m>:RELative:LOWer ..... 1518

**REFLevel<m>:ABSolute:HIGH** <Value>

The signal value that represents a high level.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Example:** `REFLevel2:ABSolute:HIGH 0.015`
Sets the high signal level for Ch1Wfm1 to 15 mV. C1W1 corresponds to suffix number 2.

**Usage:** Asynchronous command

**Manual operation:** See "High" on page 338

**REFLevel<m>:ABSolute:LOW** <Value>

The signal value that represents a low level.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Example:** `REFLevel2:ABSolute:Low 0.0015`
Sets the low signal level forCh1Wfm1 to 1.5 mV. C1W1 corresponds to suffix number 2.

**Usage:** Asynchronous command

**Manual operation:** See "Low" on page 338

---

**REFLevel<m>:ABSolute:TDIStance** <Value>

The distance between the high signal level and the upper reference level.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Example:** `REFLevel2:ABSolute:TDIStance 0.0002`
Sets the top distance for Ch1Wfm1 to 0.2 mV. C1W1 corresponds to suffix number 2.

**Example:** See: "Manual reference level definition using absolute values" on page 1176

**Usage:** Asynchronous command

**Manual operation:** See "Top distance" on page 339

---

**REFLevel<m>:ABSolute:BDIStance** <Value>

The distance between the lower reference level and the low signal value.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Example:** `REFLevel2:ABSolute:BDIStance 0.0002`
Sets the bottom distance for Ch1Wfm1 to 0.2 mV. C1W1 corresponds to suffix number 2.

**Example:** See: "Manual reference level definition using absolute values" on page 1176

**Usage:** Asynchronous command

**Manual operation:** See "Bottom distance" on page 339

---

**REFLevel<m>:ABSolute:MLEVel** <Value>

For user signal level selection, the level is the middle level between high and low signal level. The value is adjusted automatically if you change the high or low signal levels. Vice versa, if you change the middle level, the high and low signal levels are adjusted.

For user reference level selection, the level is the middle level between upper and lower reference level. The value is adjusted automatically if you change the upper or lower reference levels. Vice versa, if you change the middle level, the upper and lower reference levels are adjusted.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Example:** `REFLevel2:ABSolute:MLEVel 0.05`
Sets the middle signal level for Ch1Wfm1 to 50 mV. C1W1 corresponds to suffix number 2.

**Example:** See: "Manual reference level definition using absolute values" on page 1176

**Usage:** Asynchronous command

**Manual operation:** See "Middle" on page 338

---

**REFLevel<m>:ABSolute:ULEVel** <Value>

The upper reference level, required e.g. to determine a rise.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Example:** `REFLevel2:ABSolute:ULEVel 0.01`
Sets the upper reference level for Ch1Wfm1 to 10 mV. C1W1 corresponds to suffix number 2.

**Example:** See: "Manual reference level definition using absolute values" on page 1176

**Usage:** Asynchronous command

**Manual operation:** See "Upper level" on page 339

---

**REFLevel<m>:ABSolute:LLEVel** <Value>

The lower reference level, required e.g. to determine a fall.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Example:** `REFLevel2:ABSolute:LLEVel 0.001`
Sets the lower reference level for Ch1Wfm1 to 1 mV. C1W1 corresponds to suffix number 2.

**Example:** See: "Manual reference level definition using absolute values" on page 1176

**Usage:** Asynchronous command

**Manual operation:** See "Lower level" on page 339

---

**REFLevel<m>:RELative:UPPer** <Value>

Sets the upper relative reference level if `REFLevel<m>:RELative:MODE` is set to `USER`.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value> Percentage of the high signal level.

**Example:** `REFLevel8:RELative:LOWer 85`
Sets the upper reference level for Ch3Wfm1 to 85 %. Ch3Wfm1 corresponds to suffix number 8.

**Example:** See: "Manual reference level definition using relative values" on page 1175

**Usage:** Asynchronous command

**Manual operation:** See "Upper ref level, Middle ref level, Lower ref level" on page 338

---

**REFLevel<m>:RELative:MIDDle** <Value>

Sets the middle relative reference level if `REFLevel<m>:RELative:MODE` is set to `USER`.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value> Percentage of the high signal level.

**Example:** `REFLevel8:RELative:MIDDle 50`
Sets the middle reference level for Ch3Wfm1 to 50 %. Ch3Wfm1 corresponds to suffix number 8.

**Example:** See: "Manual reference level definition using relative values" on page 1175

**Usage:** Asynchronous command

**Manual operation:** See "Upper ref level, Middle ref level, Lower ref level" on page 338

---

**REFLevel<m>:RELative:LOWer** <Value>

Sets the lower relative reference level if `REFLevel<m>:RELative:MODE` is set to `USER`.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value> Percentage of the high signal level.

**Example:** `REFLevel8:RELative:LOWer 15`
Sets the lower reference level for Ch3Wfm1 to 15 %. Ch3Wfm1 corresponds to suffix number 8.

**Example:** See: "Manual reference level definition using relative values" on page 1175

**Usage:** Asynchronous command

**Manual operation:** See "Upper ref level, Middle ref level, Lower ref level" on page 338

### 24.12.15.4 Hysteresis

**REFLevel<m>:RELative:HYSTeresis** <Value>

Defines a hysteresis for the middle reference level. A rise or fall from the middle reference value that does not exceed the hysteresis is rejected as noise.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis" on page 340

### 24.12.15.5 Tube

REFLevel<m>:RELative:OTUBe....1519
REFLevel<m>:RELative:ITUBe....1520
REFLevel<m>:ABSolute:TOTube....1520
REFLevel<m>:ABSolute:TITube....1520
REFLevel<m>:ABSolute:BITube....1521
REFLevel<m>:ABSolute:BOTube....1521

**REFLevel<m>:RELative:OTUBe** <Value>

Defines a percentage of the signal level by which the absolute signal level may be larger than the high signal level or lower than the low signal level to be considered high or low, respectively.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Usage:** Asynchronous command

**Manual operation:** See "Relative outer" on page 341

**REFLevel<m>:RELative:ITUBe** <Value>

Defines a percentage of the signal level by which the absolute signal level may be higher than the low signal level or lower than the high signal level to be considered low or high, respectively.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Usage:** Asynchronous command

**Manual operation:** See "Relative inner" on page 341

**REFLevel<m>:ABSolute:TOTube** <Value>

Defines an area above the high signal level which is still considered to be high level.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Usage:** Asynchronous command

**Manual operation:** See "Top outer" on page 340

**REFLevel<m>:ABSolute:TITube** <Value>

Defines an area beneath the high signal level which is still considered to be high level.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Usage:** Asynchronous command

**Manual operation:** See "Top inner" on page 340

**REFLevel<m>:ABSolute:BITube** <Value>

Defines an area above the low signal level which is still considered to be low level.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Usage:** Asynchronous command

**Manual operation:** See "Bottom inner" on page 340

**REFLevel<m>:ABSolute:BOTube** <Value>

Defines an area beneath the low signal level which is still considered to be low level.

**Suffix:**
<m> Valid suffix numbers: 2 to 21, 61 to 68, 108 to 118, 151 to 157, 173 to 179
Source waveform of the measurement, see Chapter 24.4.1, "Waveform suffix", on page 1188.

**Parameters:**
<Value>

**Usage:** Asynchronous command

**Manual operation:** See "Bottom outer" on page 340

### 24.12.15.6 Results

MEASurement<m>:REFLevel:RESult:LOWer? ..... 1521
MEASurement<m>:REFLevel:RESult:MIDDle? ..... 1521
MEASurement<m>:REFLevel:RESult:UPPer? ..... 1521
MEASurement<m>:REFLevel:RESult:SIGLow? ..... 1522
MEASurement<m>:REFLevel:RESult:SIGHigh? ..... 1522
MEASurement<m>:REFLevel:RESult:BINNer? ..... 1522
MEASurement<m>:REFLevel:RESult:BOUTer? ..... 1522
MEASurement<m>:REFLevel:RESult:TINNer? ..... 1523
MEASurement<m>:REFLevel:RESult:TOUTer? ..... 1523

**MEASurement<m>:REFLevel:RESult:LOWer?**
**MEASurement<m>:REFLevel:RESult:MIDDle?**
**MEASurement<m>:REFLevel:RESult:UPPer?**

Return the lower, middle, and upper reference level, respectively.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Return values:**
<Upper> Range: -100E+24 to 100E+24
*RST: 0

**Usage:** Query only
Asynchronous command

---

**MEASurement<m>:REFLevel:RESult:SIGLow?**
**MEASurement<m>:REFLevel:RESult:SIGHigh?**

Return the signal value that represents a low or high level, respectively.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Return values:**
<SignalHigh> Range: -100E+24 to 100E+24
*RST: 0

**Usage:** Query only
Asynchronous command

---

**MEASurement<m>:REFLevel:RESult:BINNer?**

Returns the area above the low signal level which is still considered to be low level.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Return values:**
<BottomInner> Range: -100E+24 to 100E+24
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Bottom inner" on page 340

---

**MEASurement<m>:REFLevel:RESult:BOUTer?**

Returns the area beneath the low signal level which is still considered to be low level.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Return values:**
<BottomOuter> Range: -100E+24 to 100E+24
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Bottom outer" on page 340

---

**MEASurement<m>:REFLevel:RESult:TINNer?**

Returns the area beneath the high signal level which is still considered to be high level.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Return values:**
<TopInner> Range: -100E+24 to 100E+24
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Top inner" on page 340

---

**MEASurement<m>:REFLevel:RESult:TOUTer?**

Returns the area above the high signal level which is still considered to be high level.

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Return values:**
<TopOuter> Range: -100E+24 to 100E+24
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Top outer" on page 340

## 24.13 Spectrum analysis

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

### 24.13.1 Basic FFT

CALCulate:MATH<m>:FFT:LOGScale ..... 1525
CALCulate:MATH<m>:FFT:STARt ..... 1525
CALCulate:MATH<m>:FFT:STOP ..... 1525
CALCulate:MATH<m>:FFT:CFRequency ..... 1526
CALCulate:MATH<m>:FFT:FULLspan ..... 1526
CALCulate:MATH<m>:FFT:SPAN ..... 1526
CALCulate:MATH<m>:FFT:TYPE ..... 1527
CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:ADJusted? ..... 1527
CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:AUTO ..... 1527
CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:RATio ..... 1528
CALCulate:MATH<m>:FFT:BANDwidth[:RESolution][:VALue] ..... 1528
CALCulate:MATH<m>:FFT:WINDow:TYPE ..... 1529
CALCulate:MATH<m>:FFT:FRAMe:ARIThmetics ..... 1530
CALCulate:MATH<m>:FFT:FRAMe:COVerage? ..... 1530
CALCulate:MATH<m>:FFT:FRAMe:MAXCount ..... 1530
CALCulate:MATH<m>:FFT:FRAMe:OFACtor ..... 1531
CALCulate:MATH<m>:FFT:GATE:COUPling ..... 1531
TIMebase:RACTime? ..... 1532
CALCulate:MATH<m>:FFT:GATE:ABSolute:STARt ..... 1532
CALCulate:MATH<m>:FFT:GATE:ABSolute:STOP ..... 1532
CALCulate:MATH<m>:FFT:GATE:MODE ..... 1533
CALCulate:MATH<m>:FFT:GATE:RELative:STARt ..... 1533
CALCulate:MATH<m>:FFT:GATE:RELative:STOP ..... 1533
CALCulate:MATH<m>:FFT:GATE:ZCOupling ..... 1534
CALCulate:MATH<m>:FFT:GATE[:STATe] ..... 1534
CALCulate:MATH<m>:FFT:MAGNitude:LEVel ..... 1534
CALCulate:MATH<m>:FFT:MAGNitude:RANGe ..... 1535
CALCulate:MATH<m>:FFT:MAGNitude:SCALe ..... 1535
CALCulate:MATH<m>:FFT:PHASe:SCALe ..... 1536
CALCulate:MATH<m>:FFT:PHASe:SUPPression ..... 1536

CALCulate:MATH<m>:FFT:PHASe:THReshold ..... 1537
CALCulate:MATH<m>:FFT:PHASe:UNWRap ..... 1537
CALCulate:MATH<m>:FFT:COUPled:WITH<1..8> ..... 1537

**CALCulate:MATH<m>:FFT:LOGScale** <XAxisMode>

Defines the scaling method for the frequency (x-)axis of the spectrogram.

**Suffix:**
<m> 1..8

**Parameters:**
<XAxisMode> LIN | LOG

**LOG**
Logarithmic scaling

**LIN**
Linear scaling

*RST: LIN

**Usage:** Asynchronous command

**Manual operation:** See "Frequency axis" on page 408

**CALCulate:MATH<m>:FFT:STARt** <StartFreq>

Defines the start frequency of the displayed frequency span.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<StartFreq> start frequency

Range: 0 to 5E+9
Increment: 1
*RST: 2E+9
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Start frequency" on page 409

**CALCulate:MATH<m>:FFT:STOP** <StopFreq>

Defines the stop frequency of the displayed frequency span.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**

<StopFreq> stop frequency

Range: 0 to 5E+9
Increment: 1
*RST: 2E+9
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Stop frequency" on page 409

---

**CALCulate:MATH<m>:FFT:CFRequency** <CenterFreq>

Defines the position of the displayed frequency range, which is (Center - Span/2) to (Center + Span/2). The width of the range is defined using the `CALCulate:MATH<m>:FFT:SPAN` command.

**Suffix:**

<m> 1..8
math waveform

**Parameters:**

<CenterFreq> center frequency

Range: 0 to 2E+12
Increment: 1
*RST: 2.5E+9
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Center frequency" on page 409

---

**CALCulate:MATH<m>:FFT:FULLspan**

Performs FFT calculation for the full frequency span.

**Suffix:**

<m> 1..8
math waveform

**Usage:** Event
Asynchronous command

**Manual operation:** See "Full span" on page 409

---

**CALCulate:MATH<m>:FFT:SPAN** <FreqSpan>

The span is specified in Hertz and defines the width of the displayed frequency range, which is (Center - Span/2) to (Center + Span/2). The position of the span is defined using the `CALCulate:MATH<m>:FFT:CFRequency` command.

**Suffix:**
<m> 1..8
Math waveform

**Parameters:**
<FreqSpan> Frequency span
Range: 1 to 20E+9
Increment: 1
*RST: Bandwidth of the instrument model, bandwidth extension options are not considered
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Frequency span" on page 409

---

**CALCulate:MATH<m>:FFT:TYPE** <Type>

Selects between the magnitude and phase spectrum type.

**Suffix:**
<m> 1..8

**Parameters:**
<Type> MAGNitude | PHASe
*RST: MAGNitude

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 408

---

**CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:ADJusted?**

Queries the effective resolution bandwidth.

**Suffix:**
<m> 1..8
Math waveform

**Return values:**
<AdjResBW> effective resolution bandwidth
Range: 0.01 to 20E+12
*RST: 0
Default unit: Hz

**Usage:** Query only

**Manual operation:** See "Resolution BW" on page 410

---

**CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:AUTO** <State>

Couples the frequency span to the RBW.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<State> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Span/RBW Coupling" on page 409

---

**CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:RATio** <SpanRBWRatio>

This command defines the ratio of span (Hz) / resolution bandwidth (Hz).

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<SpanRBWRatio> ratio span / resolution bandwidth
Range: 1 to 1000
Increment: 1
*RST: 100

**Usage:** Asynchronous command

**Manual operation:** See "Span/RBW Ratio" on page 410

---

**CALCulate:MATH<m>:FFT:BANDwidth[:RESolution][:VALue]** <ResolutionBW>

This command defines the resolution bandwidth.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<ResolutionBW> resolution bandwidth
Range: 0.01 to 160E+6
Increment: 0.01
*RST: 2E+6
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Resolution BW" on page 410

**CALCulate:MATH<m>:FFT:WINDow:TYPE** <WindowType>

Windowing helps minimize the discontinuities at the end of the measured signal interval and thus reduces the effect of spectral leakage, increasing the frequency resolution.

Various different window functions are provided in the R&S RTP to suit different input signals. Each of the window functions has specific characteristics, including some advantages and some trade-offs. These characteristics need to be considered carefully to find the optimum solution for the measurement task.

**Suffix:**

<m> 1..8
math waveform

**Parameters:**

<WindowType> RECTangular | HAMMing | HANN | BLACkharris | GAUSsian | FLATTOP2 | FLATtop2 | KAISerbessel

**RECTangular**
The rectangular window has the best frequency resolution, but a poor amplitude accuracy and is recommended for separating two tones with almost equal amplitudes and a small frequency distance.

**HAMMing**
The Hamming window is bell shaped and has a good frequency resolution and fair amplitude accuracy. It is recommended for frequency response measurements as well as sine waves, periodic signals and narrow-band noise

**HANN**
The Hann window is bell shaped and has a slightly worse frequency resolution but smaller sidelobe level than the Hamming window. The applications are the same.

**BLACkharris**
The Blackman window is bell shaped and has a poor frequency resolution, but very good amplitude accuracy. It is recommended mainly for signals with single frequencies to detect harmonics.

**GAUSsian**
Good frequiency resolution and best magnitude resolution, recommended for weak signals and short duration

**FLATTOP2 = FLATtop2**
The flattop window has a poor frequency resolution, but the best amplitude accuracy and the sharpest side lobe. It is recommended for accurate single-tone amplitude measurements.

**KAISerbessel**
The Kaiser-Bessel window has a fair frequency resolution and good amplitude accuracy, and a very high sidelobe level. It is recommended for separating two tones with differing amplitudes and a small frequency distance.

*RST: BLACkharris

**Usage:** Asynchronous command

**Firmware/software:** Version 3.35 and higher: Use FLATTOP2 or FLATtop2 instead of FLAT2

**Manual operation:** See "Window type" on page 410

---

**CALCulate:MATH<m>:FFT:FRAMe:ARIThmetics** <Arithmetics>

The arithmetic mode defines how the final FFT result is calculated from the individual frame results.

**Suffix:**
<m> 1..8
Selects the math waveform.

**Parameters:**
<Arithmetics> OFF | ENVelope | AVERage | RMS | MINHold | MAXHold
See "Segment arithmetic" on page 412
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Segment arithmetic" on page 412

---

**CALCulate:MATH<m>:FFT:FRAMe:COVerage?**

Due to the restriction of the number of frames (see `CALCulate:MATH<m>:FFT:FRAMe:MAXCount` on page 1530), the waveform may only be analyzed partially. This command queries the percentage of the trace that was analyzed, i.e. which part of the trace was included in the frame calculation.

**Suffix:**
<m> 1..8
math waveform

**Return values:**
<FrameCoverage> Range: 0 to 100
*RST: 100
Default unit: %

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Max FFTs per acquisition" on page 411

---

**CALCulate:MATH<m>:FFT:FRAMe:MAXCount** <MaxFrameCount>

Restricts the maximum number of frames to be calculated. Due to the other parameter settings, the required number of frames may become very high, thus slowing performance. By restricting the number of frames, you can avoid performance loss without changing the other parameters.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<MaxFrameCount> Range: 1 to 10000
Increment: 10
*RST: 1000

**Usage:** Asynchronous command

**Manual operation:** See "Max FFTs per acquisition" on page 411

---

**CALCulate:MATH<m>:FFT:FRAMe:OFACtor** <OverlapFactor>

Defines the minimum factor by which two neighboring frames overlap. If the required number of frames to cover the input values allows for more overlap, the factor is increased.

The higher the overlap factor, the more frames are used. This leads to more individual results and improves detection of transient signal effects. However, it also extends the duration of the calculation.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<OverlapFactor> Range: 0 to 90
Increment: 1
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Overlap Factor" on page 412

---

**CALCulate:MATH<m>:FFT:GATE:COUPling** <GateRBWCoupling>

Defines the behaviour of the record length or RBW value in dependancy to the other FFT parameters.

See also:

- "Record Length/RBW Coupling" on page 415
- Chapter 9.1.1, "Fundamentals of FFT analysis", on page 399

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<GateRBWCoupling> LENGth | RBW

**LENGth**
The record length remains constant. If not enough samples are available for the selected RBW, the RBW will be decreased.

**RBW**
The RBW is not adapted, i.e. remains as defined by the user. The required acquisition time for this RBW is indicated. If necessary and possible, the record length is extended to acquire the required number of samples.

*RST: RBW

**Usage:** Asynchronous command

**Manual operation:** See "Record Length/RBW Coupling" on page 415

---

**TIMebase:RACTime?**

Queries the required acquisition time. If FFT gating is used and the resolution BW is set to constant, record length can be extended to acquire the required number of samples. In this case, the required acquisition time differs from the adjusted acquisition time (`TIMebase:RANGe`).

**Return values:**
<RqrdAcqTime> Required acquisition time for FFT

Range: 125E-12 to 100E+3
*RST: 0.5
Default unit: s

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Required acquisition time" on page 415

---

**CALCulate:MATH<m>:FFT:GATE:ABSolute:STARt** <Start>

Defines the starting value for the gate.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<Start>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "(Relative) Start" on page 330

---

**CALCulate:MATH<m>:FFT:GATE:ABSolute:STOP** <Stop>

Defines the end value for the gate.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<Stop> Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Usage:** Asynchronous command

**Manual operation:** See "(Relative) Stop" on page 331

---

**CALCulate:MATH<m>:FFT:GATE:MODE** <Mode>

Defines whether the gate settings are configured using absolute or relative values.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<Mode> ABS | REL
*RST: ABS

**Usage:** Asynchronous command

**Manual operation:** See "Gate description" on page 330

---

**CALCulate:MATH<m>:FFT:GATE:RELative:STARt** <RelativeStart>

Defines the starting value for the gate in percent.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<RelativeStart> Range: 0 to 100
Increment: 0.1
*RST: 0
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "(Relative) Start" on page 330

---

**CALCulate:MATH<m>:FFT:GATE:RELative:STOP** <RelativeStop>

Defines the end value for the gate in percent.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**

<RelativeStop>
Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "(Relative) Stop" on page 331

---

**CALCulate:MATH<m>:FFT:GATE:ZCOupling** <ZoomCoupling>

If enabled, the gate area is defined identically to the zoom area for the zoom diagram.

**Suffix:**

<m>
1..8
math waveform

**Parameters:**

<ZoomCoupling>
ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Zoom" on page 331

---

**CALCulate:MATH<m>:FFT:GATE[:STATe]** <State>

Enables FFT gating.

**Suffix:**

<m>
1..8
math waveform

**Parameters:**

<State>
ON | OFF
*RST: OFF

**Usage:** Asynchronous command

---

**CALCulate:MATH<m>:FFT:MAGNitude:LEVel** <VerticalMax>

Defines the reference level for dB scaling.

**Suffix:**

<m>
1..8
Math waveform

**Parameters:**

<VerticalMax>
Range: -1E+15 to 1E+15
Increment: 0.01
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Reference level" on page 418

---

**CALCulate:MATH<m>:FFT:MAGNitude:RANGe** <Range>

Defines the vertical value range in spectrum mode.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<Range> Range: 1 to 500
Increment: 1
*RST: 100
Default unit: dB

**Usage:** Asynchronous command

---

**CALCulate:MATH<m>:FFT:MAGNitude:SCALe** <MagnitudeScale>

Defines the scaling of the y-axis for magnitude spectrums. The display values are valid for 50Ω termination impendance.

Select the unit according to the vertical unit of the source waveform.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<MagnitudeScale> LINear | DBM | DB | DBUV | DBMV | DBV | DBPS | DBNS | DBUS | DBMS | DBS | DBHZ | DBKHZ | DBKHz | DBMHZ | DBMHz | DBGHZ | DBGHz | DBA | DBMA | DBUA

**LINear**
Linear scaling; displays the RMS value.

**DBM | DB | DBUV | DBMV | DBV**
Basic FFT

**DBPS | DBNS | DBUS | DBMS | DBS | DBHZ | DBKHZ | DBKHz | DBMHZ | DBMHz | DBGHZ | DBGHz**
FFT on tracks of time or frequency measurements

**DBA | DBMA | DBUA**
FFT on current waveforms

*RST: DBM

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 418

*Table 24-12: Logarithmic scaling values*

| | |
|---|---|
| DBM | dBm |
| DB | dB (related to reference level)<br>Set the reference level with `CALCulate:MATH<m>:FFT:MAGNitude:LEVel`. |
| DBUV | dBµV |
| DBMV | dBmV |
| DBV | dBV |
| DBPS | dBps |
| DBNS | dBns |
| DBUS | dBµs |
| DBMS | dBms |
| DBS | dBs |
| DBHZ | dBHz |
| DBKHZ = DBKHz | dBkHz |
| DBMHZ = DBMHz | dBMHz |
| DBGHZ = DBGHz | dBGHz |
| DBA | dBA |
| DBMA | dBmA |
| DBUA | dBµA |

**CALCulate:MATH<m>:FFT:PHASe:SCALe** <PhaseScale>

Defines the scaling unit for phase display.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<PhaseScale> DEGRees | RADians
*RST: DEGRees

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 418

**CALCulate:MATH<m>:FFT:PHASe:SUPPression** <Suppression>

Enables noise suppression. Phase calculation is restricted to frequencies with a minimum magnitude, the threshold value (see `CALCulate:MATH<m>:FFT:PHASe:THReshold` on page 1537).

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<Suppression> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Suppression" on page 418

---

**CALCulate:MATH<m>:FFT:PHASe:THReshold** <SupprThres>

Defines the minimum frequency magnitude for which phases are calculated. This setting is only available if `CALCulate:MATH<m>:FFT:PHASe:SUPPression` is set to "ON".

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<SupprThres> Range: -180 to 180
Increment: 0.1
*RST: 0
Default unit: dBm

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 418

---

**CALCulate:MATH<m>:FFT:PHASe:UNWRap** <Unwrap>

If enabled, phase shifts due to a limitation of the value range are eliminated.

**Suffix:**
<m> 1..8
math waveform

**Parameters:**
<Unwrap> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Unwrap" on page 418

---

**CALCulate:MATH<m>:FFT:COUPled:WITH<1..8>** <MathIndex>

Copies the current FFT settings of the selected math waveform (m) to the other selected math waveform (m2), and couples the two waveforms. This can be repeated for all math waveforms.

If any FFT setting for any of the coupled spectrums is changed, it is changed for all coupled spectrums.

**Suffix:**
<m> 1..8
FFT waveforms to be coupled. <m>, <m2> must be active and distinct.

**Parameters:**
<MathIndex> ON | OFF
*RST: OFF

**Example:** Prerequisite: FFt waveforms M1 and M2 are active.
`CALC:MATH1:FFT:COUP:WITH2 ON`
Couples the math waveforms M1 and M2.

**Usage:** Asynchronous command

**Manual operation:** See "Coupled to M1...M8" on page 416

### 24.13.2 Waveform data

CALCulate:MATH<m>:DATA:STYPe? 1538
CALCulate:MATH<m>:DATA:HEADer? 1538
CALCulate:MATH<m>:DATA[:VALues]? 1539

**CALCulate:MATH<m>:DATA:STYPe?**

Returns the signal type of the source of the math waveform.

**Suffix:**
<m> 1..8
Selects the math waveform.

**Return values:**
<SignalType> SOUR | SPEC | CORR | MEAS | NONE
SOURce = normal signal
SPECtrum = FFT spectrum, specific math signal
CORRelation = correlated signal, specific math signal
MEAsurement = result of a measurement
NONE = undefined

**Usage:** Query only
Asynchronous command

**CALCulate:MATH<m>:DATA:HEADer?**

Returns the header of math waveform data. The header contains attributes of the waveform.

*Table 24-13: Header data*

| Position | Meaning | Example |
|---|---|---|
| 1 | XStart in s | -9.477E-008 = - 94,77 ns |
| 2 | XStop in s | 9.477E-008 = 94,77 ns |
| 3 | Record length of the waveform in Samples | 200000 |
| 4 | Number of values per sample interval. For most waveforms the result is 1, for peak detect and envelope waveforms it is 2. If the number is 2, the number of returned values is twice the number of samples (record length). | 1 |

**Suffix:**
<m> 1..8
Selects the math waveform.

**Example:** `CALC:MATH1:DATA:HEAD?`
`-9.477E-008,9.477E-008,200000,1`

**Usage:** Query only

---

**CALCulate:MATH<m>:DATA[:VALues]?**

Returns the data of the math waveform points for transmission from the instrument to the controlling computer. The data can be used in MATLAB, for example.

To set the export format, use `FORMat[:DATA]`.

You can retrieve only Y-values, or X- and Y-values. Use `EXPort:WAVeform:INCXvalues` to define this.

**Suffix:**
<m> 1..8
Selects the math waveform.

**Return values:**
<Data> List of values according to the format and content settings.

**Usage:** Query only

### 24.13.3 Spectrum analysis (option R&S RTP-K37)

In all `CALC:MATH<m>:FFT` commands, the suffix <m> selects the math waveform.

In all `CALC:MATH<m>:FFT:SPEC:TIM` commands, the suffix <m> selects the timeline.

CALCulate:MATH<m>:FFT:SPECtrogram:CMODe ..... 1540
CALCulate:MATH<m>:FFT:USEColtab ..... 1540
CALCulate:MATH<m>:FFT:SPECtrogram:STATe ..... 1540
CALCulate:MATH<m>:FFT:SPECtrogram:TIMeline<n>:POSition ..... 1541
CALCulate:MATH<m>:FFT:SPECtrogram:TIMeline<n>:STATe ..... 1541

**CALCulate:MATH<m>:FFT:SPECtrogram:CMODe** <ColorTableMode>

Selects the color table mode for the frequency analysis display.

**Suffix:**
<m> 1..8

**Parameters:**
<ColorTableMode> INCI | AMPL

**INCI**
("Incidence") The display color is set depending on the frequency of occurrence of a value.

**AMPL**
("Amplitude") In the spectrogram and the frequency analysis display, the color is used to indicate the magnitude of the FFT signal. The higher the amplitude the higher the color in the assigned color table.

*RST: INCI

**Usage:** Asynchronous command

**CALCulate:MATH<m>:FFT:USEColtab** <UseColorTable>

If enabled, the selected waveform is displayed according to its assigned color table. For information on the available color tables, see Chapter 4.3.2, "Color tables", on page 79.

If this option is disabled, the preset color of the selected channel source is displayed, and the intensity of the specific signal color varies according to the cumulative occurrence of the values.

**Suffix:**
<m> 1..8

**Parameters:**
<UseColorTable> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Use color table" on page 410

**CALCulate:MATH<m>:FFT:SPECtrogram:STATe** <State>

Enables the spectrogram display for a math waveform.

**Suffix:**
<m> 1..8

**Parameters:**
<State> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Display" on page 422

---

**CALCulate:MATH<m>:FFT:SPECtrogram:TIMeline<n>:POSition** <Position>

Defines the position of one of the two possible time lines in a spectrogram. The time line must be enabled first, using the `CALCulate:MATH<m>:FFT:SPECtrogram:TIMeline<n>:STATe` command.

**Suffix:**
<m> 1..8

<n> 1..2

**Parameters:**
<Position> The position of the time line is defined by the index of the data acquisition in the history. How many acquisitions are available depends on the history settings.

Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Time line 1/2" on page 422

---

**CALCulate:MATH<m>:FFT:SPECtrogram:TIMeline<n>:STATe** <State>

Enables one of two possible time lines in a spectrogram diagram. A time line marks a single waveform in the spectrogram, that is: the power vs frequency results for the data acquired at a specific time. After enabling a time line, the results for that time are displayed in the spectrum diagram.

The position of the time line is defined using the `CALCulate:MATH<m>:FFT:SPECtrogram:TIMeline<n>:POSition` command.

**Suffix:**
<m> 1..8

<n> 1..2

**Parameters:**
<State> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Time line 1/2" on page 422

## 24.14 Mask testing

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

- Mask test definition ..... 1542
- Mask definition: user mask ..... 1546
- Mask definition: waveform mask ..... 1553
- Event actions ..... 1555
- Results ..... 1558
- Eye mask testing ..... 1560

### 24.14.1 Mask test definition

MTESt:ADD ..... 1542
MTESt:REMove ..... 1543
MTESt[:STATe] ..... 1543
MTESt:SHOW ..... 1543
MTESt:RST ..... 1543
MTESt:SOURce ..... 1544
MTESt:CONDition ..... 1544
MTESt:TOLerance ..... 1545
MTESt:CTYPe ..... 1545
MTESt:FILE:NAME ..... 1545
MTESt:FILE:SAVE ..... 1546
MTESt:FILE:OPEN ..... 1546
MTESt:FILE:DELete ..... 1546

---

**MTESt:ADD** <MaskTestName>

Creates a new mask test definition with the specified name.

**Setting parameters:**

<MaskTestName> String with the name of the mask test

**Example:** See Chapter 24.3.3.1, "Creating a user mask", on page 1178

**Usage:** Setting only
Asynchronous command

**MTESt:REMove** <MaskTestName>

Deletes the mask test definition with the specified name.

**Setting parameters:**
<MaskTestName> String with the name of the mask test

**Usage:** Setting only
Asynchronous command

**MTESt[:STATe]** <MaskTestName>,<State>
**MTESt[:STATe]?** <MaskTestName>

Activates and deactivates the mask test. If the acquisition is running, the test starts immediately. Otherwise, the test starts when acquisition is started.

The testing is stopped when acquisition is stopped, also due to the `MTESt:ONViolation:STOP` command, or if `MASK[:STATe]` is set to "OFF".

The coommand needs `*OPC` command synchronisation.

**Parameters:**
<State> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Example:** `MTEST:STAT 'MyMask', ON; *OPC?`
See Chapter 24.3.3.1, "Creating a user mask", on page 1178

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 426

**MTESt:SHOW** <State>

Switches the display of all mask segments on or off.

**Parameters:**
<State> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Show mask" on page 436

**MTESt:RST**

Clears all totals and results in all "Mask Test" result boxes.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Mask results > Reset all" on page 437

**MTESt:SOURce** <MaskTestName>,<Source>
**MTESt:SOURce?** <MaskTestName>

Selects the waveform to be tested against the mask.

**Parameters:**

<Source> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | XY1 | XY2 | XY3 | XY4 | SG1TL1 | SG1TL2 | SG2TL1 | SG2TL2 | SG3TL1 | SG3TL2 | SG4TL1 | SG4TL2 | SG5TL1 | SG5TL2 | SG6TL1 | SG6TL2 | SG7TL1 | SG7TL2 | SG8TL1 | SG8TL2 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | EYE1 | EYE2 | EYE3 | EYE4 | SEYE1 | SEYE2

Waveform to be tested, see Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C1W1

**Parameters for setting and query:**

<MaskTestName> String containing the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Target" on page 426

**MTESt:CONDition** <MaskTestName>,<PassFailMode>
**MTESt:CONDition?** <MaskTestName>

Sets the first criteria for a failed test, the kind of hits to be considered for test evaluation. A test has failed if the number of sample hits or acquisition hits exceeds the limit defined by `MTESt:TOLerance`.

**Parameters:**

<PassFailMode> SAMPles | ACQuisitions

**SAMPles**
Considers the number of samples that hit the mask.

**ACQuisitions**
Considers the number of acquisitions that contain at least one sample hit. How many samples hit the mask in that acquisition is not relevant.

*RST: SAMPles

**Parameters for setting and query:**

<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Fail condition, Violation tolerance" on page 434

**MTESt:TOLerance** <MaskTestName>,<TolViolCnt>
**MTESt:TOLerance?** <MaskTestName>

Sets the second criteria for a failed test, the number of tolerable sample hits or acquisition hits. Use `MTESt:CONDition` to define which hits are considered for test evaluation.

**Parameters:**
<TolViolCnt> Range: 0 to 4000000000
Increment: 1
*RST: 0

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Fail condition, Violation tolerance" on page 434

---

**MTESt:CTYPe** <MaskTestName>,<DefinitionType>
**MTESt:CTYPe?** <MaskTestName>

Sets the method of mask definition.

**Parameters:**
<DefinitionType> USER | WFML | EYEMask | PROTocol

**USER**
The mask segments are created by entering the numerical x- and y-values of the mask points.
See: Chapter 24.14.2, "Mask definition: user mask", on page 1546

**WFML**
The mask is created from the envelope of an existing waveform.
See: Chapter 24.14.3, "Mask definition: waveform mask", on page 1553

**EYEMask**
Mask for eye diagram testing. See: Chapter 24.22.4, "Eye mask testing", on page 2412.
Requires jitter option R&S RTP-K12

*RST: USER

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Mask Type" on page 426

---

**MTESt:FILE:NAME** <MaskTestName>, <Path>
**MTESt:FILE:NAME?** <MaskTestName>

Specifies a file to save the mask test.

**Parameters:**
<Path> String containing path and file name, format .xml

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Manual operation:** See "Save / load mask" on page 426

---

**MTESt:FILE:SAVE** <MaskTestName>

Saves the specified mask test. It contains the mask definition, defined actions and fail conditions.

**Setting parameters:**
<MaskTestName> String with the name of the mask test

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save / load mask" on page 426

---

**MTESt:FILE:OPEN** <MaskTestName>

Loads the specified mask test to the instrument.

**Setting parameters:**
<MaskTestName> String with the name of the mask test

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save / load mask" on page 426

---

**MTESt:FILE:DELete** <MaskTestName>

Deletes the specified mask test.

**Setting parameters:**
<MaskTestName> String with the name of the mask test

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Save / load mask" on page 426

### 24.14.2 Mask definition: user mask

The chapter contains commands required for the definition of user masks - `MTESt:CTYPe` is set to `USER`.

**Segment and point indices**

In remote control, the numbering of segments and points starts from 0. But in manual operation, the numbering starts from 1.

MTESt:SEGMent:STATe ..... 1547
MTESt:SEGMent:ADD ..... 1547
MTESt:SEGMent:COUNt? ..... 1548
MTESt:SEGMent:INSert ..... 1548
MTESt:SEGMent:REMove ..... 1548
MTESt:SEGMent:CLEar ..... 1548
MTESt:SEGMent:REGion ..... 1549
MTESt:SEGMent:POINt:ADD ..... 1549
MTESt:SEGMent:POINt:INSert ..... 1549
MTESt:SEGMent:POINt:REMove ..... 1550
MTESt:SEGMent:POINt:COUNt? ..... 1550
MTESt:SEGMent:POINt:X ..... 1550
MTESt:SEGMent:POINt:Y ..... 1551
MTESt:SEGMent:RESCale:RECalculate ..... 1551
MTESt:SEGMent:RESCale:XFACtor ..... 1552
MTESt:SEGMent:RESCale:YFACtor ..... 1552
MTESt:SEGMent:RESCale:XOFFset ..... 1552
MTESt:SEGMent:RESCale:YOFFset ..... 1553

---

**MTESt:SEGMent:STATe** <MaskTestName>, <MaskSegmIdx>,<State>
**MTESt:SEGMent:STATe?** <MaskTestName>, <MaskSegmIdx>

Enables and disables the mask segment. Disabled segments are not considered by running tests.

**Parameters:**

<State> ON | OFF

*RST: ON

**Parameters for setting and query:**

<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

**Usage:** Asynchronous command

**Manual operation:** See "Mask segments" on page 428

---

**MTESt:SEGMent:ADD** <MaskTestName>

Creates a new segment in the mask definition.

**Setting parameters:**

<MaskTestName> String with the name of the mask test

**Example:** See Chapter 24.3.3.1, "Creating a user mask", on page 1178

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Mask segments" on page 428

---

**MTESt:SEGMent:COUNt?** <MaskTestName>

Returns the number of segments in the mask definition

**Query parameters:**
<MaskTestName> String with the name of the mask test

**Return values:**
<Count> Number of segments

**Usage:** Query only

**Manual operation:** See "Mask segments" on page 428

---

**MTESt:SEGMent:INSert** <MaskTestName>, <MaskSegmIdx>

Inserts a new segment before the specified index in the mask definition.

**Setting parameters:**
<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Mask segments" on page 428

---

**MTESt:SEGMent:REMove** <MaskTestName>, <MaskSegmIdx>

Removes the specified segment from the mask definition.

**Setting parameters:**
<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Mask segments" on page 428

---

**MTESt:SEGMent:CLEar** <MaskTestName>

Deletes all mask segments of the indicated mask test.

**Setting parameters:**
<MaskTestName> String with the name of the mask test

**Usage:** Setting only
Asynchronous command

---

**MTESt:SEGMent:REGion** <MaskTestName>, <MaskSegmIdx>,<Region>
**MTESt:SEGMent:REGion?** <MaskTestName>, <MaskSegmIdx>

Defines the region of the segment that builds the mask.

**Parameters:**
<Region> UPPer | LOWer | INNer

**UPPer**
the segment points are connected to a line, the display area above this line is the mask segment

**LOWer**
the segment points are connected to a line, the display area below this line is the mask segment

**INNer**
the segment points form a closed geometrical shape, which is the mask segment

*RST: INNer

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

**Example:** See Chapter 24.3.3.1, "Creating a user mask", on page 1178

**Usage:** Asynchronous command

**Manual operation:** See "Mask segments" on page 428

---

**MTESt:SEGMent:POINt:ADD** <MaskTestName>, <MaskSegmIdx>

Adds a new point to the segment definition.

**Setting parameters:**
<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

**Example:** See Chapter 24.3.3.1, "Creating a user mask", on page 1178

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Definition of segment" on page 429

---

**MTESt:SEGMent:POINt:INSert** <MaskTestName>, <MaskSegmIdx>,
<MaskSegmPtIdx>

Inserts a new point before the specified mask segment point.

**Setting parameters:**

<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

<MaskSegmPtIdx> Number of the point. Counting starts from 0.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Definition of segment" on page 429

---

**MTESt:SEGMent:POINt:REMove** <MaskTestName>, <MaskSegmIdx>, <MaskSegmPtIdx>

Removes the specified point from the mask segment.

**Setting parameters:**

<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

<MaskSegmPtIdx> Number of the point. Counting starts from 0.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Definition of segment" on page 429

---

**MTESt:SEGMent:POINt:COUNt?** <MaskTestName>, <MaskSegmIdx>

Returns the number of defined points for the specified mask segment.

**Query parameters:**

<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

**Usage:** Query only

**Manual operation:** See "Definition of segment" on page 429

---

**MTESt:SEGMent:POINt:X** <MaskTestName>, <MaskSegmIdx>, <MaskSegmPtIdx>,<X>
**MTESt:SEGMent:POINt:X?** <MaskTestName>, <MaskSegmIdx>, <MaskSegmPtIdx>

Defines the x-value of the mask segment point.

**Parameters:**

<X>
Range: -100E+24 to 100E+24
Increment: 1E-6
*RST: 0
Default unit: s

**Parameters for setting and query:**

<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

<MaskSegmPtIdx> Number of the point. Counting starts from 0.

**Example:** See Chapter 24.3.3.1, "Creating a user mask", on page 1178

**Usage:** Asynchronous command

**Manual operation:** See "Definition of segment" on page 429

---

**MTESt:SEGMent:POINt:Y** <MaskTestName>, <MaskSegmIdx>, <MaskSegmPtIdx>,<Y>
**MTESt:SEGMent:POINt:Y?** <MaskTestName>, <MaskSegmIdx>, <MaskSegmPtIdx>

Defines the y-value of the mask segment point.

**Parameters:**

<Y>
Range: -100E+24 to 100E+24
Increment: 1E-6
*RST: 0
Default unit: V

**Parameters for setting and query:**

<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

<MaskSegmPtIdx> Number of the point. Counting starts from 0.

**Example:** See Chapter 24.3.3.1, "Creating a user mask", on page 1178

**Usage:** Asynchronous command

**Manual operation:** See "Definition of segment" on page 429

---

**MTESt:SEGMent:RESCale:RECalculate** <MaskTestName>, <MaskSegmIdx>

Multiplies and adds the given x- and y-factors and offsets to the coordinates of all points of the selected mask segment.

**Setting parameters:**

<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Apply" on page 430

**MTESt:SEGMent:RESCale:XFACtor** <MaskTestName>,
<MaskSegmIdx>,<ExpansFactX>
**MTESt:SEGMent:RESCale:XFACtor?** <MaskTestName>, <MaskSegmIdx>
**MTESt:SEGMent:RESCale:YFACtor** <MaskTestName>,
<MaskSegmIdx>,<ExpansFactY>
**MTESt:SEGMent:RESCale:YFACtor?** <MaskTestName>, <MaskSegmIdx>

Stretches or compresses the selected mask segment in horizontal (XFACtor) or vertical direction (YFACtor). The x- or y-values of all points of the selected mask segment are multiplied with this factor. Factors >1 stretch the mask segment, while factors between 0 and 1 compress it. Negative values are possible and change the algebraic sign.

Only takes effect after the `MTESt:SEGMent:RESCale:RECalculate` command.

**Parameters:**

<ExpansFactY>
Range: -100 to 100
Increment: 1
*RST: 1

**Parameters for setting and query:**

<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

**Usage:** Asynchronous command

**Manual operation:** See "Factor Y" on page 430

**MTESt:SEGMent:RESCale:XOFFset** <MaskTestName>, <MaskSegmIdx>,<OffsetX>
**MTESt:SEGMent:RESCale:XOFFset?** <MaskTestName>, <MaskSegmIdx>

Moves the mask segment horizontally. The specified offset is added to the x-values of all points of the selected mask segment.

Only takes effect after the `MTESt:SEGMent:RESCale:RECalculate` command.

**Parameters:**

<OffsetX>
Range: -50 to 50
Increment: 1E-9
*RST: 0
Default unit: s

**Parameters for setting and query:**

<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

**Usage:** Asynchronous command

**Manual operation:** See "Offset X" on page 430

**MTESt:SEGMent:RESCale:YOFFset** <MaskTestName>, <MaskSegmIdx>,<OffsetY>
**MTESt:SEGMent:RESCale:YOFFset?** <MaskTestName>, <MaskSegmIdx>

Moves the mask segment vertically. The specified offset is added to the y-values of all points of the selected mask segment.

Only takes effect after the `MTESt:SEGMent:RESCale:RECalculate` command.

**Parameters:**
<OffsetY> Range: -1000 to 1000
Increment: 1E-6
*RST: 0
Default unit: V

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

<MaskSegmIdx> Number of the segment. Counting starts from 0.

**Usage:** Asynchronous command

**Manual operation:** See "Offset Y" on page 430

### 24.14.3 Mask definition: waveform mask

The chapter contains commands required for the definition of waveform masks - `MTESt:CTYPe` is set to `WFML`.

MTESt:REFWfm ..... 1553
MTESt:WFMLupdate ..... 1554
MTESt:WFMRescale:XWIDth ..... 1554
MTESt:WFMRescale:YWIDth ..... 1554
MTESt:WFMRescale:YPOSition ..... 1555
MTESt:WFMRescale:YSTRetch ..... 1555

**MTESt:REFWfm** <MaskTestName>,<Source>
**MTESt:REFWfm?** <MaskTestName>

Sets the reference waveform from which the mask is created.

The reference waveform can be created before, or loaded from a file with REFCurve commands, see Chapter 24.10.2, "Reference waveforms", on page 1388.

**Parameters:**
<Source> REF1 | REFerence1 | REF2 | REFerence2 | REF3 | REFerence3 | REF4 | REFerence4

REF1 = REFerence1, REF2 = REFerence2, REF3 = REFerence3, REF4 = REFerence4: reference waveforms

*RST: REF1

**Parameters for setting and query:**
<MaskTestName> String containing the name of the mask test

**Usage:** Asynchronous command

**MTESt:WFMLupdate** <MaskTestName>

Creates the upper and lower mask limit from the envelope of the selected reference waveform. If the reference waveform was not defined before, it is created automatically from the mask test source waveform which is set wtih `MTESt:SOURce`.

**Setting parameters:**
<MaskTestName> String containing the name of the mask test

**Usage:** Setting only
Asynchronous command

---

**MTESt:WFMRescale:XWIDth** <MaskTestName>,<HorizontalWidth>
**MTESt:WFMRescale:XWIDth?** <MaskTestName>

Sets the width of the mask in horizontal direction. The specified factor in divisions is added to the positive x-values and subtracted from the negative x-values of the mask limits in relation to the source waveform of the mask.

**Parameters:**
<HorizontalWidth>
Range: 0 to 1000
Increment: 0.01
*RST: 0
Default unit: div

**Parameters for setting and query:**
<MaskTestName> String containing the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Horizontal width" on page 432

---

**MTESt:WFMRescale:YWIDth** <MaskTestName>,<VerticalWidth>
**MTESt:WFMRescale:YWIDth?** <MaskTestName>

Sets the width of the waveform mask in vertical direction. The specified factor in divisions is added to the y-values of the upper mask limit and subtracted from the y-values of the lower mask limit. Thus, the upper half of the mask is pulled upwards, the lower half is pulled down.

**Parameters:**
<VerticalWidth> Vertical mask width in divisions
Range: 0 to 1000
Increment: 0.01
*RST: 0
Default unit: div

**Parameters for setting and query:**
<MaskTestName> String containing the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Vertical width" on page 432

**MTESt:WFMRescale:YPOSition** <MaskTestName>,<VertPosi>
**MTESt:WFMRescale:YPOSition?** <MaskTestName>

Moves the mask vertically within the display.

**Parameters:**
<VertPosi>
Range: -1000 to 1000
Increment: 0.01
*RST: 0
Default unit: div

**Parameters for setting and query:**
<MaskTestName> String containing the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Vertical position" on page 433

**MTESt:WFMRescale:YSTRetch** <MaskTestName>,<VerticalStretch>
**MTESt:WFMRescale:YSTRetch?** <MaskTestName>

Sets the vertical scaling to stretch the mask in y-direction. The scaling axis is the horizontal line through the lowest value of the lower mask limit.

**Parameters:**
<VerticalStretch> Scale factor in %
Range: 10 to 1000
Increment: 1
*RST: 100
Default unit: %

**Parameters for setting and query:**
<MaskTestName> String containing the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Vertical stretch" on page 433

### 24.14.4 Event actions

MTESt:ONViolation:BEEP....1555
MTESt:ONViolation:STOP....1556
MTESt:ONViolation:PRINt....1556
MTESt:ONViolation:SAVewaveform....1556
MTESt:ONViolation:REPort....1557
MTESt:ONViolation:TRIGgerout....1557
MTESt:ONViolation:RUNexec....1557

**MTESt:ONViolation:BEEP** <MaskTestName>,<Beep>
**MTESt:ONViolation:BEEP?** <MaskTestName>

Generates a beep sound for the specified event.

**Parameters:**
<Beep> NOACtion | SUCCess | VIOLation
See Chapter 24.4.5, "Event parameter", on page 1195
*RST: NOACtion

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Beep" on page 435

---

**MTESt:ONViolation:STOP** <MaskTestName>,<StopAcq>
**MTESt:ONViolation:STOP?** <MaskTestName>

Stops data acquisition for the specified event.

**Parameters:**
<StopAcq> NOACtion | SUCCess | VIOLation
See Chapter 24.4.5, "Event parameter", on page 1195
*RST: NOACtion

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Stop acq." on page 435

---

**MTESt:ONViolation:PRINt** <MaskTestName>,<Print>
**MTESt:ONViolation:PRINt?** <MaskTestName>

Saves a screenshot including the measurement results for the specified event.

**Parameters:**
<Print> NOACtion | SUCCess | VIOLation
See Event parameter
*RST: NOACtion

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Save screenshot" on page 435

---

**MTESt:ONViolation:SAVewaveform** <MaskTestName>,<SaveWfm>
**MTESt:ONViolation:SAVewaveform?** <MaskTestName>

Saves the waveform data.

**Parameters:**
<SaveWfm> NOACtion | SUCCess | VIOLation
See Chapter 24.4.5, "Event parameter", on page 1195
*RST: NOACtion

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Save waveform" on page 436

---

**MTESt:ONViolation:REPort** <MaskTestName>,<Report>
**MTESt:ONViolation:REPort?** <MaskTestName>

Creates and saves a report of the current settings and results.

**Parameters:**
<Report> NOACtion | SUCCess | VIOLation
See Chapter 24.4.5, "Event parameter", on page 1195
*RST: NOACtion

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Report" on page 436

---

**MTESt:ONViolation:TRIGgerout** <MaskTestName>,<TriggerOutPulse>
**MTESt:ONViolation:TRIGgerout?** <MaskTestName>

Creates a trigger out pulse on mask violation or successful completion of the test cycle.

**Parameters:**
<TriggerOutPulse> NOACtion | SUCCess | VIOLation
*RST: NOACtion

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Trigger out pulse" on page 436

---

**MTESt:ONViolation:RUNexec** <MaskTestName>,<RunExecutable>
**MTESt:ONViolation:RUNexec?** <MaskTestName>

Starts an external application if the command is set to `VIOLation` or `SUCCess`.

Use the following commands to set up the application:

- `EXECutable:NAME` on page 1635
- `EXECutable:PARameter` on page 1635

- EXECutable:WDIRectory on page 1636

**Parameters:**
<RunExecutable> NOACtion | SUCCess | VIOLation
See Chapter 24.4.5, "Event parameter", on page 1195
*RST: NOACtion

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Start Executable" on page 436

### 24.14.5 Results

MTESt:RESult:STATe? 1558
MTESt:RESult[:RESult]? 1558
MTESt:RESult:COUNt:WAVeforms? 1559
MTESt:RESult:COUNt:REMaining? 1559
MTESt:RESult:COUNt:FWAVeforms? 1559
MTESt:RESult:COUNt:FAILures? 1560
MTESt:RESult:FRATe? 1560

**MTESt:RESult:STATe?** <MaskTestName>

Shows if the test is running or has finished. The state is set to "Finished" if no acquisitions remain (see MTESt:RESult:COUNt:REMaining? on page 1559).

**Query parameters:**
<MaskTestName>

**Return values:**
<State> RUNNing | FINished
*RST: RUNNing

**Usage:** Query only
Asynchronous command

**Manual operation:** See "State" on page 424

**MTESt:RESult[:RESult]?** <MaskTestName>

Returns the test result.

A test has failed if the number of sample hits or acquisition hits exceeds the limit of "Violation tolerance" hits (see MTESt:TOLerance on page 1545, MTESt:RESult:COUNt:FAILures? on page 1560 and MTESt:RESult:COUNt:FWAVeforms? on page 1559).

**Query parameters:**
<MaskTestName>

**Return values:**
<TestResult> PASS | FAIL
*RST: PASS

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Test result" on page 425

---

**MTESt:RESult:COUNt:WAVeforms?** <MaskTestName>

Returns the number of tested acquisitions.

**Query parameters:**
<MaskTestName>

**Return values:**
<AcqsCompleted> Range: 0 to 100E+24
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Acq. completed" on page 424

---

**MTESt:RESult:COUNt:REMaining?** <MaskTestName>

Remaining acquisitions until "Average count / Nx Single count" is reached.

The value is useful if you test a specified number of acquisitions with action "Stop acquisition" on violation, or if the acquisition has been stopped manually.

**Query parameters:**
<MaskTestName>

**Return values:**
<AcqsRemaining> Range: 0 to 100E+24
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Acq. remaining" on page 424

---

**MTESt:RESult:COUNt:FWAVeforms?** <MaskTestName>

Returns the number of acquisitions that contained at least one sample hit.

**Query parameters:**
<MaskTestName>

**Return values:**
<AcquisitionHits> Range: 0 to 100E+24
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Acquisition hits" on page 424

---

**MTESt:RESult:COUNt:FAILures?** <MaskTestName>

Returns the number of sample hits that violated the mask.

**Query parameters:**
<MaskTestName>

**Return values:**
<SampleHits> Range: 0 to 100E+24
*RST: 0

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Sample hits" on page 424

---

**MTESt:RESult:FRATe?** <MaskTestName>

Ratio of acquisition hits to the number of tested acquisitions.

**Query parameters:**
<MaskTestName>

**Return values:**
<FailRate> Range: -100E+24 to 100E+24
*RST: 0
Default unit: %

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Fail rate" on page 424

## 24.14.6 Eye mask testing

### 24.14.6.1 Definition of eye masks

To perform mask testing on eye diagrams, the option R&S RTP-K12 adds a special mask definition type to the common mask definitions: the definition type "Eye".

MTESt:CEMask ..... 1561
MTESt:EYEMask:TYPE ..... 1561
MTESt:EYEMask:HEIGht<m>[:VALue] ..... 1561
MTESt:EYEMask:WIDTh<m>[:VALue] ..... 1562
MTESt:EYEMask:MSKRight ..... 1562
MTESt:EYEMask:MSKLeft ..... 1562

MTESt:EYEMask:HPERiod ..... 1562
MTESt:EYEMask:MSKTop ..... 1563
MTESt:EYEMask:MSKBottom ..... 1563
MTESt:EYEMask:TOFFset ..... 1563
MTESt:EYEMask:BOFFset ..... 1563
MTESt:EYEMask:TBSYmmetric ..... 1564
MTESt:EYEMask:TBWidth ..... 1564
MTESt:EYEMask:HPOSition ..... 1564
MTESt:EYEMask:VPOSition ..... 1565

**MTESt:CEMask** <MaskTestName>

Converts the test and mask definitions of the current mask test to a new mask test of type "User".

**Setting parameters:**
<MaskTestName> String with the name of the mask test

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Convert to user mask" on page 441

**MTESt:EYEMask:TYPE** <MaskTestName>,<Type>
**MTESt:EYEMask:TYPE?** <MaskTestName>

Defines the outline of the eye mask.

**Parameters:**
<Type> SQUare | DIAMond | HEXagon | OCTagon
*RST: DIAMond

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Shape" on page 439

**MTESt:EYEMask:HEIGht<m>[:VALue]** <MaskTestName>, <value>

Defines the main heigth af all eye mask shapes and the minor height for octagon mask shapes.

**Suffix:**
<m> 1..2
1 - main width
2 - minor width

**Parameters:**
<value> Default unit: s

**Setting parameters:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Main height (H1), Minor height (H2)" on page 440

---

**MTESt:EYEMask:WIDTh<m>[:VALue]** <MaskTestName>, <value>

Defines the main width af all eye mask shapes and the minor width for hexagon and octagon mask shapes.

**Suffix:**
<m> 1..2
1 - main width
2 - minor width

**Parameters:**
<value> Default unit: s

**Setting parameters:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Main width (W1), Minor width (W2)" on page 439

---

**MTESt:EYEMask:MSKRight** <MaskTestName>,<Copy>
**MTESt:EYEMask:MSKRight?** <MaskTestName>
**MTESt:EYEMask:MSKLeft** <MaskTestName>,<Copy>
**MTESt:EYEMask:MSKLeft?** <MaskTestName>

Copies the eye shape to the right and left, respectively.

The distance of the copy is defined using `MTESt:EYEMask:HPERiod`.

**Parameters:**
<Copy> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Segments" on page 440

---

**MTESt:EYEMask:HPERiod** <MaskTestName>,<InterpattLength>
**MTESt:EYEMask:HPERiod?** <MaskTestName>

Defines the time distance between the shape centers if `MTESt:EYEMask:MSKLeft` and/or `MTESt:EYEMask:MSKRight` are ON.

**Parameters:**

<InterpattLength> Range: 0 to 100
Increment: 0.01
*RST: 0.5
Default unit: s

**Parameters for setting and query:**

<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Periodicity" on page 440

---

**MTESt:EYEMask:MSKTop** <MaskTestName>,<Add>
**MTESt:EYEMask:MSKTop?** <MaskTestName>
**MTESt:EYEMask:MSKBottom** <MaskTestName>,<Add>
**MTESt:EYEMask:MSKBottom?** <MaskTestName>

Enable the upper (top) and lower (bottom) mask region, respectively.

**Parameters:**

<Add> ON | OFF
*RST: OFF

**Parameters for setting and query:**

<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Segments" on page 440

---

**MTESt:EYEMask:TOFFset** <MaskTestName>,<RectDistTop>
**MTESt:EYEMask:TOFFset?** <MaskTestName>
**MTESt:EYEMask:BOFFset** <MaskTestName>,<RectDistBtom>
**MTESt:EYEMask:BOFFset?** <MaskTestName>

Voltage distance from the eye shape center that limit the upper (TOFFset) and lower (BOFFset) regions.

**Parameters:**

<RectDistBtom> Range: 0 to 100
Increment: 0.01
*RST: 0.5
Default unit: V

**Parameters for setting and query:**

<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Offset top, Offset bottom" on page 440

**MTESt:EYEMask:TBSYmmetric** <MaskTestName>,<TopBtomSymetrie>
**MTESt:EYEMask:TBSYmmetric?** <MaskTestName>

Sets bottom and top offsets to the same value so that the outer regions are symmetric to the eye shape.

**Parameters:**
<TopBtomSymetrie> ON | OFF
*RST: ON

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Symmetric" on page 440

**MTESt:EYEMask:TBWidth** <MaskTestName>,<RectangleLength>
**MTESt:EYEMask:TBWidth?** <MaskTestName>

Sets the time width of the outer regions, symmetric to the eye shape center.

**Parameters:**
<RectangleLength> Range: 0 to 100
Increment: 0.01
*RST: 0.5
Default unit: s

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Width" on page 440

**MTESt:EYEMask:HPOSition** <MaskTestName>,<PatOffsetX>
**MTESt:EYEMask:HPOSition?** <MaskTestName>

Sets the horizontal (time) value of the eye shape enter and thus defines the horizontal position of the eye shape on the display.

**Parameters:**
<PatOffsetX> Range: -100 to 100
Increment: 0.01
*RST: 0
Default unit: s

**Parameters for setting and query:**
<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Horizontal center/Vertical center" on page 439

**MTESt:EYEMask:VPOSition** <MaskTestName>,<PatOffsetY>
**MTESt:EYEMask:VPOSition?** <MaskTestName>

Sets the vertical (voltage) value of the eye shape enter and thus defines the vertical position of the eye shape on the display.

**Parameters:**

<PatOffsetY>
Range: -100 to 100
Increment: 0.01
*RST: 0
Default unit: V

**Parameters for setting and query:**

<MaskTestName> String with the name of the mask test

**Usage:** Asynchronous command

**Manual operation:** See "Horizontal center/Vertical center" on page 439

## 24.15 Search

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

- General search settings ..... 1566
- Basic trigger search conditions ..... 1568
- Edge search conditions ..... 1570
- Glitch search conditions ..... 1570
- Interval search conditions ..... 1572
- Runt search conditions ..... 1573
- Slew rate search conditions ..... 1576
- Timeout search conditions ..... 1578
- Width search conditions ..... 1579
- Window search conditions ..... 1581
- Data2clock search conditions ..... 1584
- Pattern search conditions ..... 1586
- State search conditions ..... 1591
- Search on spectrum ..... 1594
- Search gate settings ..... 1595
- Search results ..... 1598

### 24.15.1 General search settings

SEARch:ADD ..... 1566
SEARch:CLEar ..... 1566
SEARch:REMove ..... 1566
SEARch:SOURce ..... 1566
SEARch:ONLine ..... 1567
SEARch:ALL ..... 1567

**SEARch:ADD** <SearchName>

Creates a new search definition with the specified name.

**Setting parameters:**
<SearchName> String with the name of the search

**Example:** See Chapter 24.3.4.1, "Searching for a pulse of specified width", on page 1179

**Usage:** Setting only
Asynchronous command

**SEARch:CLEar** <SearchName>

Clears the search results once to start a new search.

**Setting parameters:**
<SearchName> Search definition

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Clear results" on page 469

**SEARch:REMove** <Key>

Deletes the specified search definition.

**Setting parameters:**
<Key> String with the name of the search

**Usage:** Setting only
Asynchronous command

**SEARch:SOURce** <SearchName>,<Source>
**SEARch:SOURce?** <SearchName>

Defines the source on which the search conditions are applied. The source can be any analog or digital channel, math or reference waveform as well as a serial bus configured for a supported protocol.

**Parameters:**

<Source> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | SBUS1 | SBUS2 | SBUS3 | SBUS4

Source of the search, see Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C1W1

**Parameters for setting and query:**

<SearchName> String with the name of the search

**Example:** See Chapter 24.3.4.1, "Searching for a pulse of specified width", on page 1179

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 451

---

**SEARch:ONLine** <SearchName>,<OnlineState>
**SEARch:ONLine?** <SearchName>

If enabled, a search is performed repeatedly for each new data acquisition.

**Parameters:**

<OnlineState> ON | OFF

*RST: OFF

**Parameters for setting and query:**

<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 451

---

**SEARch:ALL** <SearchName>

Performs a search for all results on the existing data from the selected source.

**Setting parameters:**

<SearchName> Search definition

**Example:** See Chapter 24.3.4.1, "Searching for a pulse of specified width", on page 1179

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Enable" on page 451

### 24.15.2 Basic trigger search conditions

SEARch:TRIGger:EDGE[:STATe] ..... 1568
SEARch:TRIGger:GLITch[:STATe] ..... 1568
SEARch:TRIGger:INTerval[:STATe] ..... 1568
SEARch:TRIGger:PATTern[:STATe] ..... 1568
SEARch:TRIGger:RUNT[:STATe] ..... 1568
SEARch:TRIGger:SETHold[:STATe] ..... 1568
SEARch:TRIGger:SLEWrate[:STATe] ..... 1568
SEARch:TRIGger:STATe[:STATe] ..... 1568
SEARch:TRIGger:TIMeout[:STATe] ..... 1568
SEARch:TRIGger:WIDTh[:STATe] ..... 1569
SEARch:TRIGger:WINDow[:STATe] ..... 1569
SEARch:TRIGger:LEVel[:VALue] ..... 1569
SEARch:TRIGger:EDGE:ACOPy ..... 1569
SEARch:TRIGger:GLITch:ACOPy ..... 1569
SEARch:TRIGger:INTerval:ACOPy ..... 1569
SEARch:TRIGger:PATTern:ACOPy ..... 1569
SEARch:TRIGger:RUNT:ACOPy ..... 1569
SEARch:TRIGger:SETHold:ACOPy ..... 1569
SEARch:TRIGger:SLEWrate:ACOPy ..... 1569
SEARch:TRIGger:STATe:ACOPy ..... 1569
SEARch:TRIGger:TIMeout:ACOPy ..... 1569
SEARch:TRIGger:WIDTh:ACOPy ..... 1570
SEARch:TRIGger:WINDow:ACOPy ..... 1570
SEARch:TRIGger:EDGE:BCOPy ..... 1570

---

**SEARch:TRIGger:EDGE[:STATe]** <SearchName>,<State>
**SEARch:TRIGger:EDGE[:STATe]?** <SearchName>
**SEARch:TRIGger:GLITch[:STATe]** <SearchName>,<State>
**SEARch:TRIGger:GLITch[:STATe]?** <SearchName>
**SEARch:TRIGger:INTerval[:STATe]** <SearchName>,<State>
**SEARch:TRIGger:INTerval[:STATe]?** <SearchName>
**SEARch:TRIGger:PATTern[:STATe]** <SearchName>,<State>
**SEARch:TRIGger:PATTern[:STATe]?** <SearchName>
**SEARch:TRIGger:RUNT[:STATe]** <SearchName>,<State>
**SEARch:TRIGger:RUNT[:STATe]?** <SearchName>
**SEARch:TRIGger:SETHold[:STATe]** <SearchName>,<State>
**SEARch:TRIGger:SETHold[:STATe]?** <SearchName>
**SEARch:TRIGger:SLEWrate[:STATe]** <SearchName>,<State>
**SEARch:TRIGger:SLEWrate[:STATe]?** <SearchName>
**SEARch:TRIGger:STATe[:STATe]** <SearchName>,<State>
**SEARch:TRIGger:STATe[:STATe]?** <SearchName>
**SEARch:TRIGger:TIMeout[:STATe]** <SearchName>,<State>
**SEARch:TRIGger:TIMeout[:STATe]?** <SearchName>

**SEARch:TRIGger:WIDTh[:STATe]** <SearchName>,<State>
**SEARch:TRIGger:WIDTh[:STATe]?** <SearchName>
**SEARch:TRIGger:WINDow[:STATe]** <SearchName>,<State>
**SEARch:TRIGger:WINDow[:STATe]?** <SearchName>

Includes the search conditions for the selected trigger event type in the next search.

**Parameters:**
<State> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Edge, Glitch, Width, Runt, Window, Timeout, Interval, Slew rate, Setup & Hold, State, Pattern" on page 452

---

**SEARch:TRIGger:LEVel[:VALue]** <SearchName>, <SignalSource>, <Value>
**SEARch:TRIGger:LEVel[:VALue]?** <Key>, <SignalSource>

Sets the voltage of the trigger level that is used to determine other parameters.

**Parameters:**
<Value> Voltage value

**Parameters for setting and query:**
<SearchName> String with the name of the search

<SignalSource> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4

Source of the search, see Chapter 24.4.2, "Waveform parameter", on page 1193

**Usage:** Asynchronous command

**Manual operation:** See "Level" on page 453

---

**SEARch:TRIGger:EDGE:ACOPy** <SearchName>
**SEARch:TRIGger:GLITch:ACOPy** <SearchName>
**SEARch:TRIGger:INTerval:ACOPy** <SearchName>
**SEARch:TRIGger:PATTern:ACOPy** <SearchName>
**SEARch:TRIGger:RUNT:ACOPy** <SearchName>
**SEARch:TRIGger:SETHold:ACOPy** <SearchName>
**SEARch:TRIGger:SLEWrate:ACOPy** <SearchName>
**SEARch:TRIGger:STATe:ACOPy** <SearchName>
**SEARch:TRIGger:TIMeout:ACOPy** <SearchName>

**SEARch:TRIGger:WIDTh:ACOPy** <SearchName>
**SEARch:TRIGger:WINDow:ACOPy** <SearchName>

Copies the trigger event configuration from Trigger A for the selected channel source to the search condition settings.

See Chapter 6.3, "Trigger types", on page 196.

**Setting parameters:**
<SearchName> Search definition

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Copy from trigger A'" on page 452

---

**SEARch:TRIGger:EDGE:BCOPy** <SearchName>

Copies the trigger event configuration from trigger B for the selected channel source to the search condition settings.

**Setting parameters:**
<SearchName> String with name of the search

**Usage:** Setting only
Asynchronous command

## 24.15.3 Edge search conditions

Trigger level setting: SEARch:TRIGger:LEVel[:VALue] on page 1569

SEARch:TRIGger:EDGE:SLOPe..... 1570

---

**SEARch:TRIGger:EDGE:SLOPe** <SearchName>,<Slope>
**SEARch:TRIGger:EDGE:SLOPe?** <SearchName>

Selects the edge type.

**Parameters:**
<Slope> POSitive | NEGative | EITHer
See Chapter 24.4.3, "Slope parameter", on page 1194.
*RST: POSitive

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 453

## 24.15.4 Glitch search conditions

Trigger level setting: SEARch:TRIGger:LEVel[:VALue] on page 1569

SEARch:TRIGger:GLITch:POLarity....1571
SEARch:TRIGger:GLITch:RANGe....1571
SEARch:TRIGger:GLITch:WIDTh....1571

**SEARch:TRIGger:GLITch:POLarity** <SearchName>,<Polarity>
**SEARch:TRIGger:GLITch:POLarity?** <SearchName>

Indicates the polarity of a pulse, that is the direction of the first pulse slope.

**Parameters:**
<Polarity> POSitive | NEGative | EITHer
See Chapter 24.4.4, "Polarity parameter", on page 1194.
*RST: POSitive

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Width" on page 454

**SEARch:TRIGger:GLITch:RANGe** <SearchName>,<RangeMode>
**SEARch:TRIGger:GLITch:RANGe?** <SearchName>

Selects which glitches are identified: shorter or longer than the specified width (see `SEARch:TRIGger:GLITch:WIDTh` on page 1571).

**Parameters:**
<RangeMode> SHORter | LONGer
*RST: SHORter

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Width" on page 454

**SEARch:TRIGger:GLITch:WIDTh** <SearchName>,<Width>
**SEARch:TRIGger:GLITch:WIDTh?** <SearchName>

Sets the length of a glitch. The instrument triggers on pulses shorter or longer than this value (see also `SEARch:TRIGger:GLITch:RANGe` on page 1571).

You need to know the expected pulse widths of the circuit to set the glitch width correctly.

**Parameters:**
<Width> Range: 100E-12 to 1E-3
Increment: 100E-6
*RST: 1E-9
Default unit: s

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Width" on page 454

### 24.15.5 Interval search conditions

Trigger level setting: `SEARch:TRIGger:LEVel[:VALue]` on page 1569

SEARch:TRIGger:INTerval:SLOPe....1572
SEARch:TRIGger:INTerval:DELTa....1572
SEARch:TRIGger:INTerval:RANGe....1573
SEARch:TRIGger:INTerval:WIDTh....1573

---

**SEARch:TRIGger:INTerval:SLOPe** <SearchName>,<Slope>
**SEARch:TRIGger:INTerval:SLOPe?** <SearchName>

Sets the edge for the search.

**Parameters:**
<Slope> POSitive | NEGative | EITHer
See Chapter 24.4.3, "Slope parameter", on page 1194.
*RST: POSitive

**Parameters for setting and query:**
<SearchName> String parameter, name of the search definition

**Usage:** Asynchronous command

**Manual operation:** See "Slope, Range, Interv. width, ±Delta" on page 459

---

**SEARch:TRIGger:INTerval:DELTa** <SearchName>,<WidthDelta>
**SEARch:TRIGger:INTerval:DELTa?** <SearchName>

Defines a range around the "Interval width" value (see `SEARch:TRIGger:INTerval:WIDTh` on page 1573).

**Parameters:**
<WidthDelta> Range: 0 to 10
Increment: 100E-9
*RST: 0
Default unit: s

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Slope, Range, Interv. width, ±Delta" on page 459

**SEARch:TRIGger:INTerval:RANGe** <SearchName>,<RangeMode>
**SEARch:TRIGger:INTerval:RANGe?** <SearchName>

Selects how the range of an interval is defined based on the interval width and delta (see `SEARch:TRIGger:INTerval:WIDTh` on page 1573 and `SEARch:TRIGger:INTerval:DELTa` on page 1572).

**Parameters:**
<RangeMode> WITHin | OUTSide | SHORter | LONGer

**WITHin**
Triggers on pulse intervals inside a given range. The range is defined by "Interv. width" and "±Delta".

**OUTSide**
Triggers on intervals outside a given range. The range definition is the same as for "Within" range.

**SHORter**
Triggers on intervals shorter than the given "Interv. width".

**LONGer**
Triggers on intervals longer than the given "Interv. width".

*RST: OUTSide

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Slope, Range, Interv. width, ±Delta" on page 459

**SEARch:TRIGger:INTerval:WIDTh** <SearchName>,<Width>
**SEARch:TRIGger:INTerval:WIDTh?** <SearchName>

Defines the time between two pulses.

**Parameters:**
<Width>
Range: 100E-12 to 864
Increment: 100E-9
*RST: 5E-9
Default unit: s

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Slope, Range, Interv. width, ±Delta" on page 459

### 24.15.6 Runt search conditions

SEARch:TRIGger:RUNT:DELTa....1574
SEARch:TRIGger:RUNT:POLarity....1574
SEARch:TRIGger:RUNT:RANGe....1574

SEARch:TRIGger:RUNT:WIDTh ..... 1575
SEARch:TRIGger:LEVel:RUNT:LOWer ..... 1575
SEARch:TRIGger:LEVel:RUNT:UPPer ..... 1575

**SEARch:TRIGger:RUNT:DELTa** <SearchName>,<WidthDelta>
**SEARch:TRIGger:RUNT:DELTa?** <SearchName>

Defines a range around the given runt width.

**Parameters:**

<WidthDelta>
Range: 100E-12 to 864
Increment: 100E-9
*RST: 100E-12
Default unit: s

**Parameters for setting and query:**

<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Runt width, ±Delta" on page 456

**SEARch:TRIGger:RUNT:POLarity** <SearchName>,<Polarity>
**SEARch:TRIGger:RUNT:POLarity?** <SearchName>

Indicates the polarity of a pulse, that is the direction of the first pulse slope.

**Parameters:**

<Polarity> POSitive | NEGative | EITHer
See Chapter 24.4.4, "Polarity parameter", on page 1194.
*RST: POSitive

**Parameters for setting and query:**

<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Runt width, ±Delta" on page 456

**SEARch:TRIGger:RUNT:RANGe** <SearchName>,<Mode>
**SEARch:TRIGger:RUNT:RANGe?** <SearchName>

Selects how the time limit of the runt pulse is defined based on the runt width and delta (see `SEARch:TRIGger:RUNT:WIDTh` on page 1575 and `SEARch:TRIGger:RUNT:DELTa` on page 1574).

**Parameters:**

<Mode> ANY | LONGer | SHORter | WITHin | OUTSide

**ANY**
Triggers on all runts fulfilling the level condition, without time limitation.

**LONGer**
Triggers on runts longer than the given "Runt width".

**SHORter**
Triggers on runts shorter than the given "Runt width".

**WITHin**
Triggers if the runt length is inside a given time range. The range is defined by "Runt width" and "±Delta".

**OUTSide**
Triggers if the runt length is outside a given time range. The range definition is the same as for "Within" range.

*RST: ANY

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Runt width, ±Delta" on page 456

---

**SEARch:TRIGger:RUNT:WIDTh** <SearchName>,<Width>
**SEARch:TRIGger:RUNT:WIDTh?** <SearchName>

For the ranges "Shorter" and "Longer", the runt width defines the maximum and minimum pulse width, respectively.

For the ranges "Within" and "Outside", the runt width defines the center of a range which is defined by "±Delta".

The range is defined using `SEARch:TRIGger:RUNT:RANGe` on page 1574.

**Parameters:**
<Width>
Range: 100E-12 to 864
Increment: 100E-9
*RST: 5E-9
Default unit: s

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Runt width, ±Delta" on page 456

---

**SEARch:TRIGger:LEVel:RUNT:LOWer** <SearchName>, <SignalSource>, <Value>
**SEARch:TRIGger:LEVel:RUNT:LOWer?** <Key>, <SignalSource>
**SEARch:TRIGger:LEVel:RUNT:UPPer** <SearchName>, <SignalSource>, <Value>
**SEARch:TRIGger:LEVel:RUNT:UPPer?** <Key>, <SignalSource>

Set the lower and upper voltage threshold, respectively.

**Parameters:**
<Value> Voltage value

**Parameters for setting and query:**

<SearchName> String with the name of the search

<SignalSource> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4

Source of the search, see Chapter 24.4.2, "Waveform parameter", on page 1193

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Lower level" on page 456

### 24.15.7 Slew rate search conditions

SEARch:TRIGger:SLEWrate:DELTa....1576
SEARch:TRIGger:SLEWrate:RANGe....1577
SEARch:TRIGger:SLEWrate:SLOPe....1577
SEARch:TRIGger:SLEWrate:TIME....1577
SEARch:TRIGger:LEVel:TRANsition:LOWer....1578
SEARch:TRIGger:LEVel:TRANsition:UPPer....1578

---

**SEARch:TRIGger:SLEWrate:DELTa** <SearchName>,<TimeDelta>
**SEARch:TRIGger:SLEWrate:DELTa?** <SearchName>

Defines a time range around the given slew rate.

**Parameters:**

<TimeDelta>
Range: 0 to 10
Increment: 100E-9
*RST: 0
Default unit: s

**Parameters for setting and query:**

<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Slew rate, ±Delta" on page 460

**SEARch:TRIGger:SLEWrate:RANGe** <SearchName>,<RangeMode>
**SEARch:TRIGger:SLEWrate:RANGe?** <SearchName>

Selects how the time limit for the slew rate is defined. The time measurement starts when the signal crosses the first trigger level - the upper or lower level depending on the selected slope - and stops when the signal crosses the second level.

**Parameters:**

<RangeMode> INSRange | OUTRange | LTHan | GTHan

**INSRange**
Triggers on slew rates inside a given time range. The range is defined by "Slew rate" and "±Delta".

**OUTRange**
Triggers on slew rates outside a given time range. The range definition is the same as for "Within" range.

**LTHan**
Triggers on slew rates shorter than the given "Slew rate" limit.

**GTHan**
Triggers on slew rates longer than the given "Slew rate" limit.

*RST: GTHan

**Parameters for setting and query:**

<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Slew rate, ±Delta" on page 460

**SEARch:TRIGger:SLEWrate:SLOPe** <SearchName>,<Slope>
**SEARch:TRIGger:SLEWrate:SLOPe?** <SearchName>

Selects the edge type.

**Parameters:**

<Slope> POSitive | NEGative | EITHer

See Chapter 24.4.3, "Slope parameter", on page 1194.

*RST: POSitive

**Parameters for setting and query:**

<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Slew rate, ±Delta" on page 460

**SEARch:TRIGger:SLEWrate:TIME** <SearchName>,<Time>
**SEARch:TRIGger:SLEWrate:TIME?** <SearchName>

For the ranges "Within" and "Outside", the slew rate defines the center of a range which is defined by the limits "±Delta".

For the ranges "Shorter" and "Longer", the slew rate defines the maximum and minimum slew rate limits, respectively.

The range is defined using `SEARch:TRIGger:SLEWrate:RANGe`.

**Parameters:**

<Time> Range: 100E-12 to 864
Increment: 100E-9
*RST: 100E-12
Default unit: s

**Parameters for setting and query:**

<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Slew rate, ±Delta" on page 460

---

**SEARch:TRIGger:LEVel:TRANsition:LOWer** <SearchName>, <SignalSource>, <Value>
**SEARch:TRIGger:LEVel:TRANsition:LOWer?** <Key>, <SignalSource>
**SEARch:TRIGger:LEVel:TRANsition:UPPer** <SearchName>, <SignalSource>, <Value>
**SEARch:TRIGger:LEVel:TRANsition:UPPer?** <Key>, <SignalSource>

Set the lower and upper voltage thresholds, respectively. When the signal crosses this level, the slew rate measurement starts or stops depending on the selected slope.

**Parameters:**

<Value> Voltage value

**Parameters for setting and query:**

<SearchName> String with the name of the search

<SignalSource> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4

Source of the search, see Chapter 24.4.2, "Waveform parameter", on page 1193

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Lower level" on page 460

### 24.15.8 Timeout search conditions

Trigger level setting: `SEARch:TRIGger:LEVel[:VALue]` on page 1569

SEARch:TRIGger:TIMeout:RANGe ..... 1579
SEARch:TRIGger:TIMeout:TIME ..... 1579

**SEARch:TRIGger:TIMeout:RANGe** <SearchName>,<TimeoutMode>
**SEARch:TRIGger:TIMeout:RANGe?** <SearchName>

Selects the relation of the signal level to the trigger level:

**Parameters:**
<TimeoutMode> HIGH | LOW | EITHer

**HIGH**
The signal level stays above the trigger level.

**LOW**
The signal level stays below the trigger level.

**EITHer**
The signal level stays above or below the trigger level.

*RST: HIGH

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Range, Time" on page 458

**SEARch:TRIGger:TIMeout:TIME** <SearchName>,<Time>
**SEARch:TRIGger:TIMeout:TIME?** <SearchName>

Defines the time limit for the timeout at which the instrument triggers.

**Parameters:**
<Time> Range: 100E-12 to 864
Increment: 100E-9
*RST: 100E-9
Default unit: s

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Range, Time" on page 458

### 24.15.9 Width search conditions

Trigger level setting: `SEARch:TRIGger:LEVel[:VALue]` on page 1569

SEARch:TRIGger:WIDTh:DELTa ..... 1580
SEARch:TRIGger:WIDTh:POLarity ..... 1580
SEARch:TRIGger:WIDTh:RANGe ..... 1580
SEARch:TRIGger:WIDTh:WIDTh ..... 1581

**SEARch:TRIGger:WIDTh:DELTa** <SearchName>,<WidthDelta>
**SEARch:TRIGger:WIDTh:DELTa?** <SearchName>

Defines a range around the given width value (see also `SEARch:TRIGger:WIDTh:WIDTh` on page 1581).

**Parameters:**
<WidthDelta>
Range: 0 to 432
Increment: 500E-12
*RST: 0
Default unit: s

**Parameters for setting and query:**
<SearchName> Search definition

**Example:** See Chapter 24.3.4.1, "Searching for a pulse of specified width", on page 1179

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Width, ±Delta" on page 455

---

**SEARch:TRIGger:WIDTh:POLarity** <SearchName>,<Polarity>
**SEARch:TRIGger:WIDTh:POLarity?** <SearchName>

Indicates the polarity of a pulse, that is the direction of the first pulse slope.

**Parameters:**
<Polarity>
POSitive | NEGative | EITHer
See Chapter 24.4.4, "Polarity parameter", on page 1194.
*RST: POSitive

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Width, ±Delta" on page 455

---

**SEARch:TRIGger:WIDTh:RANGe** <SearchName>,<RangeMode>
**SEARch:TRIGger:WIDTh:RANGe?** <SearchName>

Selects how the range of a pulse width is defined in relation to the width and delta (see `SEARch:TRIGger:WIDTh:WIDTh` on page 1581 and `SEARch:TRIGger:WIDTh:DELTa` on page 1580).

**Parameters:**
<RangeMode>
WITHin | OUTSide | SHORter | LONGer

**WITHin**
Triggers on pulses inside a given range. The range of the pulse width is defined by "Width" and "±Delta".

**OUTSide**
Triggers on pulses outside a given range. The range definition is the same as for "Within" range.

**SHORter**
Triggers on pulses shorter than the given "Width".

**LONGer**
Triggers on pulses longer than the given "Width".

*RST: WITHin

**Parameters for setting and query:**
<SearchName> Search definition

**Example:** See Chapter 24.3.4.1, "Searching for a pulse of specified width", on page 1179

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Width, ±Delta" on page 455

---

**SEARch:TRIGger:WIDTh:WIDTh** <SearchName>,<Width>
**SEARch:TRIGger:WIDTh:WIDTh?** <SearchName>

For the ranges "Within" and "Outside", the width defines the center of a range which is defined by the limits "±Delta".

For the ranges "Shorter" and "Longer", the width defines the maximum and minimum pulse width, respectively.

The range is defined using `SEARch:TRIGger:WIDTh:RANGe`.

**Parameters:**
<Width> Range: 100E-12 to 864
Increment: 100E-9
*RST: 5E-9
Default unit: s

**Parameters for setting and query:**
<SearchName> Search definition

**Example:** See Chapter 24.3.4.1, "Searching for a pulse of specified width", on page 1179

**Usage:** Asynchronous command

**Manual operation:** See "Polarity, Range, Width, ±Delta" on page 455

## 24.15.10 Window search conditions

SEARch:TRIGger:WINDow:DELTa....1582
SEARch:TRIGger:WINDow:RANGe....1582
SEARch:TRIGger:WINDow:TIMerange....1583
SEARch:TRIGger:WINDow:WIDTh....1583

SEARch:TRIGger:LEVel:WINDow:LOWer....1584
SEARch:TRIGger:LEVel:WINDow:UPPer....1584

**SEARch:TRIGger:WINDow:DELTa** <SearchName>,<WidthDelta>
**SEARch:TRIGger:WINDow:DELTa?** <SearchName>

Defines a range around the "Width" value (see `SEARch:TRIGger:WINDow:WIDTh` on page 1583).

**Parameters:**

<WidthDelta>
Range: 0 to 432
Increment: 500E-12
*RST: 0
Default unit: s

**Parameters for setting and query:**

<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Time condition, Width, ±Delta" on page 457

**SEARch:TRIGger:WINDow:RANGe** <SearchName>,<RangeMode>
**SEARch:TRIGger:WINDow:RANGe?** <SearchName>

Selects how the signal run is compared with the window.

**Parameters:**

<RangeMode> ENTer | EXIT | WITHin | OUTSide

**ENTer**
Triggers when the signal crosses the upper or lower level and thus enters the window made up of these two levels.

**EXIT**
Triggers when the signal leaves the window.

**WITHin**
Triggers if the signal stays between the upper and lower level for a specified time. The time is defined in various ways by the `SEARch:TRIGger:WINDow:TIMerange` command.

**OUTSide**
Triggers if the signal stays above the upper level or below the lower level for a specified time. The time is also defined by the `SEARch:TRIGger:WINDow:TIMerange` command.

*RST: ENTer

**Parameters for setting and query:**

<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Vertical condition" on page 457

**SEARch:TRIGger:WINDow:TIMerange** <SearchName>,<TimeRangeMode>
**SEARch:TRIGger:WINDow:TIMerange?** <SearchName>

Selects how the time limit of the window is defined. Time conditioning is available for the vertical conditions "WITHin" and "OUTSide" (see `SEARch:TRIGger:WINDow:RANGe` on page 1582).

**Parameters:**

<TimeRangeMode> WITHin | OUTSide | SHORter | LONGer

**WITHin**
Triggers if the signal stays inside or outside the vertical window limits at least for the time *Width - Delta* and for *Width + Delta* at the most.

**OUTSide**
"Outside" is the opposite definition of "Within". The instrument triggers if the signal stays inside or outside the vertical window limits for a time shorter than *Width - Delta* or longer than *Width + Delta*.

**SHORter**
Triggers if the signal crosses vertical limits before the specified "Width" time is reached.

**LONGer**
Triggers if the signal crosses vertical limits before the specified "Width" time is reached.

*RST: WITHin

**Parameters for setting and query:**

<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Time condition, Width, ±Delta" on page 457

---

**SEARch:TRIGger:WINDow:WIDTh** <SearchName>,<Width>
**SEARch:TRIGger:WINDow:WIDTh?** <SearchName>

For the ranges "Within" and "Outside", the width defines the center of a time range which is defined by the limits "±Delta".

For the ranges "Shorter" and "Longer", it defines the maximum and minimum time lapse, respectively.

The range is defined using `SEARch:TRIGger:WINDow:RANGe`.

**Parameters:**

<Width>
Range: 100E-12 to 864
Increment: 100E-9
*RST: 5E-9
Default unit: s

**Parameters for setting and query:**

<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Time condition, Width, ±Delta" on page 457

---

**SEARch:TRIGger:LEVel:WINDow:LOWer** <SearchName>, <SignalSource>, <Value>
**SEARch:TRIGger:LEVel:WINDow:LOWer?** <Key>, <SignalSource>
**SEARch:TRIGger:LEVel:WINDow:UPPer** <SearchName>, <SignalSource>, <Value>
**SEARch:TRIGger:LEVel:WINDow:UPPer?** <Key>, <SignalSource>

Set the lower and upper voltage limits for the window.

**Parameters:**
<Value> Voltage value

**Parameters for setting and query:**
<SearchName> String with the name of the search

<SignalSource> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4

Source of the search, see Chapter 24.4.2, "Waveform parameter", on page 1193

**Usage:** Asynchronous command

**Manual operation:** See "Upper level, Lower level" on page 457

## 24.15.11 Data2clock search conditions

Data level setting: `SEARch:TRIGger:LEVel[:VALue]` on page 1569

SEARch:TRIGger:SETHold:CEDGe....1584
SEARch:TRIGger:SETHold:CLEVel....1585
SEARch:TRIGger:SETHold:CSOurce....1585
SEARch:TRIGger:SETHold:HTIMe....1585
SEARch:TRIGger:SETHold:STIMe....1586

---

**SEARch:TRIGger:SETHold:CEDGe** <SearchName>,<ClockEdge>
**SEARch:TRIGger:SETHold:CEDGe?** <SearchName>

Sets the edge of the clock signal to define the time reference point for the setup and hold time.

**Parameters:**
<ClockEdge> POSitive | NEGative | EITHer

See Chapter 24.4.3, "Slope parameter", on page 1194.

*RST: POSitive

**Parameters for setting and query:**
<SearchName>

**Usage:** Asynchronous command

**Manual operation:** See "Clock source, Clock edge, Clock level" on page 461

---

**SEARch:TRIGger:SETHold:CLEVel** <SearchName>,<ClockLevel>
**SEARch:TRIGger:SETHold:CLEVel?** <SearchName>

Sets the voltage level for the clock signal. Both this command and `SEARch:TRIGger:SETHold:CEDGe` define the starting point for calculation of the setup and hold time.

**Parameters:**
<ClockLevel>
Range: -10 to 10
Increment: 1E-3
*RST: 0
Default unit: V

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Clock source, Clock edge, Clock level" on page 461

---

**SEARch:TRIGger:SETHold:CSOurce** <SearchName>,<ClockSource>
**SEARch:TRIGger:SETHold:CSOurce?** <SearchName>

Selects the waveform used for the clock signal.

**Parameters:**
<ClockSource>
C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

Source of the clock signal, see Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C1W1

**Parameters for setting and query:**
<SearchName> Search definition name

**Usage:** Asynchronous command

**Manual operation:** See "Clock source, Clock edge, Clock level" on page 461

---

**SEARch:TRIGger:SETHold:HTIMe** <SearchName>,<HoldTime>
**SEARch:TRIGger:SETHold:HTIMe?** <SearchName>

Sets the minimum time **after** the clock edge while the data signal must stay steady above or below the data level.

The hold time can be negative. In this case, the setup time is always positive. The setup/hold interval starts before the clock edge (setup time) and ends before the clock edge (hold time). If you change the negative hold time, the setup time is adjusted by the intrument.

**Parameters:**

<HoldTime>
Range: -99.999E-9 to 0.1
Increment: 1E-9
*RST: 0
Default unit: s

**Parameters for setting and query:**

<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Setup time, Hold time" on page 461

---

**SEARch:TRIGger:SETHold:STIMe** <SearchName>,<SetupTime>
**SEARch:TRIGger:SETHold:STIMe?** <SearchName>

Sets the minimum time **before** the clock edge while the data signal must stay steady above or below the data level.

The setup time can be negative. In this case, the hold time is always positive. The setup/hold interval starts after the clock edge (setup time) and ends after the clock edge (hold time). If you change the negative setup time, the hold time is adjusted by the intrument.

**Parameters:**

<SetupTime>
Range: -99.999E-9 to 0.1
Increment: 1E-9
*RST: 0
Default unit: s

**Parameters for setting and query:**

<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Setup time, Hold time" on page 461

### 24.15.12 Pattern search conditions

SEARch:TRIGger:PATTern:A[:ENABle] ..... 1587
SEARch:TRIGger:PATTern:B[:ENABle] ..... 1587
SEARch:TRIGger:PATTern:C[:ENABle] ..... 1587
SEARch:TRIGger:PATTern:D[:ENABle] ..... 1587
SEARch:TRIGger:PATTern:A:LOGic ..... 1587
SEARch:TRIGger:PATTern:B:LOGic ..... 1587
SEARch:TRIGger:PATTern:C:LOGic ..... 1587
SEARch:TRIGger:PATTern:D:LOGic ..... 1587
SEARch:TRIGger:PATTern:AB:LOGic ..... 1588

SEARch:TRIGger:PATTern:CD:LOGic....1588
SEARch:TRIGger:PATTern:ABCD:LOGic....1588
SEARch:TRIGger:PATTern:MODE....1588
SEARch:TRIGger:PATTern:TIMeout:MODE....1589
SEARch:TRIGger:PATTern:TIMeout[:TIME]....1589
SEARch:TRIGger:PATTern:WIDTh:RANGe....1590
SEARch:TRIGger:PATTern:WIDTh[:WIDTh]....1590
SEARch:TRIGger:PATTern:WIDTh:DELTa....1591

---

**SEARch:TRIGger:PATTern:A[:ENABle]** <Searchname>, <State>
**SEARch:TRIGger:PATTern:A[:ENABle]?** <Searchname>
**SEARch:TRIGger:PATTern:B[:ENABle]** <Searchname>, <State>
**SEARch:TRIGger:PATTern:B[:ENABle]?** <Searchname>
**SEARch:TRIGger:PATTern:C[:ENABle]** <Searchname>, <State>
**SEARch:TRIGger:PATTern:C[:ENABle]?** <Searchname>
**SEARch:TRIGger:PATTern:D[:ENABle]** <Searchname>, <State>
**SEARch:TRIGger:PATTern:D[:ENABle]?** <Searchname>

Enables the channel to be considered in the pattern search. The trigger source channel is selected by default.

- A[:ENABle]: CH1
- B[:ENABle]: CH2
- C[:ENABle]: CH3
- D[:ENABle]: CH4

Digital channels are not available.

**Parameters:**
<State> ON | OFF

**Parameters for setting and query:**
<Searchname> String with name of the search

**Usage:** Asynchronous command

---

**SEARch:TRIGger:PATTern:A:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:PATTern:A:LOGic?** <Searchname>
**SEARch:TRIGger:PATTern:B:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:PATTern:B:LOGic?** <Searchname>
**SEARch:TRIGger:PATTern:C:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:PATTern:C:LOGic?** <Searchname>
**SEARch:TRIGger:PATTern:D:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:PATTern:D:LOGic?** <Searchname>

Defines the logic for the indicated channel:

- A: CH1
- B: CH2
- C: CH3

- D: CH4

**Parameters:**

<Operator> DIRect | HIGH | NOT | LOW

**DIRect**
Input value remains unchanged

**NOT**
Input value is inverted

**Parameters for setting and query:**

<Searchname> String with the name of the search

**Usage:** Asynchronous command

---

**SEARch:TRIGger:PATTern:AB:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:PATTern:AB:LOGic?** <Searchname>
**SEARch:TRIGger:PATTern:CD:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:PATTern:CD:LOGic?** <Searchname>
**SEARch:TRIGger:PATTern:ABCD:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:PATTern:ABCD:LOGic?** <Searchname>

Defines the logical combination of the indicated channels after evaluating the previous logical operations:

- AB: CH1 and CH2
- CD: CH3 and CH4
- ABCD: result of AB and CD

**Parameters:**

<Operator> AND | NAND | OR | NOR

AND: logical AND, conjunctive combination
NAND:logical NOT AND
OR:logical OR, disjunctive combination
NOR: logical NOT OR

**Parameters for setting and query:**

<Searchname> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Pattern" on page 463

---

**SEARch:TRIGger:PATTern:MODE** <SearchName>,<Mode>
**SEARch:TRIGger:PATTern:MODE?** <SearchName>

Adds additional time limitation to the pattern definition.

**Parameters:**

<Mode> OFF | TIMeout | WIDTh

**OFF**
No time limitation. The event is found if the pattern condition is fulfilled.

**TIMeout**
Defines how long the result of the pattern condition stays high or low. The duration of the timeout is defined using `SEARch:TRIGger:PATTern:TIMeout[:TIME]` The result state is defined using `SEARch:TRIGger:PATTern:TIMeout:MODE`.

**WIDTh**
Defines a time range for keeping up the true result of the pattern condition. The range is defined using `SEARch:TRIGger:PATTern:WIDTh:RANGe`.

*RST: OFF

**Parameters for setting and query:**
<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Timing condition: Mode, Range, Time, Width, ±Delta" on page 463

---

**SEARch:TRIGger:PATTern:TIMeout:MODE** <SearchName>,<TimeoutMode>
**SEARch:TRIGger:PATTern:TIMeout:MODE?** <SearchName>

Defines the condition for the timeout.

**Parameters:**
<TimeoutMode> HIGH | LOW | EITHer

**EITHer**
High or low, the pattern remains unchanged for the given time.

*RST: HIGH

**Parameters for setting and query:**
<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Timing condition: Mode, Range, Time, Width, ±Delta" on page 463

---

**SEARch:TRIGger:PATTern:TIMeout[:TIME]** <SearchName>,<Time>
**SEARch:TRIGger:PATTern:TIMeout[:TIME]?** <SearchName>

Defines how long the result of the pattern condition must keep the given state.

**Parameters:**
<Time> Range: 100E-12 to 864
Increment: 100E-9
*RST: 100E-9
Default unit: s

**Parameters for setting and query:**
<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Timing condition: Mode, Range, Time, Width, ±Delta" on page 463

---

**SEARch:TRIGger:PATTern:WIDTh:RANGe** <SearchName>,<WidthRangeMode>
**SEARch:TRIGger:PATTern:WIDTh:RANGe?** <SearchName>

Defines the time range of a pulse width for keeping up the true result of the pattern condition. The width and delta are specified using `SEARch:TRIGger:PATTern:WIDTh[:WIDTh]` and `SEARch:TRIGger:PATTern:WIDTh:DELTa`.

**Parameters:**

<WidthRangeMode> WITHin | OUTSide | SHORter | LONGer

**WITHin | OUTSide**
Triggers on pulses inside or outside a given range. The range is defined by the width ±delta.

**SHORter | LONGer**
Triggers on pulses shorter or longer than the given width.

*RST: WITHin

**Parameters for setting and query:**

<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Timing condition: Mode, Range, Time, Width, ±Delta" on page 463

---

**SEARch:TRIGger:PATTern:WIDTh[:WIDTh]** <SearchName>,<Width>
**SEARch:TRIGger:PATTern:WIDTh[:WIDTh]?** <SearchName>

For the ranges WITHin and OUTSide, the width defines the center of a range that is defined by the limits ±delta.

For the ranges SHORter and LONGer, the width defines the maximum and minimum pulse width, respectively.

To set the range mode, use `SEARch:TRIGger:PATTern:WIDTh:RANGe`. To set the delta value, use `SEARch:TRIGger:PATTern:WIDTh:DELTa`.

**Parameters:**

<Width>
Range: 100E-12 to 864
Increment: 100E-9
*RST: 5E-9
Default unit: s

**Parameters for setting and query:**

<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Timing condition: Mode, Range, Time, Width, ±Delta" on page 463

**SEARch:TRIGger:PATTern:WIDTh:DELTa** <SearchName>,<WidthDelta>
**SEARch:TRIGger:PATTern:WIDTh:DELTa?** <SearchName>

Defines a range around the width value specified using `SEARch:TRIGger:PATTern:WIDTh[:WIDTh]`.

**Parameters:**

<WidthDelta>
Range: 0 to 432
Increment: 500E-12
*RST: 0
Default unit: s

**Parameters for setting and query:**

<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Timing condition: Mode, Range, Time, Width, ±Delta" on page 463

### 24.15.13 State search conditions

SEARch:TRIGger:STATe:CSOurce ..... 1591
SEARch:TRIGger:STATe:CEDGe ..... 1592
SEARch:TRIGger:STATe:CLEVel ..... 1592
SEARch:TRIGger:STATe:A[:ENABle] ..... 1592
SEARch:TRIGger:STATe:B[:ENABle] ..... 1592
SEARch:TRIGger:STATe:C[:ENABle] ..... 1592
SEARch:TRIGger:STATe:D[:ENABle] ..... 1592
SEARch:TRIGger:STATe:A:LOGic ..... 1593
SEARch:TRIGger:STATe:B:LOGic ..... 1593
SEARch:TRIGger:STATe:C:LOGic ..... 1593
SEARch:TRIGger:STATe:D:LOGic ..... 1593
SEARch:TRIGger:STATe:AB:LOGic ..... 1593
SEARch:TRIGger:STATe:CD:LOGic ..... 1593
SEARch:TRIGger:STATe:ABCD:LOGic ..... 1593

**SEARch:TRIGger:STATe:CSOurce** <SearchName>,<Source>
**SEARch:TRIGger:STATe:CSOurce?** <SearchName>

Sets the source of the clock signal.

**Parameters:**

<Source>
C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

**Parameters for setting and query:**

<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Clock source, Clock edge" on page 462

**SEARch:TRIGger:STATe:CEDGe** <SearchName>,<ClockEdge>
**SEARch:TRIGger:STATe:CEDGe?** <SearchName>

Sets the trigger edge of the clock signal.

**Parameters:**
<ClockEdge> POSitive | NEGative | EITHer
*RST: POSitive

**Parameters for setting and query:**
<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Clock source, Clock edge" on page 462

**SEARch:TRIGger:STATe:CLEVel** <SearchName>,<ClockLevel>
**SEARch:TRIGger:STATe:CLEVel?** <SearchName>

Sets the trigger level of the clock signal.

The command has the same effect as with `SEARch:TRIGger:LEVel[:VALue]`.

**Parameters:**
<ClockLevel> Range: -10 to 10
Increment: 1E-3
*RST: 0
Default unit: V

**Parameters for setting and query:**
<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Clock source, Clock edge" on page 462

**SEARch:TRIGger:STATe:A[:ENABle]** <Searchname>, <State>
**SEARch:TRIGger:STATe:A[:ENABle]?** <Searchname>
**SEARch:TRIGger:STATe:B[:ENABle]** <Searchname>, <State>
**SEARch:TRIGger:STATe:B[:ENABle]?** <Searchname>
**SEARch:TRIGger:STATe:C[:ENABle]** <Searchname>, <State>
**SEARch:TRIGger:STATe:C[:ENABle]?** <Searchname>
**SEARch:TRIGger:STATe:D[:ENABle]** <Searchname>, <State>
**SEARch:TRIGger:STATe:D[:ENABle]?** <Searchname>

Enables the channel to be considered in the state search. You can enable all channel signals except for the trigger source.

- A[:ENABle]: CH1
- B[:ENABle]: CH2
- C[:ENABle]: CH3
- D[:ENABle]: CH4

Digital channels are not available.

**Parameters:**
<State> ON | OFF

**Parameters for setting and query:**
<Searchname> String with the name of the search

**Usage:** Asynchronous command

---

**SEARch:TRIGger:STATe:A:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:STATe:A:LOGic?** <Searchname>
**SEARch:TRIGger:STATe:B:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:STATe:B:LOGic?** <Searchname>
**SEARch:TRIGger:STATe:C:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:STATe:C:LOGic?** <Searchname>
**SEARch:TRIGger:STATe:D:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:STATe:D:LOGic?** <Searchname>

Defines the logic for the indicated channel:

- A: CH1
- B: CH2
- C: CH3
- D: CH4

**Parameters:**
<Operator> DIRect | HIGH | NOT | LOW

**DIRect**
Input value remains unchanged

**NOT**
Input value is inverted

**Parameters for setting and query:**
<Searchname> String with the name of the search

**Usage:** Asynchronous command

---

**SEARch:TRIGger:STATe:AB:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:STATe:AB:LOGic?** <Searchname>
**SEARch:TRIGger:STATe:CD:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:STATe:CD:LOGic?** <Searchname>
**SEARch:TRIGger:STATe:ABCD:LOGic** <Searchname>, <Operator>
**SEARch:TRIGger:STATe:ABCD:LOGic?** <Searchname>

Defines the logical combination of the indicated channels after evaluating the previous logical operations:

- AB: CH1 and CH2
- CD: CH3 and CH4
- ABCD: result of AB and CD

**Parameters:**
<Operator> AND | NAND | OR | NOR

AND: logical AND, conjunctive combination
NAND:logical NOT AND
OR:logical OR, disjunctive combination
NOR: logical NOT OR

**Parameters for setting and query:**
<Searchname> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "State pattern" on page 462

### 24.15.14 Search on spectrum

**CURSor<m>:PEXCursion** <Value>

Defines the minimum level by which the waveform must rise or fall so that it will be identified as a maximum or a minimum by the search functions.

**Suffix:**
<m> The suffix is irrelevant

**Parameters:**
<Value>
Range: 0 to 100
Increment: 1
*RST: 5
Default unit: dB

**Usage:** Asynchronous command

**Manual operation:** See "Peak excursion" on page 319

**CURSor<m>:THReshold** <Value>

Defines an absolute threshold as an additional condition for the peak search. Only peaks that exceed the threshold are detected.

**Suffix:**
<m> The suffix is irrelevant.

**Parameters:**
<Value> Threshold in dBm

**Usage:** Asynchronous command

**Firmware/software:** Version 2.70

**Manual operation:** See "Threshold" on page 318

### 24.15.15 Search gate settings

SEARch:GATE[:STATe] ..... 1595
SEARch:GATE:MODE ..... 1595
SEARch:GATE:SHOW ..... 1595
SEARch:GATE:ABSolute:STARt ..... 1596
SEARch:GATE:ABSolute:STOP ..... 1596
SEARch:GATE:RELative:STARt ..... 1596
SEARch:GATE:RELative:STOP ..... 1597
SEARch:GATE:ZCOupling ..... 1597
SEARch:GATE:ZDIagram ..... 1597

---

**SEARch:GATE[:STATe]** <SearchName>,<State>
**SEARch:GATE[:STATe]?** <SearchName>

Performs the search only on the defined gate area of the source waveform.

**Parameters:**
<State> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Use gate" on page 466

---

**SEARch:GATE:MODE** <SearchName>,<Mode>
**SEARch:GATE:MODE?** <SearchName>

Defines whether the gate settings are configured using absolute or relative values.

**Parameters:**
<Mode> ABS | REL
*RST: ABS

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Gate description" on page 330

---

**SEARch:GATE:SHOW** <SearchName>,<DisplayState>
**SEARch:GATE:SHOW?** <SearchName>

If enabled, the gate area is indicated in the source diagram.

**Parameters:**
<DisplayState> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Display state" on page 466

---

**SEARch:GATE:ABSolute:STARt** <SearchName>,<Start>
**SEARch:GATE:ABSolute:STARt?** <SearchName>

Defines the starting value for the gate.

**Parameters:**
<Start> Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "(Relative) Start" on page 330

---

**SEARch:GATE:ABSolute:STOP** <SearchName>,<Stop>
**SEARch:GATE:ABSolute:STOP?** <SearchName>

Defines the end value for the gate.

**Parameters:**
<Stop> Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "(Relative) Stop" on page 331

---

**SEARch:GATE:RELative:STARt** <SearchName>,<RelativeStart>
**SEARch:GATE:RELative:STARt?** <SearchName>

Defines the starting value for the gate.

**Parameters:**
<RelativeStart> Range: 0 to 100
Increment: 0.1
*RST: 0
Default unit: %

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "(Relative) Start" on page 330

---

**SEARch:GATE:RELative:STOP** <SearchName>,<RelativeStop>
**SEARch:GATE:RELative:STOP?** <SearchName>

Defines the end value for the gate.

**Parameters:**
<RelativeStop> Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "(Relative) Stop" on page 331

---

**SEARch:GATE:ZCOupling** <SearchName>,<ZoomCoupling>
**SEARch:GATE:ZCOupling?** <SearchName>

If enabled, the gate area is set to the limits of a zoom area.

The zoom diagramm is selected using `SEARch:GATE:ZDIagram`.

**Parameters:**
<ZoomCoupling> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName> String with the name of the search

**Usage:** Asynchronous command

**Manual operation:** See "Zoom" on page 331

---

**SEARch:GATE:ZDIagram** <SearchName>, <DiagramName>, <ZoomName>
**SEARch:GATE:ZDIagram?** <SearchName>

Selects the zoom to which the gate area is set if `SEARch:GATE:ZCOupling` is set to "ON".

**Parameters:**
<ZoomDiagram> String with the name of the diagram, where the zoom is defined

**Parameters for setting and query:**
<SearchName> String with the name of the search

**Example:** Prerequisite: Search1 and Zoom2 are available.

```
SEARch:GATE:ZCOupling 'Search1',ON
SEARch:GATE:ZDIagram 'Search1','Diagram1','Zoom2'
SEARch:GATE:ZDIagram? 'Search1'
<-- Diagram1;Zoom2
SEARch:GATE:STATe 'Search1', ON
SEARch:GATE:SHOW 'Search2', ON
```

Enables the zoom coupling to define the gate,and selects Zoom2 as gate area. Zoom2 is based on Diagram1. Then the gate is activated and displayed.

**Usage:** Asynchronous command

**Manual operation:** See "Zoom" on page 331

## 24.15.16 Search results

SEARch:RESDiagram:HORZ:ABSolute:POSition....1598
SEARch:RESDiagram:HORZ:ABSolute:SPAN....1599
SEARch:RESDiagram:HORZ:MODE....1599
SEARch:RESDiagram:HORZ:RELative:POSition....1599
SEARch:RESDiagram:HORZ:RELative:SPAN....1600
SEARch:RESDiagram:SHOW....1600
SEARch:RESDiagram:VERT:ABSolute:POSition....1600
SEARch:RESDiagram:VERT:ABSolute:SPAN....1601
SEARch:RESDiagram:VERT:MODE....1601
SEARch:RESDiagram:VERT:RELative:POSition....1601
SEARch:RESDiagram:VERT:RELative:SPAN....1602
SEARch:RESult:LIMit....1602
SEARch:RESult:SHOW....1602
SEARch:RESult:SORT:ASCending....1603
SEARch:RESult:SORT[:MODE]....1603
SEARch:RESult[:ALL]?....1603

---

**SEARch:RESDiagram:HORZ:ABSolute:POSition** <SearchName>,<Position>
**SEARch:RESDiagram:HORZ:ABSolute:POSition?** <SearchName>

Defines the x-value of the centerpoint of the zoom area.

**Parameters:**

<Position>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**

<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "X position" on page 265

**SEARch:RESDiagram:HORZ:ABSolute:SPAN** <SearchName>,<Span>
**SEARch:RESDiagram:HORZ:ABSolute:SPAN?** <SearchName>

Defines the width of the zoom area.

**Parameters:**
<Span>
Range: 0 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "X range" on page 266

**SEARch:RESDiagram:HORZ:MODE** <SearchName>,<Mode>
**SEARch:RESDiagram:HORZ:MODE?** <SearchName>

Defines whether absolute or relative values are used to specify the x-axis values.

**Parameters:**
<Mode> ABS | REL
*RST: ABS

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "X mode" on page 266

**SEARch:RESDiagram:HORZ:RELative:POSition** <SearchName>,<RelPosi>
**SEARch:RESDiagram:HORZ:RELative:POSition?** <SearchName>

Defines the x-value of the centerpoint of the zoom area.

**Parameters:**
<RelPosi>
Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "X position" on page 265

**SEARch:RESDiagram:HORZ:RELative:SPAN** <SearchName>,<RelativeSpan>
**SEARch:RESDiagram:HORZ:RELative:SPAN?** <SearchName>

Defines the width of the zoom area.

**Parameters:**
<RelativeSpan> Range: 1E-15 to 100
Increment: 0.1
*RST: 1
Default unit: %

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "X range" on page 266

**SEARch:RESDiagram:SHOW** <SearchName>,<ShwSearchWind>
**SEARch:RESDiagram:SHOW?** <SearchName>

If enabled, a zoom window is displayed for the currently selected search result. The zoom area is indicated in the diagram that displays the source waveform of the search.

**Parameters:**
<ShwSearchWind> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Show result zoom" on page 469

**SEARch:RESDiagram:VERT:ABSolute:POSition** <SearchName>,<Position>
**SEARch:RESDiagram:VERT:ABSolute:POSition?** <SearchName>

Defines the y-value of the centerpoint of the zoom area.

**Parameters:**
<Position> Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Y position" on page 266

**SEARch:RESDiagram:VERT:ABSolute:SPAN** <SearchName>,<Span>
**SEARch:RESDiagram:VERT:ABSolute:SPAN?** <SearchName>

Defines the height of the zoom area.

**Parameters:**
<Span>
Range: 0 to 100E+24
Increment: 0.01
*RST: 0.01

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Y range" on page 266

**SEARch:RESDiagram:VERT:MODE** <SearchName>,<Mode>
**SEARch:RESDiagram:VERT:MODE?** <SearchName>

Defines whether absolute or relative values are used to specify the y-axis values.

**Parameters:**
<Mode> ABS | REL
*RST: ABS

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Y mode" on page 266

**SEARch:RESDiagram:VERT:RELative:POSition** <SearchName>,<RelPosi>
**SEARch:RESDiagram:VERT:RELative:POSition?** <SearchName>

Defines the y-value of the centerpoint of the zoom area.

**Parameters:**
<RelPosi>
Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Y position" on page 266

**SEARch:RESDiagram:VERT:RELative:SPAN** <SearchName>,<RelativeSpan>
**SEARch:RESDiagram:VERT:RELative:SPAN?** <SearchName>

Defines the height of the zoom area.

**Parameters:**
<RelativeSpan> Range: 1E-15 to 100
Increment: 0.1
*RST: 1
Default unit: %

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Y range" on page 266

---

**SEARch:RESult:LIMit** <SearchName>,<ResultListLimit>
**SEARch:RESult:LIMit?** <SearchName>

Defines the maximum number of entries in the search result table.

**Parameters:**
<ResultListLimit> Range: 1 to 1000
Increment: 1
*RST: 100

**Parameters for setting and query:**
<SearchName> Search definition

**Example:** See Chapter 24.3.4.1, "Searching for a pulse of specified width", on page 1179

**Usage:** Asynchronous command

**Manual operation:** See "Max result count" on page 469

---

**SEARch:RESult:SHOW** <SearchName>,<ShowResultTable>
**SEARch:RESult:SHOW?** <SearchName>

Displays or hides the search result table.

**Parameters:**
<ShowResultTable> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Show result table" on page 469

**SEARch:RESult:SORT:ASCending** <SearchName>,<SortAscending>
**SEARch:RESult:SORT:ASCending?** <SearchName>

If enabled, the results are listed in asscending order, i.e. the smallest value at the top.

**Parameters:**
<SortAscending> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Sort ascending" on page 469

**SEARch:RESult:SORT[:MODE]** <SearchName>,<SortMode>
**SEARch:RESult:SORT[:MODE]?** <SearchName>

Sorts the search result table by x-value position or value of the result.

**Parameters:**
<SortMode> POSition | VALue

**POSition**
Sorts the search result table by the x-value position.

**VALue**
Sorts the search result table by the value of the result.

*RST: POSition

**Parameters for setting and query:**
<SearchName> Search definition

**Usage:** Asynchronous command

**Manual operation:** See "Sort by" on page 469

**SEARch:RESult[:ALL]?** <SearchName>

Returns all search results.

**Query parameters:**
<SearchName> Search definition

**Return values:**
<Data> List of search results, separated by commas. For each result, six values are returned:
1. Acquisition index, currently always 0.
2. X-position of the search result
3. Y-position of the search result, currently not relevant
4. Type of the search result (Edge, Glitch, ...)
5. Slope or polarity of the search result

6. For runt, glitch, width, and window searches, the value contains the width. For timeout and interval searches, it contains the timeout. For transition searches, it contains the slew rate. For all other searches, the value is not relevant.
If a value is not relevant, 9.91E+37 is returned.

**Example:**
```
SEAR:RES? 'Search1'
0,1.5375e-007,-84,Edge,Positive,9.91E+37,
0,5.3e-008,-84,Edge,Positive,9.91E+37
```
The query returns two search results for edge search on rising edges at X-position 153,75 ns and 53 ns.

**Usage:** Query only
Asynchronous command

## 24.16 Data management

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

- Instrument settings....1604
- Savesets and one file....1612
- Waveform data transmission....1613
- Waveform data export to file....1613
- Waveform histogram export to file....1622
- Eye....1623
- Long term measurement results and measurement histogram export to file....1625
- Screenshots....1627
- Reports....1631
- Autonaming....1633
- CSV export....1634
- External application....1635

### 24.16.1 Instrument settings

The Mass MEMory subsystem provides commands to access the storage media and to save and reload instrument settings.

### File and directory names

The <file_name> and <directory_name> parameters are strings. Some commands use a fixed directory; for others the <file_name> can contain the complete path including the drive name and all subdirectories. If no complete path is specified, the file location is relative to the current directory, queried with `MMEMory:CDIRectory?`. The file name itself may contain the period as a separator for extensions.

Example: '`C:\TEMP\TRASH\test.txt`' for the file named `test.txt` in the `TEMP\TRASH` subdirectory of the internal hard disk drive C:\.

File and directory names can be chosen according to Windows™ conventions; the restrictions placed on file names known from DOS systems do not apply. All letters and numbers are allowed, as well as the special characters "_", "^", "$", "~", "!", "#", "%", "&", "-", "{", "}", "(", ")", "@" and "`". Reserved file names are CON, AUX, COM1, ..., COM4, LPT1, ..., LPT3, NUL and PRN.

The use of wildcards ? and * is not allowed.

MMEMory:DRIVes? ..... 1605
MMEMory:MSIS ..... 1605
MMEMory:DCATalog? ..... 1606
MMEMory:DCATalog:LENGth? ..... 1606
MMEMory:CDIRectory ..... 1606
MMEMory:MDIRectory ..... 1607
MMEMory:RDIRectory ..... 1607
MMEMory:CATalog? ..... 1607
MMEMory:CATalog:LENGth? ..... 1608
MMEMory:COPY ..... 1608
MMEMory:MOVE ..... 1609
MMEMory:DELete ..... 1609
MMEMory:DATA ..... 1609
MMEMory:ATTRibute ..... 1610
MMEMory:SAV ..... 1610
MMEMory:RCL ..... 1611
MMEMory:STORe:STATe ..... 1611
MMEMory:LOAD:STATe ..... 1612

---

**MMEMory:DRIVes?**

Returns a list of the logical drives of the instrument as configured in the operating system.

**Return values:**

<Drive> List of strings, for example, `"C:\","F:\","H:\"`

**Usage:** Query only

---

**MMEMory:MSIS** [<msus>]

Changes the default storage device to the indicated drive or network server.

**Parameters:**

<msus> String parameter. Drives are indicated with their drive letter, network servers require the UNC format.

**Example:**
```
MMEM:MSIS 'C:'
```

**Example:**
```
MMEM:MSIS '\\server1\share1'
```

---

**MMEMory:DCATalog?** [<PathName>]

Returns the subdirectories of the current or of a specified directory.

**Query parameters:**

<PathName> String parameter to specify the directory. If the directory is omitted, the command queries the contents of the current directory, to be set and queried with `MMEMory:CDIRectory`.

**Return values:**

<FileEntry> Names of the subdirectories separated by colons. The first two strings are related to the parent directory.

**Example:**
```
MMEM:DCAT?
".","..","Documents and Settings","Program
Files","temp"
```

**Usage:** Query only

---

**MMEMory:DCATalog:LENGth?** [<PathName>]

Returns the number of subdirectories of the current or of a specified directory. The number includes the parent directory strings "." and ".." and corresponds to the number of strings returned by the `MMEMory:DCATalog?` command.

**Query parameters:**

<PathName> String parameter to specify the directory. If the directory is omitted, the command queries the contents of the current directory, to be set and queried with `MMEMory:CDIRectory`.

**Return values:**

<FileEntryCount> Number of parent and subdirectories.

**Example:**
```
MMEM:DCAT:LENG?
5
```

**Usage:** Query only

---

**MMEMory:CDIRectory** [<DirectoryName>]

Changes the default directory for file access.

**Parameters:**

<DirectoryName> String parameter to specify the directory.
If the string also contains a drive letter or network server name, the command MMEMory:MSIS is executed implicitely.

*RST: "\"

**Example:** MMEM:CDIR 'C:\USER\DATA'

**Usage:** SCPI confirmed

---

**MMEMory:MDIRectory** <DirectoryName>

Creates a new directory with the specified name.

**Setting parameters:**

<DirectoryName> String parameter to specify the new directory. If the path consists of several subdirectories, the complete tree will be created if necessary.
If no drive letter or server name is indicated, the directory is created on the default storage device specified with MMEMory:MSIS.

**Example:** MMEM:MDIR 'C:\USER\DATA'

**Usage:** Setting only

---

**MMEMory:RDIRectory** <DirectoryName>

Deletes the specified directory.

**Setting parameters:**

<DirectoryName> String parameter to specify the directory to be deleted.
This directory must be empty, otherwise it is not deleted.

**Example:** MMEM:RDIR 'C:\USER\TEST'

**Usage:** Setting only

---

**MMEMory:CATalog?** [<PathName>][, <Format>]

Returns the a list of files contained in the specified directory. The result corresponds to the number of files returned by the MMEMory:CATalog:LENgth command.

**Query parameters:**

<PathName> String parameter to specify the directory. If the directory is omitted, the command queries directory specified with MMEMory:CDIRectory.

<Format> ALL | WTIMe

ALL: Extended result including file, date, time and attributes
WTIMe: Extended result including file, date, time

**Return values:**

<UsedMemory> Total amount of storage currently used in the directory, in bytes.

<FreeMemory> Total amount of storage available in the directory, in bytes.

<FileEntry> All files and subdirectories of the directory are listed with their file name, format and size in bytes. The first two strings are related to the parent directory.

**Example:**

```
MMEM:CAT? 'C:\USER\DATA'
529479,1831777894400,".,DIR,0","..,DIR,0",
"Backup,DIR,0","CSS,DIR,0","DATEN,DIR,0",
"Commands.jar,BIN,529479","FAVORITES,DIR,0",
"LOG,DIR,0","DATA,DIR,0","test,DIR,0",
"TotalCMD,DIR,0"
```

**Usage:** Query only
SCPI confirmed

---

**MMEMory:CATalog:LENGth?** [<PathName>]

Returns the number of files and subdirectories of the current or specified directory. The number includes the parent directory strings "." and ".." and it corresponds to the number of <FileEntry> strings returned by the `MMEMory:CATalog?` command.

**Query parameters:**

<PathName> String parameter, directory to be queried. If the directory is omitted, the current directory is queried, specified with `MMEMory:CDIRectory`.

**Return values:**

<Count> Number of files and subdirectories including parent directory entries.

**Example:**

```
MMEM:CDIR 'C:\USER\DATA'
MMEM:CAT:LENG?
11
```

**Usage:** Query only

---

**MMEMory:COPY** <FileSource>[, <FileDestination>]

Copies an existing file to a new file.

**Setting parameters:**

<FileSource> String parameter, contains name and path of the file to be copied. Wildcards (* and ?) are allowed.

<FileDestination> String parameter, contains name and path of the new file. If the file already exists, it is overwritten without notice. If no file destination is specified, the source file is written to the current directory specified with `MMEMory:CDIRectory`.

**Example:** `MMEM:COPY 'C:\Users\Public\Documents\Rohde-Schwarz\RTx\RefWaveforms\RefCurve_2011-03-16*.bin','E:'`
Copies all reference waveforms saved on March 16, 2011 to an external storage medium, mapped to drive `E:\`.

**Usage:** Setting only
SCPI confirmed

---

**MMEMory:MOVE** <FileSource>, <FileDestination>

Moves the specified file to a new location on the same drive and renames it.

**Setting parameters:**

<FileSource> String parameter, contains name and path of the file to be copied.
Wildcards (* and ?) are allowed. Therefore, specify a directory for <FileDestination>. Renaming is not possible.

<FileDestination> String parameter, contains name and path of the new file. If no path is specified, the <FileSource> directory is used - the file is renamed.

**Example:** `MMEM:MOVE 'C:\USER\DATA\SETUP.CFG','C:\STORE'`
Moves the file "Setup.cfg" from the directory `C:\USER\DATA` to `C:\STORE`.

**Usage:** Setting only
SCPI confirmed

---

**MMEMory:DELete** <FileName>

Removes the specified file(s). To delete directories, use `MMEMory:RDIRectory`.

**Setting parameters:**

<FileName> String parameter to specify the name and directory of the file to be removed. Wildcards (* and ?) are allowed.
If no path is defined, the current directory is used, specified with `MMEMory:CDIRectory`.

**Example:** `MMEM:DEL '*.CFG'`
Deletes all cfg files from the current directory.

**Usage:** Setting only
SCPI confirmed

**Manual operation:** See "Save to file" on page 510

---

**MMEMory:DATA** <FileName>, <Data>
**MMEMory:DATA?** <FileName>

Stores data in the specified file to the storage location specified using `MMEMory:CDIRectory`.

**Parameters:**

<Data> <block>

488.2 block data format. The delimiter EOI must be selected to achieve correct data transfer.
The block begins with character '#'. The next digit is the length of the length information, followed by this given number of digits providing the number of bytes in the following binary data.

**Parameters for setting and query:**

<FileName> String parameter, the name of the file the data is stored to.

**Example:** `MMEM:DATA 'abc.txt', #216This is the file`
#2: the length infomation has two digits
16: the binary data has 16 bytes

**Example:** `MMEM:DATA? 'abc.txt'`
Returns the data from file `abc.txt`.

---

**MMEMory:ATTRibute** <FileName>, <Attributes>
**MMEMory:ATTRibute?** <FileName>

Sets file attributes for the specified file(s). The command can be used for files only.

**Setting parameters:**

<Attributes> String with attributes and setting information.
'+' before the attribute: sets the attribute
'-' before the attribute: deletes the attribute
'R': read only
'A': archive file
'S': system file
'H': hidden file

**Parameters for setting and query:**

<FileName> String parameter, contains name and path of the file. Wildcards (* and ?) are allowed.

**Return values:**

<FileEntry> String containing: "<file_name>,<file_attributes>"

**Example:** `MMEM:ATTR 'C:\USER\DATA\*.LOG', '-R -A'`
Deletes the read-only and archive attributes from all LOG files in the directory`C:\USER\DATA\*.LOG`.

**Example:**
```
MMEM:ATTR? 'C:\USER\DATA\*.*'
"Datei1.LOG,A", "Datei2.LOG,A",
"Datei3.LOG,ASH", "Datei4.DLL,RSH",
"Datei5.INI,SH"
```

---

**MMEMory:SAV** <FileDestination>

Stores the current instrument settings to the specified file.

This command has the same effect as the combination of `*SAV` and `MMEMory:STORe:STATe`.

**Parameters:**

<FileDestination> String parameter specifying path and filename of the target file. Wildcards are not allowed.

**Example:** `MMEM:SAV 'C:\mysavefile.dfl'`
Saves the current instrument settings to the file `mysavefile.dfl` located in the directory C:\.

**Usage:** Event

**Manual operation:** See "Save to file" on page 476

---

**MMEMory:RCL** <FileSource>

Restores the instrument settings from the specified file.

This command has the same effect as the combination of `MMEMory:LOAD:STATe` and `*RCL`.

**Parameters:**

'<FileSource>' String parameter specifying the path and filename of the source file. Wildcards are not allowed.

**Example:** `MMEM:RCL 'C:\mysavefile.dfl'`
Loads and activates the instrument settings from the file `mysavefile.dfl` located in the directory C:\.

**Usage:** Event

**Manual operation:** See "Load from file" on page 476

---

**MMEMory:STORe:STATe** <MemoryNumber>, <FileName>

Stores the instrument settings from the specified internal memory to the specified file. To store the current instrument settings to the internal memory, use `*SAV` first.

**Setting parameters:**

<MemoryNumber> Number of the internal memory
Range: 1 to 99

<FileName> String parameter specifying the complete path and filename of the source file.

**Example:**
```
*SAV 4
MMEM:STORe:STATe 4,'C:
\Settings\Settings_1051.dfl'
```
Saves current instrument settings to the internal memory number 4. Then stores the settings from the internal memory number 4 to the file `C:\Settings\Settings_1051.dfl`.

**Usage:** Setting only

**MMEMory:LOAD:STATe** <MemoryNumber>, <FileName>

Loads the instrument settings from the specified file to the specified internal memory. After the file has been loaded, the settings must be activated using a `*RCL` command.

**Setting parameters:**

<MemoryNumber> Number of the internal memory
Range: 1 to 99

<FileName> String parameter specifying the complete path and filename of the source file.

**Example:**
```
MMEM:LOAD:STATe 4,'C:
\Settings\Settings_1051.dfl'
*RCL 4
```
Loads instrument settings from the file `C:\Settings\Settings_1051.dfl` to the internal memory number 4, and then activates the settings in internal memory number 4.

**Usage:** Setting only

## 24.16.2 Savesets and one file

SAVeset:CONFig:PREView....1612
SAVeset:ONEFile:NAME....1612
SAVeset:ONEFile:SAVE....1613
SAVeset:ONEFile:OPEN....1613
SAVeset:ONEFile:USERpref....1613

**SAVeset:CONFig:PREView** <Include>

If set to OFF, the saveset is stored without the preview image to reduce the file size.

Use the command each time before you save a saveset.

**Parameters:**

<Include> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**SAVeset:ONEFile:NAME** <Name>

Sets the path and the file name of the One File. The file format is ZIP.

**Parameters:**

<Name> String parameter

**Usage:** Asynchronous command

**Manual operation:** See "Save to file" on page 477

**SAVeset:ONEFile:SAVE**

**Usage:** Event

**Manual operation:** See "Save to file" on page 477

Saves the One File data to the file that is specified with `SAVeset:ONEFile:NAME`.

**SAVeset:ONEFile:OPEN**

Loads the One File that is specified with `SAVeset:ONEFile:NAME`.

**Usage:** Event

**Manual operation:** See "Load from file" on page 478

**SAVeset:ONEFile:USERpref** <Include>

If ON, the user preferences settings are also restored together with the saveset and reference waveform. User preferences settings include the user-specific display settings like diagram layout, toolbar, intensity and transparency settings.

**Parameters:**

<Include> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Include user preferences" on page 478

### 24.16.3 Waveform data transmission

The R&S RTP provides specific data export commands for the various waveform types. The commands transmit the data of the waveform points from the instrument to the controlling computer. The data can be used in MATLAB, for example.

The commands are described in the relevant chapters:

- Analog waveforms: Chapter 24.8.6, "Waveform data", on page 1252
- Reference waveforms: Chapter 24.10.2.3, "Waveform data export", on page 1395
- Math waveforms: Chapter 24.10.3, "Mathematics", on page 1400
- Spectrum waveforms: Chapter 24.13.2, "Waveform data", on page 1538
- Logic channels: Chapter 24.18.5, "MSO data", on page 2332
- I/Q data: Chapter 24.21.2, "I/Q data output", on page 2372

### 24.16.4 Waveform data export to file

The resulting files of waveforms exports are described in Chapter 12.2.1, "Waveform export files", on page 483.

EXPort:WAVeform:SOURce ... 1614
EXPort:WAVeform:MULTichannel ... 1615
CHANnel<m>:EXPortstate ... 1616
EXPort:WAVeform:NAME ... 1616
EXPort:WAVeform:SAVE ... 1617
EXPort:WAVeform:SCOPe ... 1617
EXPort:WAVeform:STARt ... 1617
EXPort:WAVeform:STOP ... 1618
EXPort:WAVeform:ZOOM ... 1618
EXPort:WAVeform:CURSorset ... 1618
EXPort:WAVeform:MEAS ... 1619
EXPort:WAVeform:DLOGging ... 1619
EXPort:WAVeform:TIMestamps ... 1620
EXPort:WAVeform:INCXvalues ... 1620
EXPort:WAVeform:RAW ... 1621
EXPort:WAVeform:DISPlayoff ... 1621
EXPort:WAVeform:FASTexport ... 1621
EXPort:WAVeform:ICDR ... 1622

**EXPort:WAVeform:SOURce** <Source>

Selects the waveform to be exported to file.

The commands takes effect if `EXPort:WAVeform:MULTichannel` is `OFF`.

**Parameters:**

<Source> | C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | MSOB1 | MSOB2 | MSOB3 | MSOB4 | TRK1 | TRK2 | TRK3 | TRK4 | TRK5 | TRK6 | TRK7 | TRK8 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | BATHtub1 | BMEas1 | SRESponse1 | SRHist1 | SDHist1 | TJHist1 | DJHist1 | DDJHist1 | PJHist1 | RJOHist1 | SRTRack1 | SDTRack1 | TJTRack1 | DJTRack1 | DDJTrack1 | PJTRack1 | RJOTrack1 | TJSPectrum1 | DDJSpectrum1 | PJSPectrum1 | RJOSpectrum1 | SEYE1 | BATHtub2 | BMEas2 | SRESponse2 | SRHist2 | SDHist2 | TJHist2 | DJHist2 | DDJHist2 | PJHist2 | RJOHist2 | SRTRack2 | SDTRack2 | TJTRack2 | DJTRack2 | DDJTrack2 | PJTRack2 | RJOTrack2 | TJSPectrum2 | DDJSpectrum2 | PJSPectrum2 | RJOSpectrum2 | SEYE2 | NBAThtub1 | NBMeas1 | TNHist1 | DNHist1 | DDNHist1 | PNHist1 | RNOHist1 | TNTRack1 | DNTRack1 | DDNTrack1 | PNTRack1 | RNOTrack1 | TNSPectrum1 | DDNSpectrum1 | PNSPectrum1 | RNOSpectrum1 | NBAThtub2 | NBMeas2 | TNHist2 | DNHist2 | DDNHist2 | PNHist2 | RNOHist2 | TNTRack2 | DNTRack2 | DDNTrack2 | PNTRack2 | RNOTrack2 | TNSPectrum2 | DDNSpectrum2 | PNSPectrum2 | RNOSpectrum2 | PJDHist1 | BUJHist1 | RJHist1 | PJDTrack1 | PJDSpectrum1 | PJDHist2 | BUJHist2 | RJHist2 | PJDTrack2 | PJDSpectrum2 | PNDHist1 | BUNHist1 | RNHist1 | PNDTrack1 | PNDSpectrum1 | PNDHist2 | BUNHist2 | RNHist2 | PNDTrack2 | PNDSpectrum2

*RST: C1W1

**Example:** See Chapter 24.3.5.2, "Exporting waveform data to file", on page 1180

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 492

---

**EXPort:WAVeform:MULTichannel** <MltChXpt>

Enables or disables the multichannel export.

If you enable the multichannel export, all active channels are included to the export data. You can change the export state using the `CHANnel<m>:EXPortstate` command.

If multichannel export is disabled, select the waveform to be exported using the `EXPort:WAVeform:SOURce` command.

Note that `CHANnel<m>[:WAVeform<n>]:DATA[:VALues]?` returns the data of all channels that are selected for export, no matter of the channel suffix.

**Parameters:**
<MltChXpt> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Multi channel" on page 491

---

**CHANnel<m>:EXPortstate** <ExportState>

Includes or excludes the indicated channel in waveform export. The data of channel waveform 1 is exported.

The commands takes effect if `EXPort:WAVeform:MULTichannel` is `ON`.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<ExportState> ON | OFF
If you enable the multichannel export, the export state of all active channels is automatically set ON.
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Selected sources" on page 492

---

**EXPort:WAVeform:NAME** <FileName>

Sets the file name, file format and path to save the waveform to.

See also: Chapter 12.2.1, "Waveform export files", on page 483

**Parameters:**
<FileName> String with path and file name with extension .xml, .bin, or .csv

**Example:**
```
EXPort:WAVeform:NAME 'C:\temp\Export_Ch11.xml'
EXPort:WAVeform:SAVE
```
Saves the waveform data in XML format to `C:\temp\Export_Ch1.xml`.

**Example:**
```
EXPort:WAVeform:NAME 'C:\temp\Export_Ch2.bin'
EXPort:WAVeform:SAVE
```
Saves the waveform data in binary format to `C:\temp\Export_Ch2.bin`.

**Usage:** Asynchronous command

**Manual operation:** See "Save to file" on page 494

**EXPort:WAVeform:SAVE**

Saves the waveform(s) to the file specified with `EXPort:WAVeform:NAME`. The file format is also set using the `...NAME` command.

**Example:** See Chapter 24.3.5.2, "Exporting waveform data to file", on page 1180

**Usage:** Event

**Manual operation:** See "Save to file" on page 494

**EXPort:WAVeform:SCOPe** <Scope>

Defines the part of the waveform record that has to be stored.

**Parameters:**

<Scope> WFM | ZOOM | CURSor | GATE | MANual

**WFM**
Complete waveform

**ZOOM**
Data included in the zoom area if a zoom is defined for the source waveform.

**CURSor**
Data between the cursor lines if a cursor measurement is defined for the source waveform.

**GATE**
data included in the measurement gate if a gated measurement is defined for the source waveform.

**MANual**
Saves the data between user-defined start and stop values to be set with `EXPort:WAVeform:STARt` and `EXPort:WAVeform:STOP`.

*RST: WFM

**Example:** See Chapter 24.3.5.2, "Exporting waveform data to file", on page 1180

**Usage:** Asynchronous command

**Manual operation:** See "Scope" on page 492

**EXPort:WAVeform:STARt** <Start>

Sets the start value of the waveform section for export, if `EXPort:WAVeform:SCOPe` is set to `Manual`.

**Parameters:**

<Start>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Scope" on page 492

---

**EXPort:WAVeform:STOP** <Stop>

Sets the end value of the waveform section for export, if `EXPort:WAVeform:SCOPe` is set to `Manual`.

**Parameters:**

<Stop>
Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Scope" on page 492

---

**EXPort:WAVeform:ZOOM** <ZoomDiagram>

Sets the zoom area to be used for limited data export if `EXPort:WAVeform:SCOPe` is set to `ZOOM`.

**Parameters:**

<ZoomDiagram> Name of the diagram on which the zoom area is based.

**Example:** See "Exporting interleaved x/y data of a zoom to CSV file" on page 1182

**Usage:** Asynchronous command

**Manual operation:** See "Scope" on page 492

---

**EXPort:WAVeform:CURSorset** <Cursorset>

Sets the cursor set to be used for limited data export if `EXPort:WAVeform:SCOPe` is set to `CURSor`.

**Parameters:**

<Cursorset>
CURSOR1 | CURSor1 | CURSOR2 | CURSor2 | CURSOR3 | CURSor3 | CURSOR4 | CURSor4

CURSOR1 = CURSor1, CURSOR2 = CURSor2, CURSOR3 = CURSor3, CURSOR4 = CURSor4

**Usage:** Asynchronous command

**Manual operation:** See "Scope" on page 492

**EXPort:WAVeform:MEAS** <MeasGate>

Sets the gate to be used for limited data export if `EXPort:WAVeform:SCOPe` is set to `GATE`.

**Parameters:**

<MeasGate> MEAS1 | MEAS2 | MEAS3 | MEAS4 | MEAS5 | MEAS6 | MEAS7 | MEAS8

Measurement for which the gate is defined.

**Example:** See "Exporting raw data of a measurement gate to BIN file" on page 1181

**Usage:** Asynchronous command

**Manual operation:** See "Scope" on page 492

**EXPort:WAVeform:DLOGging** <DataLogging>

The command enables the export of subsequent acquisitions of the selected waveforms. The waveforms are taken from a running Nx Single acquisition (data logging, history is disabled), or from the history (multiple waveforms, history is enabled).

If the history is disabled (`CHANnel<m>[:WAVeform<n>]:HISTory[:STATe]`) and data logging is enabled, a specified number of waveforms is transferred to file directly during RUN Nx SINGLE acquisition. Enabling data logging stops a running acquisition. Set the number of acquisitions to be acquired and stored with `ACQuire:COUNt` and start export using `RUNSingle`.

If the history is enabled, the subsequent waveforms are taken from the history. Specify the range with `CHANnel<m>[:WAVeform<n>]:HISTory:STARt` and `CHANnel<m>[:WAVeform<n>]:HISTory:STOP`. Then play the history with `CHANnel<m>[:WAVeform<n>]:HISTory:PLAY`.

The commands `EXPort:WAVeform:SAVE`, `CHANnel<m>[:WAVeform<n>]:DATA[:VALues]?` and `CHANnel<m>[:WAVeform<n>]:DATA:HEADer?` are not available if data logging is enabled. The `RUNContinous` command disables data logging.

If data logging is off, and the history is enabled, one waveform out of the history is written to file. Specify the waveform using `CHANnel<m>[:WAVeform<n>]:HISTory:CURRent` and save it using `EXPort:WAVeform:SAVE`.

**Parameters:**

<DataLogging> ON | OFF

*RST: OFF

**Example:** See:
"Exporting multiple running acquisitions of a single waveform to XML file" on page 1183
"Exporting multiple acquisition of the history to XML file" on page 1184
"Exporting a single acquisition of the history to BIN file" on page 1183

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 496

---

**EXPort:WAVeform:TIMestamps** <UseTimestamps>

Exports the relative timestamps of all history waveforms to the waveform data file. The time is written at the beginning of each waveform record.

**Parameters:**
<UseTimestamps> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Time stamps" on page 495

---

**EXPort:WAVeform:INCXvalues** <IncHorValues>

Includes horizontal values in the retrieved data (time or frequency values, depending on the waveform). X and Y-values are written alternately to the file. If disabled, only Y-values - mostly voltage values - are written. The X-values are always returned in 64 bit real format, regardless of the defined data format.

The setting is not available for the export of raw data.

The command affects the content of export files as well as data retrieved with:

- `CHANnel<m>[:WAVeform<n>]:DATA[:VALues]?`
- `CALCulate:MATH<m>:DATA[:VALues]?`
- `REFCurve<m>:DATA[:VALues]?`
- `DIGital<m>:DATA[:VALues]?`
- `BUS<m>:PARallel:DATA[:VALues]?`
- `ZVC:Z<m>:I<n>:DATA[:VALues]?`
- `ZVC:Z<m>:V<n>:DATA[:VALues]?`

**Parameters:**
<IncHorValues> ON | OFF
*RST: OFF

**Example:** See:
"Exporting interleaved x/y data of a single waveform to CSV file" on page 1182
"Exporting interleaved x/y data of a zoom to CSV file" on page 1182

**Usage:** Asynchronous command

**Manual operation:** See "Interleaved x/y" on page 494

**EXPort:WAVeform:RAW** <RawValues>

Enables the export of raw sample data, and sets the data format to integer 8 bit. In high definition acquisition mode, the data format is integer 16 bit. Fot INT16, you can set the byte order using the `FORMat:BORDer` command.

The raw format reduces the file size but changes also the precision of the values.

Currently, the setting is not available for the export of digital channel data and data of R&S RT-ZVC channels.

The raw export of interleaved X/Y values is not supported.

**Parameters:**

<RawValues> ON | OFF

*RST: OFF

**Example:** See:
"Exporting raw data of a single waveform to BIN file" on page 1181
"Exporting raw data of a measurement gate to BIN file" on page 1181

**Usage:** Asynchronous command

**Manual operation:** See "Raw (ADC direct)" on page 493

**EXPort:WAVeform:DISPlayoff** <FastExport>

Enables or disables the display update during an Nx Single acquisition.

**Parameters:**

<FastExport> ON | OFF

ON: Disables the display update for maximum export speed.
OFF: Enables the display update. The export is slower.

*RST: OFF

**Usage:** Asynchronous command

**EXPort:WAVeform:FASTexport** <Enable>

To improve the performance of data export to file, the measurements are performed slower while the data export speeds up.

**Setting parameters:**

<Enable> ON | OFF

**Example:** See Chapter 24.3.5.2, "Exporting waveform data to file", on page 1180

**Usage:** Setting only
Asynchronous command

**EXPort:WAVeform:ICDR** <IncludeCDRInformation>

Includes the CDR information in the waveform file.

**Parameters:**
<IncludeCDRInformation> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Include CDR info" on page 491

### 24.16.5 Waveform histogram export to file

EXPort:HISTogram:SELect....1622
EXPort:HISTogram:INCidence....1622
EXPort:HISTogram:NAME....1622
EXPort:HISTogram:SAVE....1623
EXPort:HISTogram:DATA?....1623

**EXPort:HISTogram:SELect** <Name>

Selects the histogram to be exported.

**Parameters:**
<Name> String with the histogram name.

**Example:** See "Exporting histogram data to file" on page 1177

**Usage:** Asynchronous command

**Manual operation:** See "Select" on page 498

**EXPort:HISTogram:INCidence** <Incidence>

Sets the mode of exported data: relative or absolute frequency of amplitude values.

**Parameters:**
<Incidence> ABS | REL
*RST: REL

**Example:** See "Exporting histogram data to file" on page 1177

**Usage:** Asynchronous command

**Manual operation:** See "Incidence" on page 498

**EXPort:HISTogram:NAME** <Path>

Sets the file name and path to save the histogram to.

**Parameters:**
<Path> String with path and file name. The file extension defines the file format: XML, CSV, or BIN.

**Example:** See "Exporting histogram data to file" on page 1177

**Usage:** Asynchronous command

**Manual operation:** See "Save to file" on page 498

---

**EXPort:HISTogram:SAVE**

Saves the histogram to the file specified with `EXPort:HISTogram:NAME`.

**Example:** See "Exporting histogram data to file" on page 1177

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save to file" on page 498

---

**EXPort:HISTogram:DATA?**

Transfers the histogram data to the controlling computer. The data can be used in MATLAB, for example.

To set the export data format, use `FORMat[:DATA]`.

**Return values:**
<Data> List of values according to the format settings and `EXPort:HISTogram:INCidence`.

**Example:** See "Transferring histogram data" on page 1177.

**Usage:** Query only

**Manual operation:** See "Save to file" on page 498

## 24.16.6 Eye

EXPort:EYE:DATA? ..... 1623
EXPort:EYE:NAME ..... 1624
EXPort:EYE:SAVE ..... 1624
EXPort:EYE:SELect ..... 1624
EXPort:EYE:SOURce? ..... 1624

---

**EXPort:EYE:DATA?**

Transfers the eye data to the controlling computer. The data can be used in MATLAB, for example.

To set the export data format, use `FORMat[:DATA]`.

**Usage:** Query only

**Manual operation:** See "Save to file" on page 504

**EXPort:EYE:NAME** <Path>

Sets the path, the file name, and the file format for the eye file. Available file formats are *csv and *bin.

**Parameters:**
<Path>

**Usage:** Asynchronous command

**Manual operation:** See "Save to file" on page 504

**EXPort:EYE:SAVE**

Saves the selected eye to the file that is specified with `EXPort:EYE:NAME`.

**Usage:** Event

**Manual operation:** See "Save to file" on page 504

**EXPort:EYE:SELect** <SelectableEye>

Selects the eye that you want to save to a file.

**Parameters:**
<SelectableEye> EYE1 | EYE2 | EYE3 | EYE4
*RST: EYE1

**Usage:** Asynchronous command

**Manual operation:** See "Select/ Source" on page 504

**EXPort:EYE:SOURce?**

Returns the source of the eye measurement, selected with `EXPort:EYE:SELect`.

**Return values:**
<Source> C1W1 | C1W2 | C1W3 | C1W4 | C2W1 | C2W2 | C2W3 | C2W4 | C3W1 | C3W2 | C3W3 | C3W4 | C4W1 | C4W2 | C4W3 | C4W4
*RST: NONE

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Select/ Source" on page 504

### 24.16.7 Long term measurement results and measurement histogram export to file

EXPort:MEASurement:SELect....1625
EXPort:MEASurement:TYPE....1625
EXPort:MEASurement:NAME....1625
EXPort:MEASurement:SAVE....1626
EXPort:MEASurement:DATA?....1626

**EXPort:MEASurement:SELect** <SelcMeas>

Selects the measurement for export of long term or measurement histogram data.

**Parameters:**

<SelcMeas> MEAS1 | MEAS2 | MEAS3 | MEAS4 | MEAS5 | MEAS6 | MEAS7 | MEAS8

*RST: MEAS1

**Example:** See "Exporting long-term measurement data to file" on page 1178

**Usage:** Asynchronous command

**Manual operation:** See "Measurement" on page 502

**EXPort:MEASurement:TYPE** <ExportType>

You can export the result data of the long term measurement, or the measurement histogram, or the track data.

To export the measurement histogram, it must be enabled using `MEASurement<m>:STATistics:HISTogram`.

To export the long term results, the long term measurement must be enabled using `MEASurement<m>:LTMeas[:STATe]`.

To export a track, the track must be enabled before. Track measurements require an option, see "Enable (Track)" on page 390.

**Parameters:**

<ExportType> LONGterm | HISTogram | TRACk

LONGTERM = LONGterm, HISTOGRAM = HISTogram

*RST: HISTogram

**Example:** See "Exporting long-term measurement data to file" on page 1178

**Usage:** Asynchronous command

**Manual operation:** See "Export type" on page 502

**EXPort:MEASurement:NAME** <Path>

Sets the file name and path to save the long term or measurement histogram data to.

**Parameters:**

<Path> String with path and file name. The file extension defines the file format: XML, CSV, or BIN.

**Example:** See "Exporting long-term measurement data to file" on page 1178

**Usage:** Asynchronous command

---

**EXPort:MEASurement:SAVE**

Saves the long term or measurement histogram results to the file specified using `EXPort:MEASurement:NAME`.

The measurement data can be exported as absolute or relative values, which is defined using `EXPort:HISTogram:INCidence`.

**Example:** See "Exporting long-term measurement data to file" on page 1178

**Usage:** Event

**Manual operation:** See "Save to file" on page 503

---

**EXPort:MEASurement:DATA?**

Transfers the long term measurement data to the controlling computer. The data can be used in MATLAB, for example. The transferred data are the same as the data written to files.

To set the export data format, use `FORMat[:DATA]`.

**Return values:**

<Data> List of values according to the format settings
Long term data:
If statistics are disabled, the current value of each long term point is returned.
If statistics are enabled (`MEASurement<m>:STATistics[:ENABle]`), seven values for each long term point are returned: upper peak, lower peak, RMS, standard deviation, average, number of measured results per long term point, number of waveforms per long term point.
For measurement histograms, absolute values are returned.
See also: Chapter 12.2.5, "Result plots analysis", on page 500.

**Example:** See "Transferring long-term measurement data" on page 1178

**Usage:** Query only

**Manual operation:** See "Save to file" on page 503

### 24.16.8 Screenshots

The HCOPy subsystem and some other commands control the output of display information for documentation purposes. The instrument allows two independent output configurations which can be set separately with the suffix.

Note that the remote mode is intended for maximum performance. Therefore, the display does not follow the remote commmands consistantly. To get a correct screenshot, turn the display on using `SYST:DISP:UPD ON`.

HCOPy:DESTination<1..2> ..... 1627
MMEMory:NAME ..... 1627
HCOPy:DEVice<m>:LANGuage ..... 1628
HCOPy:DEVice<m>:INVerse ..... 1628
HCOPy:WBKG ..... 1629
HCOPy:CMAP<m>:DEFault ..... 1629
HCOPy:SSD ..... 1629
HCOPy:ISBA ..... 1630
HCOPy:IMMediate<m>[:DUM] ..... 1630
HCOPy:IMMediate<m>:NEXT ..... 1630

**HCOPy:DESTination<1..2>** <Medium>

Selects the output medium: file, printer or clipboard.

**Suffix:**
1..2 Selects the output configuration.

**Parameters:**
<Medium> MMEM | SYST:COMM:CLIP

String parameter

**MMEM**
Directs the display image to a file. The `MMEMory:NAME` command defines the file name. The file format is defined with `HCOPy:DEVice<m>:LANGuage`.

**SYST:COMM:CLIP**
Directs the hardcopy to the clipboard.

*RST: SYST:COMM:CLIP

**Example:** `HCOP:DEST 'MMEM'`
See also Chapter 24.3.5.1, "Saving a screenshot to file", on page 1180

**Usage:** Asynchronous command

**Manual operation:** See "Save to file" on page 510

**MMEMory:NAME** <FileName>

Defines the file name when an image of the display is stored to a file.

**Setting parameters:**

<FileName> String parameter specifiying path and file name of the screenshot.

**Example:** See Chapter 24.3.5.1, "Saving a screenshot to file", on page 1180

**Usage:** Setting only
SCPI confirmed
Asynchronous command

**Manual operation:** See "Save to file" on page 510

---

**HCOPy:DEVice<m>:LANGuage** <FileFormat>

Defines the file format for output of the display image to file.

To set the output to file, use `HCOPy:DESTination<1..2>` with parameter `'MMEM'`.

**Suffix:**

<m> 1..2
Selects the output configuration.

**Parameters:**

<FileFormat> PNG | JPG | BMP | TIFF | PDF
*RST: PNG

**Example:** See Chapter 24.3.5.1, "Saving a screenshot to file", on page 1180

**Usage:** Asynchronous command

**Manual operation:** See "Save to file" on page 510

---

**HCOPy:DEVice<m>:INVerse** <InverseColor>

Inverts the colors of the output, i.e. a dark waveform is printed on a white background. Screenshots are inverted per default when you save using remote commands. Turn inversion off if you want the screen display in the screenshot.

See also:

- `HCOPy:WBKG` on page 1629
- "White background" on page 509

**Suffix:**

<m> 1..2
Selects the output configuration.

**Parameters:**

<InverseColor> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Inverse color" on page 510

**HCOPy:WBKG** <WhiteBackground>

Inverts the background color. So you can print waveforms with normal waveform colors on white background.

If both `HCOPy:WBKG` and `HCOPy:DEVice<m>:INVerse` are ON, the instrument inverts the background twice, and it appears black.

See also: "White background" on page 509.

**Parameters:**

<WhiteBackground> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "White background" on page 509

**HCOPy:CMAP<m>:DEFault** <PrintColorSet>

Defines the default color set for printing of the display image.

To set the output to printer, use `HCOPy:DESTination<1..2>` with parameter `'SYST:COMM:PRIN'`.

**Suffix:**

<m> 1..2
Selects the output configuration.

**Parameters:**

<PrintColorSet> DEF1 | DEF4

**DEF1**
Current screen colors with white background and black grid.

**DEF4**
Current screen colors without any changes (black background).

*RST: DEF1

**Usage:** Asynchronous command

**HCOPy:SSD** <ShowSetupDialog>

Enables or disables the display of open dialog boxes in screenshots. Use this command if you want to document settings in screenshots.

**Parameters:**

<ShowSetupDialog> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Firmware/software:** FW 3.20

**Manual operation:** See "Show setup dialog" on page 510

**HCOPy:ISBA** IncludeSignalBar

If the value is set to `OFF`, the screenshot shows only the diagram area, without the signal bar.

**Parameters:**

IncludeSignalBar ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show only grid" on page 510

**HCOPy:IMMediate<m>[:DUM]**

Starts the immediate output of the display image, depending on the `HCOPy:DESTination<1..2>` destination setting.

By default, screenshots are inverted when saving them with remote commands. If you want the screen display in the screenshot, turn inversion off (`HCOPy:DEVice<m>:INVerse`).

To get a correct screenshot of the diagrams, results, and dialog boxes, turn on the display using `SYST:DISP:UPD ON`.

**Suffix:**

<m> 1..2
Selects the output configuration.

**Example:**

```
SYST:DISP:UPD ON
HCOP:DEST 'MMEM'
MMEM:NAME 'C:\Temp\Print.bmp'
HCOP:IMMediate; *OPC?
```

**Example:** See Chapter 24.3.5.1, "Saving a screenshot to file", on page 1180

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save to file" on page 510

**HCOPy:IMMediate<m>:NEXT**

Starts the output of the next display image, depending on the `HCOPy:DESTination<1..2>` destination setting.

If the screenshot is saved to a file, the file name used in the last saving process is automatically counted up to the next unused name.

**Suffix:**

<m> 1..2
Selects the output configuration.

**Example:** See Chapter 24.3.5.1, "Saving a screenshot to file", on page 1180

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save to file" on page 510

## 24.16.9 Reports

The following commands configure and save report files. To configure the screenshot that is included in the report, use the commands explained in Chapter 24.16.8, "Screenshots", on page 1627.

REPort:LANGuage....1631
REPort:PAPersize....1631
REPort:LOGType....1631
REPort:LOGO....1632
REPort:USER....1632
REPort:COMMent....1632
REPort:FILE:NAME....1632
REPort:FILE:SAVE....1633

---

**REPort:LANGuage** <Language>

Sets the language to be used in the report. Available languages are listed in the data sheet.

**Parameters:**
<Language> String with the english language name, upper case.

**Example:** `REPort:LANGuage 'Spanish'`

**Manual operation:** See "Language" on page 512

---

**REPort:PAPersize** <PaperSize>

Selects the paper size: A4 or US Letter.

**Parameters:**
<PaperSize> A4 | USL
*RST: A4

**Usage:** Asynchronous command

**Manual operation:** See "Paper size" on page 513

---

**REPort:LOGType** <Logo>

By default, the Rohde & Schwarz logo is shown in the header of the report pages. You can switch the logo off, or select your logo to be shown.

**Parameters:**

<Logo> RS | CUST | NONE

**CUST**
Select the logo file using `REPort:LOGO`.

*RST: RS

**Usage:** Asynchronous command

**Manual operation:** See "Logo" on page 513

---

**REPort:LOGO** <LogoFile>

Defines the logo file that is used on the report if `REPort:LOGType` is set to `CUSTom`.

**Parameters:**

<LogoFile> String with the path and filename of the logo image.

**Example:**
```
REPort:LOGO 'C:\Company files\logo.jpg'
```

**Usage:** Asynchronous command

**Manual operation:** See "Logo" on page 513

---

**REPort:USER** <User>

Enter the user name that appears in the general information section at the beginning of the report.

**Parameters:**

<User> String parameter

**Usage:** Asynchronous command

**Manual operation:** See "User name / Comment" on page 513

---

**REPort:COMMent** <Comment>

Enter a comment that appears in the general information section at the beginning of the report.

**Parameters:**

<Comment> String parameter

**Usage:** Asynchronous command

**Manual operation:** See "User name / Comment" on page 513

---

**REPort:FILE:NAME** <ReportFile>

Sets the file name and path to save the report to.

**Parameters:**

<ReportFile> String with path and file name. The file extension defines the file format: PDF, HTML, or DOC.

**Usage:** Asynchronous command

**Manual operation:** See "Save to file" on page 513

---

**REPort:FILE:SAVE**

Saves the report to the specified file.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save to file" on page 513

### 24.16.10 Autonaming

MMEMory:AUTonaming:PREFix ..... 1633
MMEMory:AUTonaming:USERtext ..... 1633
MMEMory:AUTonaming:DATE ..... 1633
MMEMory:AUTonaming:INDex ..... 1633
MMEMory:AUTonaming:TIME ..... 1633
MMEMory:AUTonaming:TEXT ..... 1633
MMEMory:AUTonaming:DEFaultpath ..... 1634
MMEMory:AUTonaming:RESPath ..... 1634
MMEMory:AUTonaming:RESall ..... 1634

---

**MMEMory:AUTonaming:PREFix** <State>
**MMEMory:AUTonaming:USERtext** <State>
**MMEMory:AUTonaming:DATE** <State>
**MMEMory:AUTonaming:INDex** <State>
**MMEMory:AUTonaming:TIME** <State>

Includes or excludes the name part in the file name pattern for automatic file name generation. This name is used as the default file name.

The prefix indicates the type of data that is saved, for example, Histogram, RefCurve, Settings.

To define a user text, use `MMEMory:AUTonaming:TEXT`.

**Parameters:**
<State> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Time" on page 109

---

**MMEMory:AUTonaming:TEXT** <NameString>

Defines a text, that can be included in the autonaming pattern.

**Parameters:**
<NameString> String parameter

**Usage:** Asynchronous command

**Manual operation:** See "Text input" on page 108

---

**MMEMory:AUTonaming:DEFaultpath** <Path>

Sets the path where data and settings files will be stored. The factory default path is:

- `"C:\Users\Public\Documents\Rohde-Schwarz\RTx"` if no USB flash drive is connected
- Drive letter of the USB flash drive, for example, `"E:\"` or `"F:\"` if a USB flash drive is connected.

**Parameters:**
<Path> String parameter

**Usage:** Asynchronous command

**Manual operation:** See "Default path for all file operations" on page 109

---

**MMEMory:AUTonaming:RESPath**

Resets the path for file operations to the factory default path.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Reset" on page 109

---

**MMEMory:AUTonaming:RESAll**

Resets all autonaming settings to the default value, including the path.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Reset" on page 109

### 24.16.11 CSV export

EXPort:RESult:DECSymbol ..... 1634
EXPort:RESult:DELimiter ..... 1635

---

**EXPort:RESult:DECSymbol** <DecimalSymbol>

Sets if point or comma is used as a decimal symbol.

**Parameters:**
<DecimalSymbol> POINt | COMMa
*RST: POINt

**Usage:** Asynchronous command

**Manual operation:** See "Decimal symbol" on page 110

---

**EXPort:RESult:DELimiter** <CsvDelimiter>

Selects the list separator symbol from a list.

**Parameters:**
<CsvDelimiter> SEMICOLON | COMMA | SPACE | TAB | COLON
*RST: COMMA

**Usage:** Asynchronous command

**Manual operation:** See "Csv delimiter" on page 110

## 24.16.12 External application

The commands configure an external application that can be started at various events, for example, trigger event, or mask test violation.

EXECutable:NAME ..... 1635
EXECutable:PARameter ..... 1635
EXECutable:WDIRectory ..... 1636

---

**EXECutable:NAME** <ApplicationPath>

Sets the path to the application executable.

**Parameters:**
<ApplicationPath> String parameter containing path, filename, and file extension

**Example:**
```
EXECutable:NAME 'C:\Program
Files\Wireshark\Wireshark.exe'
TRIGger:EVENt:RUNexec TRIGger
```

**Usage:** Asynchronous command

---

**EXECutable:PARameter** <AppParameters>

Sets optional parameters for the external executable.

**Parameters:**
<AppParameters> String parameter

**Usage:** Asynchronous command

**EXECutable:WDIRectory** <WorkDirectory>

Sets the working directory for the executable.

**Parameters:**
<WorkDirectory> String parameter

**Usage:** Asynchronous command

## 24.17 Protocols


- Configuration settings for all serial protocols ..... 1636
- Trigger Settings for all serial protocols ..... 1641
- I²C (option R&S RTP-K1) ..... 1643
- SPI (option R&S RTP-K1) ..... 1677
- UART/RS-232/RS-422/RS-485 (option R&S RTP-K2) ..... 1697
- CAN (option R&S RTP-K3/K9) ..... 1708
- LIN (option R&S RTP-K3) ..... 1752
- MIL-1553 (option R&S RTP-K6) ..... 1783
- ARINC 429 (option R&S RTP-K7) ..... 1810
- Ethernet 10BASE-T and 100BASE-TX (option R&S RTP-K8) ..... 1827
- Ethernet 100BASE-T1 (option R&S RTP-57) ..... 1852
- Ethernet 1000BASE-T1 (option R&S RTP-K58) ..... 1883
- RFFE (option R&S RTP-K40) ..... 1916
- D-PHY (option R&S RTP-K42) ..... 1949
- M-PHY (option R&S RTP-K44) ..... 1982
- Custom: manchester / NRZ (option R&S RTP-K50) ..... 2043
- 8B/10B (option R&S RTP-K52) ..... 2088
- MDIO (option R&S RTP-K55) ..... 2104
- USB (option R&S RTP-K60) ..... 2120
- USB 3.1 (option R&S RTP-K61/K62) ..... 2162
- USBPD (option R&S RTP-K63) ..... 2200
- Space wire (option R&S RTP-K65) ..... 2225
- PCIe (option R&S RTPK72/K73) ..... 2244
- DDR (option R&S RTP-K91/K93) ..... 2293


### 24.17.1 Configuration settings for all serial protocols


BUS<m>:TYPE ..... 1637
BUS<m>[:STATe] ..... 1637
BUS<m>:SETReflevels ..... 1638
BUS<m>:FAUToset ..... 1638
BUS<m>:RESult ..... 1638
BUS<m>:THReshold ..... 1638
BUS<m>:FORMat ..... 1639
BUS<m>:NEWList ..... 1639
BUS<m>:SYMBols ..... 1639

BUS<m>:ZCOupling....1640
DISPlay:RESultboxes:DEPosition....1640
BUS<m>:EXPResult:DETail....1640
BUS<m>:EXPResult:SAVE....1640
BUS<m>:EXPResult:TIME....1641

**BUS<m>:TYPE** <Type>

Defines the bus or protocol type for analysis. The type of available buses depends on the installed options.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Type> I2C | SPI | UART | CAN | CANFd | LIN | MILS1553 | MILStd1553 | ARIN429 | ARINc429 | SWIRe | MDIO | HBTO | USB | USBPD | ETHernet | CMSB | RFFE | DPHY | MPHY | EBTB | USB3 | PCIE | TBTO | DDR

MILS1553 = MILStd1553: specification MIL-STD-1553
ARIN429 = ARINc429: specification ARINC 429
HBTO: Ethernet 100BASE-T1
TBTO: Ethernet 1000BASE-T1
CMSB: custom decode serial bus
EBTB: 8b/10b general decoding
SWIRe: SpaceWire

*RST: I2C

**Usage:** Asynchronous command

**Manual operation:** See "Protocol" on page 517

**BUS<m>[:STATe]** <State>

Enables the decoding of the specified bus.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<State> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Decode" on page 517

**BUS<m>:SETReflevels**

Executes the measurement of reference levels and sets the thresholds to the middle reference level of the measured amplitude.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Threshold" on page 528

**BUS<m>:FAUToset**

Starts software algorithms for determining the signal threshold levels and bitrate.

**Suffix:**
<m> 1..4

**Usage:** Event
Asynchronous command

**Manual operation:** See "Threshold" on page 528

**BUS<m>:RESult** <ShowResultTable>

Opens a table with decoded data of the serial signal. The function affects all protocol types and requires the option for the analyzed protocol.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ShowResultTable> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Show table" on page 519

**BUS<m>:THReshold** <ShwThresLines>

If ON, the threshold levels are displayed in the diagram.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ShwThresLines> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show thresholds" on page 528

---

**BUS<m>:FORMat** <DataFormat>

Sets the number format for decoded data values of the indicated serial bus. It defines the format in the decode table, and in the combs of the decoded signal on the screen.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<DataFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG
ASCII = ASCii
USIG = unsigned
*RST: HEX

**Manual operation:** See "Data format" on page 519

---

**BUS<m>:NEWList** <FileName>

Loads a label list file.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Setting parameters:**
<FileName> String parameter with path and file name.

**Example:**
```
BUS1:NEWList 'C:\Protocols\CAN.csv'
BUS1:SYMBols ON
```

**Usage:** Setting only

**Manual operation:** See "Load symbols" on page 519

---

**BUS<m>:SYMBols** <UseTranslation>

Activates the lable list to be used for decoding.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<UseTranslation> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Show symbols" on page 519

**BUS<m>:ZCOupling** <ZoomCoupling>

If enabled, the protocol decode zoom and result table are synchronized.

**Suffix:**
<m> 1..4

**Parameters:**
<ZoomCoupling> ON | OFF
*RST: OFF

**Manual operation:** See "Zoom coupling" on page 519

**DISPlay:RESultboxes:DEPosition** <DecodePosition>

Defines the position of the protocol decode result table on the screen.

**Parameters:**
<DecodePosition> PREV | FLOA | DOCK

**PREV**
Preview: result icon on the sidebar.

**FLOA**
Floating result box in front of the diagrams.

**DOCK**
Docked: fixed tab below the diagrams.

*RST: DOCK

**Usage:** Asynchronous command

**Manual operation:** See "Table position" on page 93

**BUS<m>:EXPResult:DETail** <XptResDetails>

Includes the detailed results for all frames for the exported protocol results.

**Suffix:**
<m> 1..4

**Parameters:**
<XptResDetails> ON | OFF
*RST: ON

**Manual operation:** See "Include details" on page 523

**BUS<m>:EXPResult:SAVE** <FileName>

Saves the selected results to the indicated file.

**Suffix:**
<m> 1..4

**Setting parameters:**
<FileName>

**Usage:** Setting only

**Manual operation:** See "Export to file" on page 523

---

**BUS<m>:EXPResult:TIME** <XptResTim>

Includes the frame timing for the exported protocol results.

**Suffix:**
<m> 1..4

**Parameters:**
<XptResTim> ON | OFF
*RST: ON

**Manual operation:** See "Include frame timing" on page 523

## 24.17.2 Trigger Settings for all serial protocols

The following commands are available for all serial protocols that have a protocol trigger.

TRIGger<m>:SOURce[:SELect] ..... 1641
TRIGger<m>:SOURce:SBSelect ..... 1642
BUS<m>:TYPE ..... 1642

---

**TRIGger<m>:SOURce[:SELect]** <SourceDetailed>

Selects the source of the trigger signal.

**Suffix:**
<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger
Available values depend on the selected trigger source. For input channels CHAN1...4, a trigger sequence can be configured.
For all other trigger sources, only suffix 1 is allowed.

**Parameters:**
<SourceDetailed> CHAN1 | CHANnel1 | CHAN2 | CHANnel2 | CHAN3 | CHANnel3 | CHAN4 | CHANnel4 | EXTernanalog | SBUS | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | LOGIC | MSOB1 | MSOB2 | MSOB3 | MSOB4 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | LINE

**CHAN1 = CHANnel1, CHAN2 = CHANnel2, CHAN3 = CHANnel3, CHAN4 = CHANnel4**
Input channels

**EXTernanalog**
External analog signal connected to the External Trigger Input. For this source, only the analog edge trigger is available.

**LINE**
The instrument generates the trigger from the AC power input and synchronizes the signal to the AC power frequency.

**SBUS**
Serial bus

**D0...D15**
Digital channels (option R&S RTP-B1)
See also: Chapter 24.18.4, "Trigger settings for digital signals and parallel buses", on page 2321

**LOGIc**
Logic combination of digital channels, used as trigger source (option R&S RTP-B1)

**MSOB1 | MSOB2 | MSOB3 | MSOB4**
Parallel bus (option R&S RTP-B1)

**Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4**
Input channels of R&S RT-ZVCmulti-channel power probe. Only available in the A-trigger with trigger type EDGE.

**DIFF1 | DIFF2 | COMMON1 | COMMON2**
Differential signals

*RST: CHAN1

**Usage:** Asynchronous command

**Manual operation:** See "[Source]" on page 37

---

**TRIGger<m>:SOURce:SBSelect** <SerialBus>

Selects the serial bus to be triggered on.

**Suffix:**
<m> 1..3
Event in a trigger sequence: 1 = A-event only

**Parameters:**
<SerialBus> SBUS1 | SBUS2 | SBUS3 | SBUS4

**Usage:** Asynchronous command

**Firmware/software:** Version 2.70

**Manual operation:** See "Serial bus" on page 530

---

**BUS<m>:TYPE** <Type>

Defines the bus or protocol type for analysis. The type of available buses depends on the installed options.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Type> I2C | SPI | UART | CAN | CANFd | LIN | MILS1553 | MILStd1553 | ARIN429 | ARINc429 | SWIRe | MDIO | HBTO | USB | USBPD | ETHernet | CMSB | RFFE | DPHY | MPHY | EBTB | USB3 | PCIE | TBTO | DDR

MILS1553 = MILStd1553: specification MIL-STD-1553
ARIN429 = ARINc429: specification ARINC 429
HBTO: Ethernet 100BASE-T1
TBTO: Ethernet 1000BASE-T1
CMSB: custom decode serial bus
EBTB: 8b/10b general decoding
SWIRe: SpaceWire

*RST: I2C

**Usage:** Asynchronous command

**Manual operation:** See "Protocol" on page 517

## 24.17.3 I²C (option R&S RTP-K1)


- Configuration....1643
- Trigger....1646
- Decode results....1652
- I²C search settings....1661
- I²C search results....1670


### 24.17.3.1 Configuration


BUS<m>:I2C:SCL:SOURce....1643
BUS<m>:I2C:SDA:SOURce....1644
BUS<m>:I2C:SCL:THReshold....1644
BUS<m>:I2C:SDA:THReshold....1645
BUS<m>:I2C:TECHnology....1645
BUS<m>:I2C:RWBit....1645


---

**BUS<m>:I2C:SCL:SOURce** <SCLSource>

Sets the waveform of the clock line.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**

<SCLSource> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

Digital and analog channels cannot be used at the same time for data and clock lines.
See Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C2W1

**Usage:** Asynchronous command

**Manual operation:** See "SDA, SCL" on page 528

---

**BUS<m>:I2C:SDA:SOURce** <SDASource>

Sets the waveform of the data line.

**Suffix:**

<m> 1..4
Selects the serial bus.

**Parameters:**

<SDASource> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

Digital and analog channels cannot be used at the same time for data and clock lines.
See Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "SDA, SCL" on page 528

---

**BUS<m>:I2C:SCL:THReshold** <SCLThreshold>

Sets a user-defined threshold value for the clock line.

Alternatively, you can set the threshold according to the signal technology with `BUS<m>:I2C:TECHnology`.

**Suffix:**

<m> 1..4
Selects the serial bus.

**Parameters:**

<SCLThreshold> User-defined clock threshold

Range: -12 to 12
Increment: 0.1
*RST: 0
Default unit: V

**Manual operation:** See "Threshold" on page 528

**BUS<m>:I2C:SDA:THReshold** <SDAThreshold>

Sets a user-defined threshold value for the data line.

Alternatively, you can set the threshold according to the signal technology with `BUS<m>:I2C:TECHnology`.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<SDAThreshold> User-defined data threshold

Range: -12 to 12
Increment: 0.1
*RST: 0
Default unit: V

**Manual operation:** See "Threshold" on page 528

**BUS<m>:I2C:TECHnology** <Technology>

Sets the threshold voltage clock and data lines as defined for various signal technologies.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Technology> V15 | V25 | V165 | V125 | V09 | VM13 | V38 | V20 | V0 | MANual

**V15 | V25 | V165 | V125 | V09 | V38 | V20 | V0**
1.5 V, 2.5 V, 1.65 V ... respectively

**VM13**
-1.3 V (negative value)

**MANual**
Manual setting of user-defined values with `BUS<m>:I2C:SCL:THReshold` and `BUS<m>:I2C:SDA:THReshold`.

*RST: V165

**Manual operation:** See "Threshold" on page 528

**BUS<m>:I2C:RWBit** <BusConfig>

Defines if the R/W bit of a 7-bit address is considered separately or as part of the address. 10-bit addresses are not affected. The setting defines which address lengths are available with `TRIGger<m>:I2C:AMODe`.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<BusConfig> SEParate | INADdress

**SEParate**
7-bit address and separate R/W bit.

**INADdress**
8-bit address with R/W bit included.

*RST: SEParate

**Manual operation:** See "R/W bit" on page 528

### 24.17.3.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:I2C:TYPE ..... 1647
TRIGger<m>:I2C:ACCess ..... 1647
TRIGger<m>:I2C:ADNack ..... 1647
TRIGger<m>:I2C:DWNack ..... 1648
TRIGger<m>:I2C:DRNack ..... 1648
TRIGger<m>:I2C:AMODe ..... 1648
TRIGger<m>:I2C:ACONdition ..... 1649
TRIGger<m>:I2C:ADDRess ..... 1649
TRIGger<m>:I2C:ADDTo ..... 1649
TRIGger<m>:I2C:ADOR<n>:ENABle ..... 1649
TRIGger<m>:I2C:ADOR<n>:ADRType ..... 1650
TRIGger<m>:I2C:ADOR<n>[:VALue] ..... 1650
TRIGger<m>:I2C:ADOR<n>:RWBit ..... 1650
TRIGger<m>:I2C:DPOPerator ..... 1651
TRIGger<m>:I2C:DPOSition ..... 1651
TRIGger<m>:I2C:DPTO ..... 1651
TRIGger<m>:I2C:DCONdition ..... 1651
TRIGger<m>:I2C:DMIN ..... 1652
TRIGger<m>:I2C:DMAX ..... 1652

**TRIGger<m>:I2C:TYPE** <Type>

Selects the trigger type for I²C analysis.

**Parameters:**

<Type> STARt | REPStart | STOP | NACK | ADDRess | ADOR | ADAT

**STARt**
Start condition

**REPStart**
Repeated start - the start condition occurs without previous stop condition.

**NACK**
Missing acknowledge bit. To localize specific missing acknowledge bits, use `TRIGger<m>:I2C:ADNack`, `TRIGger<m>:I2C:DWNack`, and `TRIGger<m>:I2C:DRNack`.

**ADDRess**
Triggers on one specific address

**ADOR**
Triggers on an OR combination with up to four four address conditions.

**ADAT**
Triggers on a combination of address and data condition.

**STOP**
Stop condition, end of frame

*RST: STARt

**Usage:** Asynchronous command

**Manual operation:** See "Trigger type" on page 531

**TRIGger<m>:I2C:ACCess** <RWBitAddress>

Sets the trigger condition for the R/W bit - the transfer direction of the data.

**Parameters:**

<RWBitAddress> READ | WRITe | EITHer

**EITHer**
Transfer direction is not relevant.

*RST: EITHer

**Usage:** Asynchronous command

**Manual operation:** See "R/W bit" on page 534

**TRIGger<m>:I2C:ADNack** <AddressNack>

Triggers if the address acknowledge bit is missing - no slave recognizes the address.

**Parameters:**
<AddressNack> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "No Ack conditions" on page 533

---

**TRIGger<m>:I2C:DWNack** <DataWriteNack>

Triggers if a date acknowledge bit is missing - the addressed slave does not accept the data.

**Parameters:**
<DataWriteNack> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "No Ack conditions" on page 533

---

**TRIGger<m>:I2C:DRNack** <DataReadNack>

Triggers on the end of the read process when the master reads data from the slave. This Nack is sent according to the protocol definition, it is not an error.

**Parameters:**
<DataReadNack> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "No Ack conditions" on page 533

---

**TRIGger<m>:I2C:AMODe** <AddressType>

Sets the address length. The setting affects the address input with `TRIGger<m>:I2C:ADDRess` and `TRIGger<m>:I2C:ADDTo`.

**Parameters:**
<AddressType> BIT7 | BIT7_RW | BIT10 | ANY

**BIT7**
Enter the 7 address bits. Only available if `BUS<m>:I2C:RWBitSEParate` is set.

**BIT7_RW**
Enter 7 address bits and the R/W bit. Only available if `BUS<m>:I2C:RWBitINADdress` is set.

**BIT10**
10-bit address

**ANY**
Only available for trigger type "Address + data" (`TRIGger<m>:I2C:TYPE ADAT`). Used to trigger on data only, regardless of the address.
*RST: BIT7

**Usage:** Asynchronous command

**Manual operation:** See "Adress Type" on page 533

---

**TRIGger<m>:I2C:ACONdition** <AddressOperator>

Sets the operator to set a specific address or an address range. The address values are set with `TRIGger<m>:I2C:ADDRess` and `TRIGger<m>:I2C:ADDTo`.

**Parameters:**
<AddressOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange
*RST: EQUal

---

**TRIGger<m>:I2C:ADDRess** <Address>

Triggers on the specified slave address, or sets the the start value of an address range depending on the condition set with `TRIGger<m>:I2C:ACONdition`.

**Parameters:**
<Address> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Manual operation:** See "Address" on page 533

---

**TRIGger<m>:I2C:ADDTo** <AddressTo>

Sets the the end value of an address range if the condition is set to an address range with `TRIGger<m>:I2C:ACONdition`.

**Parameters:**
<AddressTo> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Usage:** Asynchronous command

**Manual operation:** See "Address" on page 533

---

**TRIGger<m>:I2C:ADOR<n>:ENABle** <UseAddress>

Includes the indicated ADOR address in the "address OR" trigger condition.

**Suffix:**
<n> 1..4
Index of the address in an "address OR" condition (OR slot)

**Parameters:**
<UseAddress> ON | OFF
*RST: OFF

**Manual operation:** See "OR x" on page 534

---

**TRIGger<m>:I2C:ADOR<n>:ADRType** <AddressType>

Sets the address type for the indicated ADOR address in the "address OR" trigger condition.

**Suffix:**
<n> 1..4
Index of the address in an "address OR" condition (OR slot)

**Parameters:**
<AddressType> BIT7 | BIT7_RW | BIT10
*RST: BIT7

**Manual operation:** See "Adress Type" on page 533

---

**TRIGger<m>:I2C:ADOR<n>[:VALue]** <Address>

Defines the address pattern of the indicated ADOR address in the "address OR" trigger condition.

**Suffix:**
<n> 1..4
Index of the address in an "address OR" condition (OR slot)

**Parameters:**
<Address> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Manual operation:** See "Address" on page 533

---

**TRIGger<m>:I2C:ADOR<n>:RWBit** <RWBit>

Defines the R/W bit of the indicated ADOR address in the "address OR" trigger condition.

**Suffix:**
<n> 1..4
Index of the address in an "address OR" condition (OR slot)

**Parameters:**
<RWBit> UNDefined | READ | WRITe | EITHer
**UNDefined**
Return value only
*RST: EITHer

**Manual operation:** See "R/W bit" on page 534

**TRIGger<m>:I2C:DPOPerator** <DataPosOperator>

Sets the operator for the data position. You can defined an exact position, or a position range.

**Parameters:**

<DataPosOperator> ANY | OFF | EQUal | GETHan | INRange | RANGe

**ANY = OFF**
The position of the required pattern is not relevant.

**EQUal | GETHan**
Equal, Greater or equal than. These conditions require one data position to be set with `TRIGger<m>:I2C:DPOSition`.

**INRange = RANGe**
In range: Set the minimum and maximum value of the range with `TRIGger<m>:I2C:DPOSition` and `TRIGger<m>:I2C:DPTO`.

*RST: ANY

**Manual operation:** See "Position" on page 534

**TRIGger<m>:I2C:DPOSition** <DataPosition>

Sets the number of data bytes before the first byte of interest. These bytes are ignored.

**Parameters:**

<DataPosition> The index 0 is associated with the first data byte.

Range: 0 to 4095
Increment: 1
*RST: 0

**Manual operation:** See "Position" on page 534

**TRIGger<m>:I2C:DPTO** <DataPositionTo>

Defines the last byte of interest, if `TRIGger<m>:I2C:DPOPerator` is set to `RANGe`.

**Parameters:**

<DataPositionTo> Range: 0 to 4095
Increment: 1
*RST: 0

**Manual operation:** See "Position" on page 534

**TRIGger<m>:I2C:DCONdition** <DataOperator>

Sets the operator to set a specific data value or a data range.

**Parameters:**

<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Manual operation:** See "Data" on page 534

---

**TRIGger<m>:I2C:DMIN** <Data>

Specifies the data bit pattern, or sets the the start value of a data pattern range.. Enter the bytes in msb first bit order. The maximum pattern length is 64 bit. Waveform data is compared with the pattern byte-by-byte.

**Parameters:**

<Data> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 534

---

**TRIGger<m>:I2C:DMAX** <DataTo>

Sets the the end value of an data range if `TRIGger<m>:I2C:DCONdition` is set to `INRange` or `OORange`.

**Parameters:**

<DataTo> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 534

### 24.17.3.3 Decode results

To load and acivate a label list, use:

- `BUS<m>:NEWList` on page 1639
- `BUS<m>:SYMBols` on page 1639

BUS<m>:I2C:FRAMe<n>:DATA? 1653
BUS<m>:I2C:FCOunt? 1653
BUS<m>:I2C:FRAMe<n>:AACCess? 1653
BUS<m>:I2C:FRAMe<n>:ACCess? 1654
BUS<m>:I2C:FRAMe<n>:ACOMplete? 1654
BUS<m>:I2C:FRAMe<n>:ADBStart? 1654
BUS<m>:I2C:FRAMe<n>:ADDRess? 1655
BUS<m>:I2C:FRAMe<n>:ADEVice? 1655
BUS<m>:I2C:FRAMe<n>:AMODe? 1656
BUS<m>:I2C:FRAMe<n>:ASTart? 1656
BUS<m>:I2C:FRAMe<n>:BITRate? 1656
BUS<m>:I2C:FRAMe<n>:RWBStart? 1657
BUS<m>:I2C:FRAMe<n>:STATus? 1657
BUS<m>:I2C:FRAMe<n>:STARt? 1658

BUS<m>:I2C:FRAMe<n>:STOP? ..... 1658
BUS<m>:I2C:FRAMe<n>:SYMBol? ..... 1658
BUS<m>:I2C:FRAMe<n>:BCOunt? ..... 1658
BUS<m>:I2C:FRAMe<n>:BYTE<o>:ACCess? ..... 1659
BUS<m>:I2C:FRAMe<n>:BYTE<o>:ACKStart? ..... 1659
BUS<m>:I2C:FRAMe<n>:BYTE<o>:COMPlete? ..... 1660
BUS<m>:I2C:FRAMe<n>:BYTE<o>:STARt? ..... 1660
BUS<m>:I2C:FRAMe<n>:BYTE<o>:VALue? ..... 1660

### BUS<m>:I2C:FRAMe<n>:DATA?

Returns the data words of the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<Data> Comma-separated list of integer values (N, D1, D2,…, DN). N is the number of bytes in the frame, and D1…DN are the values of the bytes.

**Example:**

```
BUS:I2C:FRAMe4:DATA?
<-- 3,74,164,18
```

**Usage:** Query only

### BUS<m>:I2C:FCOunt?

Returns the number of decoded frames.

**Suffix:**

<m> 1..4
Selects the serial bus.

**Return values:**

<Count> Total number of decoded frames.

**Usage:** Query only

### BUS<m>:I2C:FRAMe<n>:AACCess?

Returns the address acknowledge bit value for the indicated frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<AddressAckBit> INComplete | ACK | NACK | EITHer
*RST: INComplete

**Usage:** Query only

---

**BUS<m>:I2C:FRAMe<n>:ACCess?**

Returns the value of the R/W bit of the indicated frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<RWBit> UNDefined | READ | WRITe | EITHer
*RST: UNDefined

**Usage:** Query only

---

**BUS<m>:I2C:FRAMe<n>:ACOMplete?**

Returns if the address is completely contained in the acquisition.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<AddressComplete> ON | OFF
*RST: OFF

**Usage:** Query only

---

**BUS<m>:I2C:FRAMe<n>:ADBStart?**

Returns the start time of the address acknowledge bit.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<AddrAckBtStrt> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:I2C:FRAMe<n>:ADDRess?**

Returns the device address value of the indicated frame, that is, the address value that is shown in the decoded cells and in the decode results table.

If the frame has a 7-bit address, the command considers the status of `BUS<m>:I2C:RWBit`. If `BUS<m>:I2C:RWBit INADdress` is set, the returned address includes the R/W bit (8 bit). Otherwise, the pure address without the R/W bit is returned (7 bit, same result as returned with `BUS<m>:I2C:FRAMe<n>:ADEVice?`).

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<AddressValue> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and default – are decimal values.
Range: 0 to 1023
*RST: 0

**Usage:** Query only

---

**BUS<m>:I2C:FRAMe<n>:ADEVice?**

Returns the pure device address of the indicated frame *without* the R/W bit.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<DeviceAddress> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and default – are decimal values.
Range: 0 to 1023
*RST: 0

**Usage:** Query only

**BUS<m>:I2C:FRAMe<n>:AMODe?**

Returns the address length.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<AddressType> BIT7 | BIT7_RW | BIT10 | AUTO | ANY
*RST: BIT7

**Usage:** Query only

**BUS<m>:I2C:FRAMe<n>:ASTart?**

Returns the start time of the address for the indicated frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<AddressStart> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:I2C:FRAMe<n>:BITRate?**

Returns the primary bit rate.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<PrimaryBitRate> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

**BUS<m>:I2C:FRAMe<n>:RWBStart?**

Returns the start time of the R/W bit

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<RWBitStart> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:I2C:FRAMe<n>:STATus?**

Returns the overall state of the frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<FrameState> INComplete | OK | UNEXpstop | INSufficient | ADDifferent

**INComplete**
The stop bit is missing.

**OK**
The frame is valid.

**UNEXpstop**
A stop bit was detected but clock and data are continued.

**INSufficient**
The frame is not completely contained in the acquisition. The acquired part of the frame is valid.

**ADDifferent**
Error in 10 bit address. In case of a read access on a 10 bit address, the first address byte is sent twice, first as write, the second as read. The first seven bits of the byte must be identical. If they are not identical, the ADDiffernt error is indicated.

*RST: OK

**Usage:** Query only

**BUS<m>:I2C:FRAMe<n>:STARt?**

Returns the start time of the specified frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<FrameStart> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:I2C:FRAMe<n>:STOP?**

Returns the end time of the specified frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<FrameStop> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:I2C:FRAMe<n>:SYMBol?**

Returns the symbolic label of the specified frame if the label list is enabled.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the number of the frame in the current acquisition, 1...n.

**Return values:**
<Translation> String with symbolic name of the address

**Usage:** Query only

**BUS<m>:I2C:FRAMe<n>:BCOunt?**

Returns the number of bytes in the specified frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<Count> Byte count

**Usage:** Query only

---

**BUS<m>:I2C:FRAMe<n>:BYTE<o>:ACCess?**

Returns the acknowledge bit value of the specified data byte.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the byte number.

**Return values:**
<AckBit> INComplete | ACK | NACK | EITHer
*RST: INComplete

**Usage:** Query only

---

**BUS<m>:I2C:FRAMe<n>:BYTE<o>:ACKStart?**

Returns the start time of the acknowledge bit of the specified byte.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the byte number.

**Return values:**
<AckBitStart> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and reset – are decimal values.
Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:I2C:FRAMe<n>:BYTE<o>:COMPlete?**

Returns if the indicated byte is completely contained in the acquisition.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the byte number.

**Return values:**

<ValueComplete> ON | OFF
*RST: OFF

**Usage:** Query only

**BUS<m>:I2C:FRAMe<n>:BYTE<o>:STARt?**

Returns the start time of the specified data byte.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the byte number.

**Return values:**

<Start> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:I2C:FRAMe<n>:BYTE<o>:VALue?**

Returns the data value of the specified byte.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the byte number.

**Return values:**

<Value> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and default – are decimal values.

Range: 0 to 255
*RST: 0

**Usage:** Query only

### 24.17.3.4 I²C search settings

In search setup commands, you have to specify the <SearchName> parameter. It is a string parameter that contains the search definition name. The commands are similar to I²C trigger commands.

SEARch:TRIGger:I2C:SCONdition....1661
SEARch:TRIGger:I2C:RCONdition....1662
SEARch:TRIGger:I2C:STCNdition....1662
SEARch:TRIGger:I2C:NACKnowledge....1662
SEARch:TRIGger:I2C:SADDress....1662
SEARch:TRIGger:I2C:ADOR....1663
SEARch:TRIGger:I2C:ADData....1663
SEARch:TRIGger:I2C:ACONdition....1664
SEARch:TRIGger:I2C:AMODe....1664
SEARch:TRIGger:I2C:ADDRess....1665
SEARch:TRIGger:I2C:ADDTo....1665
SEARch:TRIGger:I2C:ACCess....1665
SEARch:TRIGger:I2C:ADDO<m>:ENABle....1666
SEARch:TRIGger:I2C:ADDO<m>:ADRType....1666
SEARch:TRIGger:I2C:ADDO<m>[:VALue]....1666
SEARch:TRIGger:I2C:ADDO<m>:RWBit....1667
SEARch:TRIGger:I2C:DPOPerator....1667
SEARch:TRIGger:I2C:DPOSition....1668
SEARch:TRIGger:I2C:DPTO....1668
SEARch:TRIGger:I2C:DCONdition....1668
SEARch:TRIGger:I2C:DMIN....1669
SEARch:TRIGger:I2C:DMAX....1669
SEARch:TRIGger:I2C:ADNack....1670
SEARch:TRIGger:I2C:DRNack....1670
SEARch:TRIGger:I2C:DWNack....1670

---

**SEARch:TRIGger:I2C:SCONdition** <SearchName>,<Start>
**SEARch:TRIGger:I2C:SCONdition?** <SearchName>

Enables the search for the start of the message.

**Parameters:**

<Start> ON | OFF

*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Select search types" on page 537

---

**SEARch:TRIGger:I2C:RCONdition** <SearchName>,<RepeatedStart>
**SEARch:TRIGger:I2C:RCONdition?** <SearchName>

Enables the search for a start condition without previous stop condition.

**Parameters:**
<RepeatedStart> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Select search types" on page 537

---

**SEARch:TRIGger:I2C:STCNdition** <SearchName>,<Stop>
**SEARch:TRIGger:I2C:STCNdition?** <SearchName>

Enables the search for the start of the message.

**Parameters:**
<Stop> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Select search types" on page 537

---

**SEARch:TRIGger:I2C:NACKnowledge** <SearchName>,<NoAcknowledge>
**SEARch:TRIGger:I2C:NACKnowledge?** <SearchName>

Searches for missing address acknowledge bits.

**Parameters:**
<NoAcknowledge> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Select search types" on page 537

---

**SEARch:TRIGger:I2C:SADDress** <SearchName>,<Address>
**SEARch:TRIGger:I2C:SADDress?** <SearchName>

Enables the search for one specific address condition or for a combination of address conditions.

To define the address condition, use the following commands:

- SEARch:TRIGger:I2C:ACONdition on page 1664
- SEARch:TRIGger:I2C:ADDRess on page 1665
- SEARch:TRIGger:I2C:ADDTo on page 1665
- SEARch:TRIGger:I2C:AMODe on page 1664
- SEARch:TRIGger:I2C:ACCess on page 1665

**Parameters:**
<Address> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Select search types" on page 537

---

**SEARch:TRIGger:I2C:ADOR** <SearchName>,<AddressOr>
**SEARch:TRIGger:I2C:ADOR?** <SearchName>

Enables the search for one to four address conditions.

- SEARch:TRIGger:I2C:ADDO<m>:ENABle on page 1666
- SEARch:TRIGger:I2C:ADDO<m>:ADRType on page 1666
- SEARch:TRIGger:I2C:ADDO<m>[:VALue] on page 1666
- SEARch:TRIGger:I2C:ADDO<m>:RWBit on page 1667

**Parameters:**
<AddressOr> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Select search types" on page 537

---

**SEARch:TRIGger:I2C:ADData** <SearchName>,<AddressData>
**SEARch:TRIGger:I2C:ADData?** <SearchName>

Enables the search for a combination of address and data conditions.

**Parameters:**
<AddressData> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Select search types" on page 537

**SEARch:TRIGger:I2C:ACONdition** <SearchName>,<AddressOperator>
**SEARch:TRIGger:I2C:ACONdition?** <SearchName>

Sets the operator to set a specific address or an address range.

**Parameters:**
<AddressOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These condition require one identifier pattern to be set with `SEARch:TRIGger:I2C:ADDRess` on page 1665.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `SEARch:TRIGger:I2C:ADDRess` on page 1665 and `SEARch:TRIGger:I2C:ADDTo` on page 1665.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Address setup: Condition, Type, Addr. min, Addr. max, R/W bit" on page 538

**SEARch:TRIGger:I2C:AMODe** <SearchName>,<AddressType>
**SEARch:TRIGger:I2C:AMODe?** <SearchName>

Sets the address length.

**Parameters:**
<AddressType> BIT7 | BIT7_RW | BIT10 | ANY

**BIT7 | BIT10**
Enter only address bits in the address pattern.

**BIT7_RW**
Enter seven address bits and also the R/W bit in the address pattern.

**ANY**
Only available for search criteria "Address and data" (`SEARch:TRIGger:I2C:ADData` is set `ON`). Used to search for data only, regardless of the address.

*RST: BIT7

**Parameters for setting and query:**
<SearchName>

**Usage:** Asynchronous command

**Manual operation:** See "Address setup: Condition, Type, Addr. min, Addr. max, R/W bit" on page 538

**SEARch:TRIGger:I2C:ADDRess** <SearchName>,<Address>
**SEARch:TRIGger:I2C:ADDRess?** <SearchName>

Specifies an address pattern, or sets the the start value of an address range.

**Parameters:**
<Address> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.
The pattern length is defined with `SEARch:TRIGger:I2C:AMODe`.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Address setup: Condition, Type, Addr. min, Addr. max, R/W bit" on page 538

**SEARch:TRIGger:I2C:ADDTo** <SearchName>,<AddressTo>
**SEARch:TRIGger:I2C:ADDTo?** <SearchName>

Sets the the end value of an address range if `SEARch:TRIGger:I2C:ACONdition` is set to `INRange` or `OORange`.

**Parameters:**
<AddressTo> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.
The pattern length is defined with `SEARch:TRIGger:I2C:AMODe`.

**Parameters for setting and query:**
<SearchName>

**Usage:** Asynchronous command

**Manual operation:** See "Address setup: Condition, Type, Addr. min, Addr. max, R/W bit" on page 538

**SEARch:TRIGger:I2C:ACCess** <SearchName>,<RWBitAddress>
**SEARch:TRIGger:I2C:ACCess?** <SearchName>

Sets the transfer direction of the data.

**Parameters:**
<RWBitAddress> READ | WRITe | EITHer
*RST: EITHer

**Parameters for setting and query:**
<SearchName>

**Usage:** Asynchronous command

**Manual operation:** See "Address setup: Condition, Type, Addr. min, Addr. max, R/W bit" on page 538

**SEARch:TRIGger:I2C:ADDO<m>:ENABle** <SearchName>,<UseAddress>
**SEARch:TRIGger:I2C:ADDO<m>:ENABle?** <SearchName>

Includes the indicated ADOR address in the "address OR" search condition.

**Suffix:**
<m> 1..4
Index of the address in an "address OR" condition (OR slot)

**Parameters:**
<UseAddress> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Address OR setup: , OR x Address type, Address, R/W bit" on page 539

**SEARch:TRIGger:I2C:ADDO<m>:ADRType** <SearchName>,<AddressType>
**SEARch:TRIGger:I2C:ADDO<m>:ADRType?** <SearchName>

Sets the address type for the indicated ADOR address in the "address OR" search condition.

**Suffix:**
<m> 1..4
Index of the address in an "address OR" condition (OR slot)

**Parameters:**
<AddressType> BIT7 | BIT7_RW | BIT10

**BIT7 | BIT10**
Enter only address bits in the address pattern.

**BIT7_RW**
Enter seven address bits and also the R/W bit in the address pattern.

*RST: BIT7

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Address OR setup: , OR x Address type, Address, R/W bit" on page 539

**SEARch:TRIGger:I2C:ADDO<m>[:VALue]** <SearchName>,<Address>
**SEARch:TRIGger:I2C:ADDO<m>[:VALue]?** <SearchName>

Defines the address pattern of the indicated ADOR address in the "address OR" search condition.

**Suffix:**
<m> 1..4
Index of the address in an "address OR" condition (OR slot)

**Parameters:**
<Address> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.
The pattern length is defined with `SEARch:TRIGger:I2C:ADDO<m>:ADRType`.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Address OR setup: , OR x Address type, Address, R/W bit" on page 539

---

**SEARch:TRIGger:I2C:ADDO<m>:RWBit** <SearchName>,<RWBit>
**SEARch:TRIGger:I2C:ADDO<m>:RWBit?** <SearchName>

Defines the R/W bit of the indicated ADOR address in the "address OR" search condition.

**Suffix:**
<m> 1..4
Index of the address in an "address OR" condition (OR slot)

**Parameters:**
<RWBit> UNDefined | READ | WRITe | EITHer

**UNDefined**
Only return value

*RST: EITHer

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Address OR setup: , OR x Address type, Address, R/W bit" on page 539

---

**SEARch:TRIGger:I2C:DPOPerator** <SearchName>,<DataPosOperator>
**SEARch:TRIGger:I2C:DPOPerator?** <SearchName>

Sets the operator for the data position. You can defined an exact position, or a position range.

**Parameters:**
<DataPosOperator> ANY | OFF | EQUal | GETHan | INRange | RANGe

**ANY = OFF**
The position of the required pattern is not relevant.

**EQUal | GETHan**
Equal, Greater or equal than. These conditions require one data position to be set with `SEARch:TRIGger:I2C:DPOSition`.

**INRange = RANGe**
In range: Set the minimum and maximum value of the range with `SEARch:TRIGger:I2C:DPOSition` and `SEARch:TRIGger:I2C:DPTO`.

*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: Condition, Position, Index min, Index max, Value min, Value max" on page 538

---

**SEARch:TRIGger:I2C:DPOSition** <SearchName>,<DataPosition>
**SEARch:TRIGger:I2C:DPOSition?** <SearchName>

Defines the first byte of interest. All bytes before that byte are ignored.

**Parameters:**
<DataPosition> The index 0 is associated with the first data byte.

Range: 0 to 4095
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: Condition, Position, Index min, Index max, Value min, Value max" on page 538

---

**SEARch:TRIGger:I2C:DPTO** <SearchName>,<DataPositionTo>
**SEARch:TRIGger:I2C:DPTO?** <SearchName>

Defines the last byte of interest, if `SEARch:TRIGger:I2C:DPOPerator` defines a range.

**Parameters:**
<DataPositionTo> Range: 0 to 4095
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: Condition, Position, Index min, Index max, Value min, Value max" on page 538

---

**SEARch:TRIGger:I2C:DCONdition** <SearchName>,<DataOperator>
**SEARch:TRIGger:I2C:DCONdition?** <SearchName>

Sets the operator to set a specific data value or a data range.

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange
*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: Condition, Position, Index min, Index max, Value min, Value max" on page 538

---

**SEARch:TRIGger:I2C:DMIN** <SearchName>,<Data>
**SEARch:TRIGger:I2C:DMIN?** <SearchName>

Specifies the data bit pattern, or sets the the start value of a data pattern range.. Enter the bytes in msb first bit order. The maximum pattern length is 64 bit. Waveform data is compared with the pattern byte-by-byte.

**Parameters:**
<Data> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Parameters for setting and query:**
<SearchName>

**Usage:** Asynchronous command

**Manual operation:** See "Data setup: Condition, Position, Index min, Index max, Value min, Value max" on page 538

---

**SEARch:TRIGger:I2C:DMAX** <SearchName>,<DataTo>
**SEARch:TRIGger:I2C:DMAX?** <SearchName>

Sets the the end value of an address range if `SEARch:TRIGger:I2C:DCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<DataTo> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Parameters for setting and query:**
<SearchName>

**Usage:** Asynchronous command

**Manual operation:** See "Data setup: Condition, Position, Index min, Index max, Value min, Value max" on page 538

**SEARch:TRIGger:I2C:ADNack** <SearchName>,<AddressNack>
**SEARch:TRIGger:I2C:ADNack?** <SearchName>

**Parameters:**
<AddressNack> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Usage:** Asynchronous command

**SEARch:TRIGger:I2C:DRNack** <SearchName>,<DataReadNack>
**SEARch:TRIGger:I2C:DRNack?** <SearchName>

Searches for the end of the read process when the master reads data from the slave. This Nack is sent according to the protocol definition, it is not an error.

**Parameters:**
<DataReadNack> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Usage:** Asynchronous command

**Manual operation:** See "No ACK setup: Addr/W/R NACK" on page 539

**SEARch:TRIGger:I2C:DWNack** <SearchName>,<DataWriteNack>
**SEARch:TRIGger:I2C:DWNack?** <SearchName>

Searches for missing data write acknowledge bits.

**Parameters:**
<DataWriteNack> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Usage:** Asynchronous command

**Manual operation:** See "No ACK setup: Addr/W/R NACK" on page 539

#### 24.17.3.5 I²C search results

The search on decoded CAN data returns the same results as the queries for decode results.

In search result commands, you have to specify the <SearchName> parameter. It is a string parameter that contains the search definition name.

For a description of the returned values, see the corresponding commands in Chapter 24.17.3.3, "Decode results", on page 1652.

SEARch:RESult:I2C:FCOunt?....1671
SEARch:RESult:I2C:FRAMe<m>:STATus?....1671
SEARch:RESult:I2C:FRAMe<m>:STARt?....1672
SEARch:RESult:I2C:FRAMe<m>:STOP?....1672
SEARch:RESult:I2C:FRAMe<m>:AACCess?....1672
SEARch:RESult:I2C:FRAMe<m>:ACCess?....1672
SEARch:RESult:I2C:FRAMe<m>:ACOMplete?....1673
SEARch:RESult:I2C:FRAMe<m>:ADBStart?....1673
SEARch:RESult:I2C:FRAMe<m>:ADDRess?....1673
SEARch:RESult:I2C:FRAMe<m>:ADEVice?....1674
SEARch:RESult:I2C:FRAMe<m>:AMODe?....1674
SEARch:RESult:I2C:FRAMe<m>:ASTart?....1674
SEARch:RESult:I2C:FRAMe<m>:DATA?....1674
SEARch:RESult:I2C:FRAMe<m>:RWBStart?....1675
SEARch:RESult:I2C:FRAMe<m>:SYMBol?....1675
SEARch:RESult:I2C:FRAMe<m>:BCOunt?....1675
SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:ACCess?....1675
SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:ACKStart?....1676
SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:COMPlete?....1676
SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:STARt?....1676
SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:VALue?....1677

---

**SEARch:RESult:I2C:FCOunt?** <SearchName>

**Query parameters:**
<SearchName>

**Return values:**

| | | |
|---|---|---|
| <Count> | Range: | 0 to 100000 |
| | Increment: | 1 |
| | *RST: | 0 |

**Usage:** Query only

---

**SEARch:RESult:I2C:FRAMe<m>:STATus?** <SearchName>

**Suffix:**

| | |
|---|---|
| <m> | * |

**Query parameters:**
<SearchName>

**Return values:**

| | | |
|---|---|---|
| <State> | INComplete \| OK \| UNEXpstop \| INSufficient \| ADDifferent | |
| | *RST: | OK |

**Usage:** Query only

**SEARch:RESult:I2C:FRAMe<m>:STARt?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:I2C:FRAMe<m>:STOP?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:I2C:FRAMe<m>:AACCess?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<AddressAckBit> INComplete | ACK | NACK | EITHer
*RST: INComplete

**Usage:** Query only

**SEARch:RESult:I2C:FRAMe<m>:ACCess?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<RWBit> UNDefined | READ | WRITe | EITHer
*RST: UNDefined

**Usage:** Query only

---

**SEARch:RESult:I2C:FRAMe<m>:ACOMplete?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<AddressComplete> ON | OFF
*RST: OFF

**Usage:** Query only

---

**SEARch:RESult:I2C:FRAMe<m>:ADBStart?** <SearchName>

Returns the start time of the address acknowledge bit.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<AddrAckBtStrt> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:I2C:FRAMe<m>:ADDRess?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<AddressValue> Range: 0 to 2047
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:I2C:FRAMe<m>:ADEVice?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<DeviceAddress> Range: 0 to 1023
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:I2C:FRAMe<m>:AMODe?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<AddressType> BIT7 | BIT7_RW | BIT10 | AUTO | ANY
*RST: BIT7

**Usage:** Query only

**SEARch:RESult:I2C:FRAMe<m>:ASTart?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<AddressStart> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:I2C:FRAMe<m>:DATA?** <SearchName>

Returns the data bytes of the indicated frame.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

---

**SEARch:RESult:I2C:FRAMe<m>:RWBStart?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<RWBitStart> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:I2C:FRAMe<m>:SYMBol?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Translation>

**Usage:** Query only

---

**SEARch:RESult:I2C:FRAMe<m>:BCOunt?**

**Suffix:**
<m> *

**Return values:**
<Count>

**Usage:** Query only

---

**SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:ACCess?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<ByteAckBit> INComplete | ACK | NACK | EITHer
*RST: INComplete

**Usage:** Query only

---

**SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:ACKStart?** <SearchName>

Returns the start time of the acknowledge bit of the indicated data byte.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<ByteAckBitStart> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:COMPlete?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<ByteComplete> ON | OFF
*RST: OFF

**Usage:** Query only

---

**SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:STARt?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**

| | | |
|---|---|---|
| <ByteStart> | Range: | -100E+24 to 100E+24 |
| | Increment: | 100E-12 |
| | *RST: | 0 |
| | Default unit: | s |

**Usage:** Query only

---

**SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:VALue?** <SearchName>

**Suffix:**

| | |
|---|---|
| <m> | * |
| <n> | * |

**Query parameters:**
<SearchName>

**Return values:**

| | | |
|---|---|---|
| <Value> | Range: | 0 to 255 |
| | Increment: | 1 |
| | *RST: | 0 |

**Usage:** Query only

## 24.17.4 SPI (option R&S RTP-K1)


- SPI bus configuration....1677
- SPI trigger....1682
- SPI decode results....1685
- SPI search settings....1690
- SPI search results....1693


### 24.17.4.1 SPI bus configuration


BUS<m>:SPI:BORDer....1678
BUS<m>:SPI:WSIZe....1678
BUS<m>:SPI:SCLK:SOURce....1678
BUS<m>:SPI:SSELect:SOURce....1679
BUS<m>:SPI:SSELect:POLarity....1679
BUS<m>:SPI:MISO:SOURce....1679
BUS<m>:SPI:MISO:POLarity....1680
BUS<m>:SPI:MOSI:SOURce....1680
BUS<m>:SPI:MOSI:POLarity....1681
BUS<m>:SPI:SCLK:THReshold....1681
BUS<m>:SPI:MISO:THReshold....1681
BUS<m>:SPI:MOSI:THReshold....1681
BUS<m>:SPI:SSELect:THReshold....1681
BUS<m>:SPI:FRCondition....1681
BUS<m>:SPI:TIMeout....1682

**BUS<m>:SPI:BORDer** <BitOrder>

Defines if the data of the messages starts with msb (most significant bit) or lsb (least significant bit).

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<BitOrder> LSBF | MSBF
*RST: MSBF

**Usage:** Asynchronous command

**Manual operation:** See "Bit order" on page 543

**BUS<m>:SPI:WSIZe** <WordLength>

Sets the number of bits in a message.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<WordLength> Number of bits
Range: 4 to 32
Increment: 1
*RST: 8

**Usage:** Asynchronous command

**Manual operation:** See "Word length" on page 543

**BUS<m>:SPI:SCLK:SOURce** <SCLKSource>

Sets the input channel of the clock line.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<SCLKSource> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15
Digital and analog channels cannot be used at the same time for data, clock and slave select lines.
See Chapter 24.4.2, "Waveform parameter", on page 1193
*RST: C2W1

**Usage:** Asynchronous command

**Manual operation:** See "SCLK source" on page 541

---

**BUS<m>:SPI:SSELect:SOURce** <SlaveSelectSource>

Sets the input channel of the slave select line.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<SlaveSelectSource> NONE | C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

Digital and analog channels cannot be used at the same time for data, clock and slave select lines.
See Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: None

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 542

---

**BUS<m>:SPI:SSELect:POLarity** <SSPolarity>

Selects whether transmitted slave select signal is high active (high = 1) or low active (low = 1).

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<SSPolarity> ACTLow | ACTHigh

*RST: ACTLow

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 542

---

**BUS<m>:SPI:MISO:SOURce** <MISOSource>

Sets the input channel of the MISO line.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**

<MISOSource> NONE | C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

Digital and analog channels cannot be used at the same time for data, clock and slave select lines.
See Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: None

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 542

---

**BUS<m>:SPI:MISO:POLarity** <MISOPolarity>

Selects whether transmitted data is high active (high = 1) or low active (low = 1).

**Suffix:**

<m> 1..4
Selects the serial bus.

**Parameters:**

<MISOPolarity> ACTLow | ACTHigh

*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 542

---

**BUS<m>:SPI:MOSI:SOURce** <MOSISource>

Sets the input channel of the MOSI line.

**Suffix:**

<m> 1..4
Selects the serial bus.

**Parameters:**

<MOSISource> NONE | C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

Digital and analog channels cannot be used at the same time for data, clock and slave select lines.
See Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 542

**BUS<m>:SPI:MOSI:POLarity** <MOSIPolarity>

Selects whether transmitted data is high active (high = 1) or low active (low = 1).

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<MOSIPolarity> ACTLow | ACTHigh
*RST: ACTHigh

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 542

**BUS<m>:SPI:SCLK:THReshold** <SCLKThreshold>
**BUS<m>:SPI:MISO:THReshold** <MISOThreshold>
**BUS<m>:SPI:MOSI:THReshold** <MOSIThreshold>
**BUS<m>:SPI:SSELect:THReshold** <SSThreshold>

Set user-defined threshold values for the clock, MISO, MOSI and slave select lines.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<SSThreshold> User-defined value
Range: -12 to 12
Increment: 0.1
*RST: 0
Default unit: V

**Manual operation:** See "Set Thresholds" on page 542

**BUS<m>:SPI:FRCondition** <FrameCondition>

Defines the start of a frame. A frame contains a number of successive words, at least one word.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<FrameCondition> SS | CLKTimeout

**SS**
Start and end of the frame is defined by the active state of the slave select signal, see `BUS<m>:SPI:SSELect:POLarity`.

**CLKTimeout**
Defines a timeout on the clock line SCLK as limiter between two frames. The timeout condition is used for SPI connections without an SS line.

*RST: SS

**Manual operation:** See "Frame condition" on page 544

---

**BUS<m>:SPI:TIMeout** <ClockTimeout>

Defines a timeout on the clock line SCLK as limiter between two frames. The timeout condition is used for SPI connections without an SS line.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ClockTimeout> Range: 50E-9 to 10
Increment: 1E-6
*RST: 1E-3
Default unit: s

**Manual operation:** See "Timeout" on page 544

### 24.17.4.2 SPI trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:SPI:TYPE ..... 1682
TRIGger<m>:SPI:PALignment ..... 1683
TRIGger<m>:SPI:DPOPerator ..... 1683
TRIGger<m>:SPI:DPOSition ..... 1684
TRIGger<m>:SPI:DPTO ..... 1684
TRIGger<m>:SPI:FCONdition ..... 1684
TRIGger<m>:SPI:MISopattern ..... 1684
TRIGger<m>:SPI:MOSipattern ..... 1685

---

**TRIGger<m>:SPI:TYPE** <Type>

Selects the trigger type for SPI analysis.

**Parameters:**
<Type> SSACtive | TIMeout | MOSI | MISO | MOMI

**SSACtive**
Start of the message: slave select signal SS changes to the active state.

**TIMeout**
Triggers on the next message start after the "Timeout" time.

**MOSI**
Triggers on a specified data pattern in that is expected on the MOSI line. Define the pattern with `TRIGger<m>:SPI:MOSipattern`.

**MISO**
Triggers on a specified data pattern in that is expected on the MISO line. Define the pattern with `TRIGger<m>:SPI:MISopattern`

**MOMI**
Triggers on a specified data patterns on the MISO and MISO lines.

*RST: SSACtive

**Usage:** Asynchronous command

**Manual operation:** See "Trigger type" on page 545

---

**TRIGger<m>:SPI:PALignment** <DataAlignment>

Defines how the specified data pattern is searched.

**Parameters:**
<DataAlignment> WORD | BIT

**WORD**
The pattern is matched only at word boundaries.

**BIT**
Bit-by bit: the pattern can be at any position in the data word.

*RST: WORD

**Usage:** Asynchronous command

**Manual operation:** See "Search mode" on page 547

---

**TRIGger<m>:SPI:DPOPerator** <DataPosOperator>

Sets the operator for the data position. You can defined an exact position, or a position range.

**Parameters:**
<DataPosOperator> ANY | OFF | EQUal | GETHan | INRange | RANGe

**ANY = OFF**
The position of the required pattern is not relevant.

**EQUal | GETHan**
Equal, Greater or equal than. These conditions require one data position to be set with `TRIGger<m>:SPI:DPOSition`.

**INRange = RANGe**
Set the minimum and maximum value of the range with `TRIGger<m>:SPI:DPOSition` and `TRIGger<m>:SPI:DPTO`.

*RST: ANY

**Manual operation:** See "Position" on page 546

---

**TRIGger<m>:SPI:DPOSition** <DataPosition>

Sets the number of bits or words to be ignored before the first bit or word od interest. The effect is defined by `TRIGger<m>:SPI:PALignment`.

**Parameters:**
<DataPosition> The index 0 is associated with the first data byte.
Range: 0 to 4095 for triggering on one line (MISO or MOSI), 2047 for triggering on both lines.
Increment: 1
*RST: 0

**Manual operation:** See "Index min, Index max" on page 547

---

**TRIGger<m>:SPI:DPTO** <DataPositionTo>

Defines the last bit or word of interest, if `TRIGger<m>:SPI:DPOPerator` is set to `INRange`.

**Parameters:**
<DataPositionTo> Range: 1 to 4095 for triggering on one line (MISO or MOSI), 2047 for triggering on both lines.
Increment: 1
*RST: 1

**Manual operation:** See "Index min, Index max" on page 547

---

**TRIGger<m>:SPI:FCONdition** <DataOperator>

Selects the operator for the MISO and MOSI pattern.

**Parameters:**
<DataOperator> EQUal | NEQual
*RST: EQUal

**Manual operation:** See "Condition" on page 546

---

**TRIGger<m>:SPI:MISopattern** <MISOPattern>

Specifies the pattern to be triggered on the MISO line.

**Parameters:**

<MISOPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Usage:** Asynchronous command

**Manual operation:** See "MOSI pattern, MISO pattern" on page 546

---

**TRIGger<m>:SPI:MOSipattern** <MOSIPattern>

Specifies the pattern to be triggered on the MOSI line.

**Parameters:**

<MOSIPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Usage:** Asynchronous command

**Manual operation:** See "MOSI pattern, MISO pattern" on page 546

### 24.17.4.3 SPI decode results

BUS<m>:SPI:FRAMe<n>:DATA? 1685
BUS<m>:SPI:FCOunt? 1686
BUS<m>:SPI:FRAMe<n>:COUNt? 1686
BUS<m>:SPI:FRAMe<n>:BITRate? 1686
BUS<m>:SPI:FRAMe<n>:STATus? 1686
BUS<m>:SPI:FRAMe<n>:STARt? 1687
BUS<m>:SPI:FRAMe<n>:STOP? 1687
BUS<m>:SPI:FRAMe<n>:WCOunt? 1688
BUS<m>:SPI:FRAMe<n>:WORD<o>:STARt? 1688
BUS<m>:SPI:FRAMe<n>:WORD<o>:STOP? 1688
BUS<m>:SPI:FRAMe<n>:WORD<o>:MISO? 1689
BUS<m>:SPI:FRAMe<n>:WORD<o>:FMISo? 1689
BUS<m>:SPI:FRAMe<n>:WORD<o>:MOSI? 1690
BUS<m>:SPI:FRAMe<n>:WORD<o>:FMOSi? 1690

---

**BUS<m>:SPI:FRAMe<n>:DATA?**

Returns the data words of the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<FrameData> Comma-separated sequence of integer values (N, L1, R1,…, LN, RN). N is the number of word pairs in the frame, and {L1,R1} …{LN,RN} are the value pairs. The values Lx and Rx are associated with the MOSI and the MISO channel, respectively. If a channel is disabled, an empty value is returned.

**Example:** `BUS:SPI:FRAMe3:DATA?`
`<-- 2,10,108,35,70` (MOSI+MISO)
`2,10,,35,` (MOSI only)
`2,,108,,70` (MISO only)

**Usage:** Query only

---

**BUS<m>:SPI:FCOunt?**
**BUS<m>:SPI:FRAMe<n>:COUNt?**

Returns the number of decoded frames.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *

**Return values:**

<Count> Total number of decoded frames.

**Usage:** Query only

---

**BUS<m>:SPI:FRAMe<n>:BITRate?**

Returns the primary bit rate.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<PrimaryBitRate> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**BUS<m>:SPI:FRAMe<n>:STATus?**

Returns the overall state of the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<FrameState> OK | VOID | INCFirst | INCLast | INSufficient

OK: the frame is valid.
VOID: the frame is empty.
INCFirst: INComplete First word. The first word does not have the expected word length.
INCLast: INComplete Last word. The last word does not have the expected word length.
INSufficient: The frame is not completely contained in the acquisition. The acquired part of the frame is valid.

*RST: OK

**Usage:** Query only

---

**BUS<m>:SPI:FRAMe<n>:STARt?**

Returns the start time of the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<FrameStart> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:SPI:FRAMe<n>:STOP?**

Returns the end time of the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<FrameStop> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:SPI:FRAMe<n>:WCOunt?**

Returns the number of words in the specified frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<WordCount> Range: 0 to 4096
*RST: 0

**Usage:** Query only

---

**BUS<m>:SPI:FRAMe<n>:WORD<o>:STARt?**

Returns the start time of the specified data word.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the word number.

**Return values:**
<Start> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:SPI:FRAMe<n>:WORD<o>:STOP?**

Returns the end time of the specified data word.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the word number.

**Return values:**

<Stop> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:SPI:FRAMe<n>:WORD<o>:MISO?**

Returns the data value of the specified word on the MISO line.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the word number.

**Return values:**

<MISOValue> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and reset – are decimal values.
Range: 0 to 4294967295
*RST: 0

**Usage:** Query only

---

**BUS<m>:SPI:FRAMe<n>:WORD<o>:FMISo?**

Returns the formatted value of the specified word on the MISO line.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the word number.

**Return values:**

<FormattedMISOVal>

**Usage:** Query only

**BUS<m>:SPI:FRAMe<n>:WORD<o>:MOSI?**

Returns the data value of the specified word on the MOSI line.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the word number.

**Return values:**

<MOSIValue> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and reset – are decimal values.
Range: 0 to 4294967295
*RST: 0

**Usage:** Query only

**BUS<m>:SPI:FRAMe<n>:WORD<o>:FMOSi?**

Returns the formatted value of the specified word on the MOSI line.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the word number.

**Return values:**

<FormattedMOSIVal>

**Usage:** Query only

### 24.17.4.4 SPI search settings

In search setup commands, you must specify the <SearchName> parameter. It is a string parameter that contains the search definition name. All commands are similar to SPI trigger commands.

SEARch:TRIGger:SPI:TYPE ..... 1691
SEARch:TRIGger:SPI:FCONdition ..... 1691
SEARch:TRIGger:SPI:MISopattern ..... 1692
SEARch:TRIGger:SPI:MOSipattern ..... 1692
SEARch:TRIGger:SPI:DPOPerator ..... 1692

SEARch:TRIGger:SPI:DPOSition....1692
SEARch:TRIGger:SPI:DPTO....1693
SEARch:TRIGger:SPI:PALignment....1693

**SEARch:TRIGger:SPI:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:SPI:TYPE?** <SearchName>

Sets the event to be searched for.

**Parameters:**
<Type> SSACtive | TIMeout | MOSI | MISO | MOMI

**SSACtive**
Searches for the start of the frame when slave select signal SS changes to the active state. This type is available if the slave select line is configured in the bus setup, and `BUS<m>:SPI:FRCondition` is `SS`.

**TIMeout**
Searches for the start of the frame when the clock idle time exceeds the timeout. This type is available if the slave select line is configured in the bus setup, and `BUS<m>:SPI:FRCondition` is `CLKTimeout`.

**MOSI | MISO**
Searches for a specified data pattern expected on the MOSI line or on the MISO line, respectively.

**MOMI**
Searches in parallel for specified data patterns expected on the MOSI and MISO lines.

*RST: SSACtive

**Parameters for setting and query:**
<SearchName>

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 549

**SEARch:TRIGger:SPI:FCONdition** <SearchName>,<DataOperator>
**SEARch:TRIGger:SPI:FCONdition?** <SearchName>

Selects the operator for the data pattern: equal or not equal.

**Parameters:**
<DataOperator> EQUal | NEQual

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Firmware/software:** FW 3.30

**Manual operation:** See "MOSI and MISO data search" on page 549

**SEARch:TRIGger:SPI:MISopattern** <SearchName>,<MISOPattern>
**SEARch:TRIGger:SPI:MISopattern?** <SearchName>
**SEARch:TRIGger:SPI:MOSipattern** <SearchName>,<MOSIPattern>
**SEARch:TRIGger:SPI:MOSipattern?** <SearchName>

Specifies a data pattern for the MISO or MOSI line, respectively.

**Parameters:**
<MISOPattern>
<MOSIPattern>
Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Parameters for setting and query:**
<SearchName>

**Usage:** Asynchronous command

**Firmware/software:** FW 3.30

**Manual operation:** See "MOSI and MISO data search" on page 549

---

**SEARch:TRIGger:SPI:DPOPerator** <SearchName>,<DataPosOperator>
**SEARch:TRIGger:SPI:DPOPerator?** <SearchName>

Operator for the data position. You can define an exact position, a position range, or let the position undefined (ANY).

**Parameters:**
<DataPosOperator> ANY | OFF | EQUal | GETHan | INRange | RANGe
ANY = OFF, INRange = RANGe
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Firmware/software:** FW 3.30

**Manual operation:** See "MOSI and MISO data search" on page 549

---

**SEARch:TRIGger:SPI:DPOSition** <SearchName>,<DataPosition>
**SEARch:TRIGger:SPI:DPOSition?** <SearchName>

Sets the number of bits or words before the first word of interest, see also `SEARch:TRIGger:SPI:PALignment`. These offset bits/words are skipped. The index 0 is associated with the first data bit or word.

If the position operator defines a range, also define the last bit/word of interest using `SEARch:TRIGger:SPI:DPTO`

**Parameters:**
<DataPosition> Range: 0 to 32767
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Firmware/software:** FW 3.30

**Manual operation:** See "MOSI and MISO data search" on page 549

---

**SEARch:TRIGger:SPI:DPTO** <SearchName>,<DataPositionTo>
**SEARch:TRIGger:SPI:DPTO?** <SearchName>

Sets the the end value of a data postion range.

**Parameters:**
<DataPositionTo> Range: 1 to 32767
Increment: 1
*RST: 1

**Parameters for setting and query:**
<SearchName>

**Firmware/software:** FW 3.30

**Manual operation:** See "MOSI and MISO data search" on page 549

---

**SEARch:TRIGger:SPI:PALignment** <SearchName>,<DataAlignment>
**SEARch:TRIGger:SPI:PALignment?** <SearchName>

Defines how the specified data pattern is searched.

**Parameters:**
<DataAlignment> WORD | BIT

**WORD**
The pattern is matched only at word boundaries.

**BIT**
Bit-by-bit: the pattern can start at any position in the message.

*RST: WORD

**Parameters for setting and query:**
<SearchName>

**Usage:** Asynchronous command

**Firmware/software:** FW 3.30

**Manual operation:** See "MOSI and MISO data search" on page 549

### 24.17.4.5 SPI search results

The search on decoded SPI data returns the same results as the queries for decode results.

In search result commands, you must specify the <SearchName> parameter. It is a string parameter that contains the search definition name.

The suffix FRAMe<m> indicates the frame index. The suffix WORD<n> indicates the word index inside a frame-

For a description of the returned values, see the corresponding commands in Chapter 24.17.4.3, "SPI decode results", on page 1685.

All SPI search commands are first implemented in firmware version 3.30.

SEARch:RESult:SPI:FCOunt? ..... 1694
SEARch:RESult:SPI:FRAMe<m>:COUNt? ..... 1694
SEARch:RESult:SPI:FRAMe<m>:DATA? ..... 1694
SEARch:RESult:SPI:FRAMe<m>:STARt? ..... 1695
SEARch:RESult:SPI:FRAMe<m>:STATus? ..... 1695
SEARch:RESult:SPI:FRAMe<m>:STOP? ..... 1695
SEARch:RESult:SPI:FRAMe<m>:WCOunt? ..... 1696
SEARch:RESult:SPI:FRAMe<m>:WORD<n>:MISO? ..... 1696
SEARch:RESult:SPI:FRAMe<m>:WORD<n>:MOSI? ..... 1696
SEARch:RESult:SPI:FRAMe<m>:WORD<n>:STARt? ..... 1696
SEARch:RESult:SPI:FRAMe<m>:WORD<n>:STOP? ..... 1697

---

**SEARch:RESult:SPI:FCOunt?** <SearchName>

**Query parameters:**
<SearchName>

**Return values:**

| | | |
|---|---|---|
| <Count> | Range: | 0 to 100000 |
| | Increment: | 1 |
| | *RST: | 0 |

**Usage:** Query only

---

**SEARch:RESult:SPI:FRAMe<m>:COUNt?**

Returns the number of frames that have matched the search criteria. In the search result table on the display, the number of rows is the number of frames that match the search criteria.

**Suffix:**
<m> *

**Return values:**
<Count>

**Usage:** Query only

---

**SEARch:RESult:SPI:FRAMe<m>:DATA?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameData>

**Usage:** Query only

---

**SEARch:RESult:SPI:FRAMe<m>:STARt?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameStart> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:SPI:FRAMe<m>:STATus?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameState> OK | VOID | INCFirst | INCLast | INSufficient
*RST: OK

**Usage:** Query only

---

**SEARch:RESult:SPI:FRAMe<m>:STOP?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameStop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:SPI:FRAMe<m>:WCOunt?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameWordCount> Range: 0 to 4096
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:SPI:FRAMe<m>:WORD<n>:MISO?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<WordMISOValue> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:SPI:FRAMe<m>:WORD<n>:MOSI?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<WordMOSIValue> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:SPI:FRAMe<m>:WORD<n>:STARt?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameWordStart> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:SPI:FRAMe<m>:WORD<n>:STOP?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameWordStop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

## 24.17.5 UART/RS-232/RS-422/RS-485 (option R&S RTP-K2)


- Configuration....1697
- Trigger....1703
- Decode results....1705


### 24.17.5.1 Configuration


BUS<m>:UART:RX:SOURce....1698
BUS<m>:UART:TX:SOURce....1698
BUS<m>:UART:RX:THReshold....1698
BUS<m>:UART:TX:THReshold....1699
BUS<m>:UART:TECHnology....1699
BUS<m>:UART:BITRate....1700
BUS<m>:UART:BAUDrate....1700
BUS<m>:UART:PARity....1700
BUS<m>:UART:POLarity....1701
BUS<m>:UART:SBIT....1701
BUS<m>:UART:SSIZe....1701
BUS<m>:UART:PACKets....1702
BUS<m>:UART:TOUT....1702
BUS<m>:UART:EWORd....1703

**BUS<m>:UART:RX:SOURce** <RxSource>

Selects the input channel for the receiver signal.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<RxSource> NONE | C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

Digital and analog channels cannot be used at the same time for RX and TX lines.
See Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: None

**Usage:** Asynchronous command

**Manual operation:** See "Source: Tx, Rx" on page 552

**BUS<m>:UART:TX:SOURce** <TxSource>

Selects the input channel for the transmitter signal.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<TxSource> NONE | C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

Digital and analog channels cannot be used at the same time for RX and TX lines.
See Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "Source: Tx, Rx" on page 552

**BUS<m>:UART:RX:THReshold** <RxThreshold>

Sets a user-defined threshold value for the Rx line.

Alternatively, you can set the threshold according to the signal technology with `BUS<m>:UART:TECHnology`.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**

<RxThreshold> User-defined clock threshold

Range: -15 to 15
Increment: 0.1
*RST: 0
Default unit: V

**Manual operation:** See "Threshold" on page 552

---

**BUS<m>:UART:TX:THReshold** <TxThreshold>

Sets a user-defined threshold value for the Tx line.

Alternatively, you can set the threshold according to the signal technology with `BUS<m>:UART:TECHnology`.

**Suffix:**

<m> 1..4
Selects the serial bus.

**Parameters:**

<TxThreshold> User-defined clock threshold

Range: -15 to 15
Increment: 0.1
*RST: 0
Default unit: V

**Manual operation:** See "Threshold" on page 552

---

**BUS<m>:UART:TECHnology** <Technology>

Sets the threshold voltage Tx and Rx lines as defined for various signal technologies.

**Suffix:**

<m> 1..4
Selects the serial bus.

**Parameters:**

<Technology> V15 | V25 | V165 | V125 | V09 | VM13 | V38 | V20 | V0 | MANual

**V15 | V25 | V165 | V125 | V09 | V38 | V20 | V0**
1.5 V, 2.5 V, 1.65 V ... respectively

**VM13**
-1.3 V (negative value)

**MANual**
Manual setting of user-defined values with `BUS<m>:UART:RX:THReshold` and `BUS<m>:UART:TX:THReshold`.

*RST: V165

**Manual operation:** See "Threshold" on page 552

**BUS<m>:UART:BITRate** <Bitrate>

Sets the number of transmitted bits per second.

**Suffix:**
<m> 1..4

**Parameters:**
<Bitrate> Range: 300 to 20000000
Increment: 1
*RST: 9600
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "Bit rate" on page 552

**BUS<m>:UART:BAUDrate** <Bitrate>

Same as `BUS<m>:UART:BITRate`.

**Suffix:**
<m> 1..4

**Parameters:**
<Bitrate> Range: 300 to 20000000
Increment: 1
*RST: 9600
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "Bit rate" on page 552

**BUS<m>:UART:PARity** <Parity>

Defines the optional parity bit that is used for error detection.

See also: "Parity" on page 553.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Parity> NONE | ODD | EVEN | MARK | SPC | DC

**MARK**
The parity bit is always a logic 1.

**SPC**
SPaCe: The parity bit is always a logic 0.

**DC**
Don't Care: the parity is ignored.

*RST: NONE

**Usage:** Asynchronous command

**Manual operation:** See "Parity" on page 553

---

**BUS<m>:UART:POLarity** <Polarity>

Defines the idle state of the bus. The idle state corresponds to a logic 1. The transmitted data on the bus is high (high = 1) or low (low = 1) active.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Polarity> IDLLow | IDLHigh
*RST: IDLHigh

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 552

---

**BUS<m>:UART:SBIT** <StopBits>

Sets the number of stop bits: 1; 1.5 or 2 stop bits are possible.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<StopBits> B1 | B15 | B2
*RST: B1

**Usage:** Asynchronous command

**Manual operation:** See "Stop bits" on page 553

---

**BUS<m>:UART:SSIZe** <DataBits>

Sets the number of data bits in a message.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<DataBits> Number of data bits. 9 data bits are only possible with parity = none.
Range: 5 to 9
Increment: 1
*RST: 8

**Usage:** Asynchronous command

**Manual operation:** See "Data bits" on page 553

---

**BUS<m>:UART:PACKets** <Packets>

Defines the method of packet separation. A packet is a number of subsequent words in a date stream.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Packets> NONE | EWORd | TOUT

**NONE**
Packets are not considered.

**EWORd**
End word, the end condition of a packet is a pattern. To define the end word, use `BUS<m>:UART:EWORd`

**TOUT**
Defines a timeout between the packets. To set the timeout, use `BUS<m>:UART:TOUT`

*RST: NONE

**Firmware/software:** FW 2.25

**Manual operation:** See "Packets" on page 554

---

**BUS<m>:UART:TOUT** <InterframeTime>

Sets the timeout between packets in a UART data stream. A new packet starts with the first start bit after the timeout.

The command is relevant if `BUS<m>:UART:PACKets` is set to `TOUT`.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<InterframeTime> Range: 1E-6 to 1
Increment: 1
*RST: 1E-3
Default unit: s

**Usage:** Asynchronous command

**Firmware/software:** FW 2.25

**Manual operation:** See "Packets" on page 554

**BUS<m>:UART:EWORd** <EndOfFrame>

Sets the end pattern of the packets. A new packet starts with the first start bit after the defined end pattern.

The command is relevant if `BUS<m>:UART:PACKets` is set to `EWORd`.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<EndOfFrame> End word value in decimal format (range 0 to 255) or hexadecimal format (prefix #H). The query always returns hexadecimal values.

**Example:**
```
:BUS:UART:PACK EWOR
:BUS:UART:EWOR 10 // Decimal value
:BUS:UART:EWOR?
#H0A // Query returns hex
```

**Example:**
```
:BUS:UART:PACK EWOR
:BUS:UART:EWOR #Hff // Hexadecimal, prefix #H
:BUS:UART:EWOR?
#HFF
```

**Usage:** Asynchronous command

**Firmware/software:** FW 2.25

**Manual operation:** See "Packets" on page 554

### 24.17.5.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:UART:TYPE....1704
TRIGger<m>:UART:SOURce....1704
TRIGger<m>:UART:DPOPerator....1704
TRIGger<m>:UART:DPOSition....1704
TRIGger<m>:UART:DPTO....1705
TRIGger<m>:UART:FCONdition....1705
TRIGger<m>:UART:DATA....1705

**TRIGger<m>:UART:TYPE** <Type>

Selects the trigger type for UART analysis.

See also: "Type" on page 555

**Parameters:**

<Type> STBT | PCKS | DATA | PRER | BRKC | STPerror

STBT: Start bit
PCKS: Packet start
DATA: Serial pattern
PRER: Parity error
BRKC: Break condition
STPerror: Stop error

*RST: STBT

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 555

**TRIGger<m>:UART:SOURce** <Source>

Selects the transmitter or receiver line as trigger source.

**Parameters:**

<Source> TX | RX

*RST: TX

**Usage:** Asynchronous command

**Manual operation:** See "Trigger source" on page 556

**TRIGger<m>:UART:DPOPerator** <DataPosOperator>

Sets the operator for the data position. You can defined an exact position, or a position range.

**Parameters:**

<DataPosOperator> EQUal | GETHan | INRange | RANGe

INRange = RANGe

*RST: GETHan

**Usage:** Asynchronous command

**Manual operation:** See "Position" on page 557

**TRIGger<m>:UART:DPOSition** <DataPosition>

Sets the number of words before the first word of interest. These offset words are ignored.

**Parameters:**
<DataPosition> Number of words
Range: 0 to 32767
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Index min, Index max" on page 557

---

**TRIGger<m>:UART:DPTO** <DataPositionTo>

Defines the last word of interest, if `TRIGger<m>:UART:DPOPerator` defines a position range.

**Parameters:**
<DataPositionTo> Range: 0 to 32767
Increment: 1
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Index min, Index max" on page 557

---

**TRIGger<m>:UART:FCONdition** <DataOperator>

Selects the operator for the data pattern (`TRIGger<m>:UART:DATA`).

**Parameters:**
<DataOperator> EQUal | NEQual
*RST: EQUal

**Manual operation:** See "Condition" on page 556

---

**TRIGger<m>:UART:DATA** <Data>

Specifies the data pattern to be found on the specified trigger source, in binary or hex format. Enter the words in msb first bit order.

**Parameters:**
<Data> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Usage:** Asynchronous command

**Manual operation:** See "Pattern" on page 557

### 24.17.5.3 Decode results

BUS<m>:UART:WORD<n>:RXValue? ..... 1706
BUS<m>:UART:WORD<n>:TXValue? ..... 1706

BUS<m>:UART:WORD<n>:BITRate? ..... 1706
BUS<m>:UART:WORD<n>:COUNt? ..... 1706
BUS<m>:UART:WORD<n>:SOURce? ..... 1707
BUS<m>:UART:WORD<n>:STARt? ..... 1707
BUS<m>:UART:WORD<n>:STATe? ..... 1707

**BUS<m>:UART:WORD<n>:RXValue?**
**BUS<m>:UART:WORD<n>:TXValue?**

Returns the value of the specified word on the Rx line or Tx line, respectively.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**
<Value> To set the value format, use `FORMat:BPATtern`.
The stated values for range, increment and reset are decimal values.

Range: 0 to 511
*RST: 0

**Usage:** Query only

**BUS<m>:UART:WORD<n>:BITRate?**

Returns the primary bit rate.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
The suffix is irrelevant.

**Return values:**
<PrimaryBitRate> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

**BUS<m>:UART:WORD<n>:COUNt?**

Returns the number of words in the acquisition.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
The suffix is irrelevant.

**Return values:**
<Count> Number of words

**Usage:** Query only

---

**BUS<m>:UART:WORD<n>:SOURce?**

Returns the line on which the specified word was transferred.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**
<WordSource> TX | RX
*RST: TX

**Usage:** Query only

---

**BUS<m>:UART:WORD<n>:STARt?**

Returns the start time of the specified word.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**
<WordStart> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:UART:WORD<n>:STATe?**

Returns the status of the specified word.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**

<WordState> OK | FRSTart | FRENd | FRME | BREak | STERror | SPERror | PRERror | INSufficient

OK: the frame is valid.
BREak:stop bit error with 0x00 word
STERror: StarT ERror, incorrect start bit
SPERror: StoP ERror, incorrect stop bit
PRERror:PaRity ERror, incorrect parity bit.
INSufficient: The frame is not completely contained in the acquisition. The acquired part of the frame is valid.

*RST: OK

**Usage:** Query only

## 24.17.6 CAN (option R&S RTP-K3/K9)


- Configuration....1708
- Trigger....1713
- Decode results....1722
- Search settings....1731
- Search results....1741
- Symbolic trigger, decode and search....1747


### 24.17.6.1 Configuration


BUS<m>:CAN:DATA:SOURce....1708
BUS<m>:CAN:TYPE....1709
BUS<m>:CAN:FDATa:PSTandard....1709
BUS<m>:CAN:DATA:THReshold....1710
BUS<m>:CAN:TECHnology....1710
BUS<m>:CAN:BITRate....1710
BUS<m>:CAN:FDATa:ENABle....1711
BUS<m>:CAN:FDATa:DBITrate....1711
BUS<m>:CAN:FDATa:SAMPlepoint....1711
BUS<m>:CAN:SAMPlepoint....1711
BUS<m>:CAN:FDATa:T1Segment....1712
BUS<m>:CAN:T1Segment....1712
BUS<m>:CAN:FDATa:T2Segment....1712
BUS<m>:CAN:T2Segment....1712
BUS<m>:CAN:FDATa:JWIDth....1713
BUS<m>:CAN:JWIDth....1713


**BUS<m>:CAN:DATA:SOURce** <DataSource>

Sets the source of the data line that is selected with `BUS<m>:CAN:TYPE`.

Usually, the source is one of the analog channels. Reference and math waveforms are available as source if the trigger source is one of the analog channels but not the serial bus.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<DataSource> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15
*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 561

---

**BUS<m>:CAN:TYPE** <SignalType>

Selects the CAN-High or CAN-Low line. Both lines are required for differential signal transmission used by CAN.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<SignalType> CANH | CANL
*RST: CANL

**Manual operation:** See "Type" on page 562

---

**BUS<m>:CAN:FDATa:PSTandard** <ProtStd>

Only available for CAN FD buses. Selects whether the tested signal is an ISO CAN FD signal or not.

**Suffix:**
<m> 1..4

**Parameters:**
<ProtStd> ISO | NISO

**ISO**
Signals are decoded according to the the ISO CAN FD protocol. This protocol has an additional stuff count field before the CRC sequence.

**NISO**
Non-ISO. Signals are decoded according to the the Bosch CAN FD protocol.

*RST: ISO

**Firmware/software:** FW 3.35

**Manual operation:** See "Standard" on page 562

---

**BUS<m>:CAN:DATA:THReshold** <Threshold>

Sets a user-defined threshold value.

Alternatively, you can set the threshold according to the signal technology with `BUS<m>:CAN:TECHnology`.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Threshold> Range: -12 to 12
Increment: 0.1
*RST: 0
Default unit: V

**Manual operation:** See "Threshold" on page 562

---

**BUS<m>:CAN:TECHnology** <Technology>

Sets the threshold voltage as defined for various signal technologies.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Technology> V25 | V3 | V2 | V0 | MANual

**V25**
2.5 Volt (CMOS 5.0 V)

**V3**
3.0 Volt (CAN_H HS / CAN_L LS)

**V2**
2.0 Volt (CAN_L HS / CAN_H LS)

**V0**
Ground

**MANual**
Manual setting of user-defined values with `BUS<m>:CAN:DATA:THReshold`.

*RST: V25

**Manual operation:** See "Threshold" on page 562

---

**BUS<m>:CAN:BITRate** <Bitrate>

For CAN buses, the "Bit rate" sets the number of transmitted bits per second.

For CAN FD buses, this parameter is called "Arbitration rate" and sets the bit rate of the arbitration phase.

The maximum bit rate for High Speed CAN is 1 Mbit/s. The bit rate is uniform and fixed for a given CAN or CAN FD bus.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Bitrate> Range: 100 to 5E+6
Increment: 1
*RST: 100E+3
Default unit: bps

**Manual operation:** See "Bit rate (CAN) / Arbitration rate (CAN FD)" on page 563

---

**BUS<m>:CAN:FDATa:ENABle** <CANFDEnabled>

Enables the CAN FD protocol configuration. The setting is available in CAN FD option R&S RTP-K9.

**Suffix:**
<m> 1..4

**Parameters:**
<CANFDEnabled> ON | OFF
*RST: ON

**Manual operation:** See "CAN-FD" on page 562

---

**BUS<m>:CAN:FDATa:DBITrate** <FlexDatBitrate>

Sets the bit rate of the data phase. The data rate can be higher than the arbitration rate, but it is uniform and fixed for a given CAN FD bus. The setting is available in CAN FD option R&S RTP-K9.

**Suffix:**
<m> 1..4

**Parameters:**
<FlexDatBitrate> Range: 100 to 15E+6
Increment: 1
*RST: 1E+6
Default unit: bps

**Manual operation:** See "Data rate" on page 563

---

**BUS<m>:CAN:FDATa:SAMPlepoint** <FlexDatSmpPt>
**BUS<m>:CAN:SAMPlepoint** <SamplePoint>

Sets the position of the sample point within the bit in percent of the nominal bit time.

Alternatively, you can set the sample point with `BUS<m>:CAN:T1Segment` and `BUS<m>:CAN:T2Segment`.

For CAN FD signals, `BUS<m>:CAN:SAMPlepoint` defines the synchronization of the arbitration phase, and `BUS<m>:CAN:FDATa:SAMPlepoint` defines the synchronization of the data phase.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<FlexDatSmpPt>
<SamplePoint>
Range: 12 to 96 if number of segments is 24. The range reduces if the signal has fewer segments.
Increment: 5
*RST: 66
Default unit: %

**Manual operation:** See "Sample point" on page 565

---

**BUS<m>:CAN:FDATa:T1Segment** <FlexDatTimeSeg1>
**BUS<m>:CAN:T1Segment** <TimeSeg1>

Sets the number of time quanta before the sample point (T1Segment). T1Segment comprises the segments Synch_seg, Prop_seg, and Phase_seg1 which are specified in the CAN standard.

Make sure to set also `BUS<m>:CAN:T2Segment` for correct definition of the sample point. Alternatively, you can use `BUS<m>:CAN:SAMPlepoint`.

See also: Chapter 13.5.2.2, "Advanced settings", on page 563

For CAN FD signals, `BUS<m>:CAN:T1Segment` defines the synchronization of the arbitration phase, and `BUS<m>:CAN:FDATa:T1Segment` defines the synchronization of the data phase.

**Suffix:**
<m> 1..4

**Parameters:**
<FlexDatTimeSeg1>
<TimeSeg1>
Time quanta
Range: 3 to 23
Increment: 1
*RST: 6.6

**Manual operation:** See "Time segment1, Time segment2" on page 565

---

**BUS<m>:CAN:FDATa:T2Segment** <FlexDatTimeSeg2>
**BUS<m>:CAN:T2Segment** <TimeSeg2>

Sets the number of time quanta after the sample point (T2Segment). T2Segment matches Phase_seg2 specified in the CAN standard.

Make sure to set also BUS<m>:CAN:T1Segment on page 1712 for correct definition of the sample point. Alternatively, you can use BUS<m>:CAN:SAMPlepoint.

See also: Chapter 13.5.2.2, "Advanced settings", on page 563

For CAN FD signals, BUS<m>:CAN:T2Segment defines the synchronization of the arbitration phase, and BUS<m>:CAN:FDATa:T2Segment defines the synchronization of the data phase.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<FlexDatTimeSeg2>
<TimeSeg2> Time quanta
Range: 1 to 21
Increment: 1
*RST: 3.4

**Manual operation:** See "Time segment1, Time segment2" on page 565

---

**BUS<m>:CAN:FDATa:JWIDth** <FlexDatJumpWdt>
**BUS<m>:CAN:JWIDth** <JumpWidth>

Defines the maximum number of time quanta for phase correction. Time segment1 may be lengthened or Time segment2 may be shortened due to resynchronization. Resynchronization corrects the phase error of an edge caused by the drift of the oscillators.

For CAN FD signals, this setting defines the synchronization of the arbitration phase.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<JumpWidth> Time quanta
Range: 1 to 4, available maximum depends on the number of segments and the sample point
Increment: 1
*RST: 1

**Manual operation:** See "Jump width" on page 565

### 24.17.6.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- TRIGger<m>:SOURce[:SELect] is set to SBUS.
- The sources of the serial bus are channel signals: use BUS<m>:...:SOURce commands.

- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:CAN:TYPE ..... 1714
TRIGger<m>:CAN:FDATa:STANdard ..... 1715
TRIGger<m>:CAN:FTYPe ..... 1715
TRIGger<m>:CAN:ITYPe ..... 1716
TRIGger<m>:CAN:ICONdition ..... 1716
TRIGger<m>:CAN:IMIN ..... 1716
TRIGger<m>:CAN:IMAX ..... 1717
TRIGger<m>:CAN:FDATa:FDF ..... 1717
TRIGger<m>:CAN:FDATa:BRS ..... 1717
TRIGger<m>:CAN:FDATa:ESI ..... 1717
TRIGger<m>:CAN:DCONdition ..... 1718
TRIGger<m>:CAN:DMIN ..... 1718
TRIGger<m>:CAN:DMAX ..... 1718
TRIGger<m>:CAN:BORDer ..... 1719
TRIGger<m>:CAN:DLCCondition ..... 1719
TRIGger<m>:CAN:DLC ..... 1719
TRIGger<m>:CAN:NDBYtes? ..... 1720
TRIGger<m>:CAN:FDATa:DPOPerator ..... 1720
TRIGger<m>:CAN:FDATa:DPOSition ..... 1720
TRIGger<m>:CAN:FDATa:DPTO ..... 1721
TRIGger<m>:CAN:ACKerror ..... 1721
TRIGger<m>:CAN:BITSterror ..... 1721
TRIGger<m>:CAN:CRCerror ..... 1721
TRIGger<m>:CAN:FORMerror ..... 1722
TRIGger<m>:CAN:FDATa:SCERror ..... 1722

---

**TRIGger<m>:CAN:TYPE** <Type>

Selects the trigger type for CAN analysis.

See: "Trigger type" on page 568.

**Parameters:**

<Type> STOF | FTYP | ID | IDDT | ERRC

**STOF**
STart Of Frame: triggers on the first edge of the dominant SOF bit (synchronization bit).

**FTYP**
Frame TYPe: triggers on a specified frame type (data, remote, error, or overload) and on the identifier format.
To set the frame type, use `TRIGger<m>:CAN:FTYPe`. ´Set the identifier format with `TRIGger<m>:CAN:ITYPe`

**ID**
IDentifier: Sets the trigger to one specific identifier or an identifier range. To set the identifier, use `TRIGger<m>:CAN:ICONdition`, `TRIGger<m>:CAN:IMIN`, and `TRIGger<m>:CAN:IMAX`.

**IDDT**
IDentifier and DaTa: Combination of identifier and data conditions To set the identifier condition, use `TRIGger<m>:CAN:ICONdition`, `TRIGger<m>:CAN:IMIN`, and `TRIGger<m>:CAN:IMAX`.
To set the data condition, use `TRIGger<m>:CAN:DCONdition`, `TRIGger<m>:CAN:DMIN`, and `TRIGger<m>:CAN:DMAX`.

**ERRC**
ERRor Condition: Define the error types with `TRIGger<m>:CAN:ACKerror`, `TRIGger<m>:CAN:BITSterror`, `TRIGger<m>:CAN:CRCerror`, `TRIGger<m>:CAN:FORMerror`, `TRIGger<m>:CAN:FDATa:SCERror` on page 1722.

*RST: STOF

**Manual operation:** See "Trigger type" on page 568

---

**TRIGger<m>:CAN:FDATa:STANdard** <Standard>

Selects the CAN standard. Use `ANY` if the standard of the signal is unknown. The setting is available in CAN FD option R&S RTP-K9.

**Parameters:**
<Standard> ANY | CAN | CANFd
*RST: CAN

**Manual operation:** See "Standard" on page 570

---

**TRIGger<m>:CAN:FTYPe** <FrameType>

Selects the CAN frame type if `TRIGger<m>:CAN:TYPE` is set to FTYP (frame type) or ID (identifier).

For data and remote frames, the identifier format has to be set with `TRIGger<m>:CAN:ITYPe`.

See also: "Frame type" on page 570

**Parameters:**
<FrameType> ANY | DATA | REMote | ERRor | OVERload
Available values depend on the CAN standard and on the `TRIGger<m>:CAN:TYPE` setting:
Remote frames are not available in the CAN FD protocol.
If the trigger type is set to FTYP (frame type), you can set the values `DATA | REMote | ERRor | OVERload`.
If the trigger type is set to ID (identifier), you can set the values `ANY | DATA | REMote`.
*RST: ANY

**Manual operation:** See "Frame type" on page 570

---

**TRIGger<m>:CAN:ITYPe** <IdentifierType>

Selects the format of data and remote frames.

Remote frames are not available in the CAN FD protocol.

**Parameters:**

<IdentifierType> ANY | B11 | B29

**B11**
11 bit identifier (standard format). The instrument triggers on the sample point of the IDE bit.

**B29**
29 bit identifier (extended format). The instrument triggers on the sample point of the RTR bit.

**ANY**
The ID type and ID pattern are not relevant for the trigger condition.

*RST: ANY

**Manual operation:** See "ID type" on page 570

---

**TRIGger<m>:CAN:ICONdition** <IdOperator>

Sets the operator to set a specific identifier or an identifier range.

**Parameters:**

<IdOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These condition require one identifier pattern to be set with `TRIGger<m>:CAN:IMIN`.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `TRIGger<m>:CAN:IMIN` and `TRIGger<m>:CAN:IMAX` on page 1717.

*RST: EQUal

**Manual operation:** See "Identifier setup: Condition, Identifier min, Identifier max" on page 571

---

**TRIGger<m>:CAN:IMIN** <IdPattern>

Specifies a message identifier pattern, or sets the the start value of an identifier range.

**Parameters:**

<IdPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Manual operation:** See "Identifier setup: Condition, Identifier min, Identifier max" on page 571

---

**TRIGger<m>:CAN:IMAX** <IdPatternTo>

Sets the the end value of an identifier range if `TRIGger<m>:CAN:ICONdition` is set to `INRange` or `OORange`.

**Parameters:**

<IdPatternTo> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Manual operation:** See "Identifier setup: Condition, Identifier min, Identifier max" on page 571

---

**TRIGger<m>:CAN:FDATa:FDF** <FDFBit>

Specifies the CAN FD frame format. It corresponds to the EDL bit (extended data length), which only exists in CAN FD format.

**Parameters:**

<FDFBit> ONE | ZERO | DC

ONE: CAN FD.
ZERO: CAN.
DC: don't care, the format is not relevant.

*RST: DC

**Manual operation:** See "FD bits" on page 571

---

**TRIGger<m>:CAN:FDATa:BRS** <BRSBit>

Sets the bit rate switch bit.

**Parameters:**

<BRSBit> ONE | ZERO | DC

ONE: the bit rate switches from the bit rate of the arbitration phase to the faster data rate.

*RST: ONE

**Manual operation:** See "FD bits" on page 571

---

**TRIGger<m>:CAN:FDATa:ESI** <ESIBit>

Sets the error state indicator bit.

**Parameters:**

<ESIBit> ONE | ZERO | DC

DC: don't care, bit is nor relevant

*RST: DC

**Manual operation:** See "FD bits" on page 571

---

**TRIGger<m>:CAN:DCONdition** <DataOperator>

Sets the operator to set a specific data pattern or a data pattern range.

**Parameters:**

<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with `TRIGger<m>:CAN:DMIN`.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `TRIGger<m>:CAN:DMIN` and `TRIGger<m>:CAN:DMAX`.

*RST: EQUal

**Manual operation:** See "Data setup: DLC, NDB, Transfer, Condition, Data min, Data max" on page 572

---

**TRIGger<m>:CAN:DMIN** <DataPattern>

Specifies a data pattern, or sets the the start value of a data pattern range.

**Parameters:**

<DataPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data setup: DLC, NDB, Transfer, Condition, Data min, Data max" on page 572

---

**TRIGger<m>:CAN:DMAX** <DataPatternTo>

Sets the the end value of an data range if `TRIGger<m>:CAN:DCONdition` is set to `INRange` or `OORange`.

**Parameters:**

<DataPatternTo> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data setup: DLC, NDB, Transfer, Condition, Data min, Data max" on page 572

**TRIGger<m>:CAN:BORDer** <Endianness>

Sets the byte order (endianness) of the data transfer. Only for CAN protocol.

**Parameters:**

<Endianness> BENDian | LENDian

**BENDian**
Big endian, data is analyzed and evaluated in the order of reception.

**LENDian**
Little endian, the instrument reads the complete data, reverses the byte order of the data, and compares it with the specified data word.

*RST: BENDian

**Manual operation:** See "Data setup: DLC, NDB, Transfer, Condition, Data min, Data max" on page 572

**TRIGger<m>:CAN:DLCCondition** <DLCOperator>

Operator to set the data length code for triggering on CAN and CAN FD data.

For details, see "Data setup: DLC, NDB, Transfer, Condition, Data min, Data max" on page 572.

The number of data bytes to be found is set with `TRIGger<m>:CAN:DLC`.

See also: `TRIGger<m>:CAN:BORDer`.

**Parameters:**

<DLCOperator> EQUal | GETHan

For little endian transfer direction, EQUal must be set.

*RST: GETHan

**Manual operation:** See "Data setup: DLC, NDB, Transfer, Condition, Data min, Data max" on page 572

**TRIGger<m>:CAN:DLC** <WordCount>

Sets the Data Length Code, the number of data bytes to be found. For complete definition, set also the operator with `TRIGger<m>:CAN:DLCCondition` on page 1719.

**Parameters:**

<WordCount> Range: CAN: 1 to 8, CAN FD: 1 to 15 (64 bytes)
Increment: 1
*RST: 1

**Manual operation:** See "Data setup: DLC, NDB, Transfer, Condition, Data min, Data max" on page 572

**TRIGger<m>:CAN:NDBYtes?**

Returns the number of data bytes defined by DLC. DLC and NDB are different in CAN FD for DLCs > 8.

See also: "Data setup: DLC, NDB, Transfer, Condition, Data min, Data max" on page 572.

**Return values:**

<NDBytes> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Query only

**Manual operation:** See "Data setup: DLC, NDB, Transfer, Condition, Data min, Data max" on page 572

**TRIGger<m>:CAN:FDATa:DPOPerator** <DataPosOperator>

Sets the operator to define an exact position or a data range where the instrument looks for the specified data pattern. The setting is available in CAN FD option R&S RTP-K9.

The position can be defined if the data field of the frame is longer than 8 bytes - if `TRIGger<m>:CAN:DLC`≥9.

**Parameters:**

<DataPosOperator> ANY | OFF | EQUal | GETHan | INRange | RANGe

**ANY = OFF**
The data position is not relevant for the trigger condition.

**EQUal | GETHan**
Equal, Greater or equal than. These conditions require one data position to be set with `TRIGger<m>:CAN:FDATa:DPOSition`.

**INRange = RANGe**
In range: Set the minimum and maximum value of the range with `TRIGger<m>:CAN:FDATa:DPOSition` and `TRIGger<m>:CAN:FDATa:DPTO`.

*RST: ANY

**Manual operation:** See "Data position" on page 573

**TRIGger<m>:CAN:FDATa:DPOSition** <DataPosition>

Defines the number of the first data byte at which the data pattern may start. The setting is available in CAN FD option R&S RTP-K9.

**Parameters:**

<DataPosition> Range: 1 to 57
Increment: 1
*RST: 1

**Manual operation:** See "Data position" on page 573

---

**TRIGger<m>:CAN:FDATa:DPTO** <DataPositionTo>

Sets the number of the last byte at which the required data pattern may start.

**Parameters:**

<DataPositionTo> Range: 8 to 64
Increment: 1
*RST: 8

**Manual operation:** See "Data position" on page 573

---

**TRIGger<m>:CAN:ACKerror** <AckError>

Triggers when the transmitter does not receive an acknowledgment - a dominant bit during the Ack Slot.

The trigger type has to be set before: `TRIGger<m>:CAN:TYPE` to `ERRC`.

**Parameters:**

<AckError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 573

---

**TRIGger<m>:CAN:BITSterror** <BitStuffError>

Triggers if astuff error occurs - when the 6th consecutive equal bit level in the mentioned fields is detected.

The trigger type has to be set before: `TRIGger<m>:CAN:TYPE` to `ERRC`.

**Parameters:**

<BitStuffError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 573

---

**TRIGger<m>:CAN:CRCerror** <ChecksumError>

Triggers on CRC errors. A CRC error occurs when the CRC calculated by the receiver differs from the received value in the CRC sequence.

The trigger type has to be set before: `TRIGger<m>:CAN:TYPE` to `ERRC`.

**Parameters:**

<ChecksumError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 573

**TRIGger<m>:CAN:FORMerror** <FormError>

Triggers when a fixed-form bit field contains one or more illegal bits.

The trigger type has to be set before: `TRIGger<m>:CAN:TYPE` to `ERRC`.

**Parameters:**
<FormError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 573

**TRIGger<m>:CAN:FDATa:SCERror** <StuffCountError>

Triggers on stuff count errors. A stuff count error occurs if th received stuff count value does not match the value calculated from the own stuff bit count.

The trigger type `TRIGger<m>:CAN:TYPE` must be set to `ERRC`.

Only relevant for CAN FD signals in ISO standard.

**Parameters:**
<StuffCountError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 573

### 24.17.6.3 Decode results

To load and acivate a label list, use:

- `BUS<m>:NEWList` on page 1639
- `BUS<m>:SYMBols` on page 1639

BUS<m>:CAN:FCOunt? 1723
BUS<m>:CAN:FDATa:FRAMe<n>:STANdard? 1723
BUS<m>:CAN:FRAMe<n>:STATus? 1723
BUS<m>:CAN:FRAMe<n>:NDBYtes? 1724
BUS<m>:CAN:FRAMe<n>:STARt? 1725
BUS<m>:CAN:FRAMe<n>:STOP? 1725
BUS<m>:CAN:FRAMe<n>:SYMBol? 1725
BUS<m>:CAN:FRAMe<n>:TYPE? 1725
BUS<m>:CAN:FRAMe<n>:DATA? 1726
BUS<m>:CAN:FRAMe<n>:ACKState? 1726
BUS<m>:CAN:FRAMe<n>:CSSTate? 1726
BUS<m>:CAN:FRAMe<n>:DLCState? 1726
BUS<m>:CAN:FRAMe<n>:IDSTate? 1726
BUS<m>:CAN:FRAMe<n>:ACKValue? 1726
BUS<m>:CAN:FRAMe<n>:BITRate? 1727
BUS<m>:CAN:FRAMe<n>:CSValue? 1727
BUS<m>:CAN:FRAMe<n>:DLCValue? 1728
BUS<m>:CAN:FRAMe<n>:IDTYpe? 1728
BUS<m>:CAN:FRAMe<n>:IDValue? 1728

BUS<m>:CAN:FRAMe<n>:BSEPosition? ..... 1729
BUS<m>:CAN:FRAMe<n>:FERCause? ..... 1729
BUS<m>:CAN:FRAMe<n>:SDEXport? ..... 1729
BUS<m>:CAN:FDATa:FRAMe<n>:SCValue? ..... 1730
BUS<m>:CAN:FRAMe<n>:BYTE<o>:STATe? ..... 1730
BUS<m>:CAN:FRAMe<n>:BYTE<o>:VALue? ..... 1730

**BUS<m>:CAN:FCOunt?**

Returns the number of decoded frames of the acquisition.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Return values:**
<Count> Total number of decoded frames.
Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:CAN:FDATa:FRAMe<n>:STANdard?**

Returns the CAN standard. The setting is available in CAN FD option R&S RTP-K9.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> The frame suffix is irrelevant.

**Return values:**
<Standard> CAN | CANFd
*RST: CAN

**Usage:** Query only

**Manual operation:** See "CAN-FD" on page 562

**BUS<m>:CAN:FRAMe<n>:STATus?**

Returns the overall state of the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<FrameState> OK | FORM | BTST | CRC | CRCD | NOACk | ACKD | EOFD | CAERror | FCERor | INSufficient | SERRror | SFERror | SCERror | SAERror | SCAE | SCFE

OK: the frame is valid.
FORM: Fixed-bit form error
BTST: Bit stuffing error occured.
CRC: Cyclic redundancy check failed.
CRCD: Wrong CRC delimiter occured.
NOACK: Acknowlegde is missing.
ACKD: Wrong ACK delimiter occured.
EOFD: Wrong end of frame.
CAERror: CRC error followed by an acknowledgement error (missing acknowledge)
FCERor: CRC error followed by a form error (wrong CRC delimiter or wrong ACK delimiter)
INSufficient: The frame is not completely contained in the acquisition. The acquired part of the frame is valid.
SERRror: Stuff count error (CAN-FD ISO only)
SFER: Stuff count error and FORM error (CAN-FD ISO only)
SCER: Stuff count error and CRC error (CAN-FD ISO only)
SAER: Stuff count error and ACK error (CAN-FD ISO only)
SCAE: Stuff count error and CRC error and ACK error (CAN-FD ISO only)
SCFE: Stuff count error and CRC error and FORM error (CAN-FD ISO only)

*RST: OK

**Usage:** Query only

---

**BUS<m>:CAN:FRAMe<n>:NDBYtes?**

REturns the number of data bytes.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<NDBytes> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Query only

**Firmware/software:** FW 3.35

**BUS<m>:CAN:FRAMe<n>:STARt?**
**BUS<m>:CAN:FRAMe<n>:STOP?**

Return the start time and stop time of the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<Start>, <Stop> Time
Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:CAN:FRAMe<n>:SYMBol?**

Returns the symbolic label of the specified frame if the label list is enabled.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the number of the frame in the current acquisition, 1...n.

**Return values:**
<Label> String with symbolic label of the identifier

**Example:** `BUS:CAN:FRAMe:SYMBol?`
Response: `Temperature`

**Usage:** Query only

**BUS<m>:CAN:FRAMe<n>:TYPE?**

Returns the frame type of the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<FrameType> DATA | REMote | ERR | OVLD
Data, remote, error or overload frame.
*RST: DATA

**Usage:** Query only

---

**BUS<m>:CAN:FRAMe<n>:DATA?**

Returns the data of the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<Data> Comma-separated list of values. The first value is the number of bytes, followed by the values of the data bytes.

**Example:**
```
BUS1:CAN:FRAMe2:DATA?
--> 3,208,231,32
```
Returns the data of the second frame: the number of bytes is 3 data (first value).

**Usage:** Query only

---

**BUS<m>:CAN:FRAMe<n>:ACKState?**
**BUS<m>:CAN:FRAMe<n>:CSSTate?**
**BUS<m>:CAN:FRAMe<n>:DLCState?**
**BUS<m>:CAN:FRAMe<n>:IDSTate?**

Return the states of following parts of a message

- ACKState: state of acknowledgement field
- CSSTate: state of checksum field (CRC)
- DLCState: state of data length code
- IDSTate: identifier state

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<State> OK | ERRor | UNDF
UNDF: Undefined
*RST: OK

**Usage:** Query only

---

**BUS<m>:CAN:FRAMe<n>:ACKValue?**

Returns the value of the acknowledge slot for the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<AckValue> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and reset – are decimal values.
Range: 0 to 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:CAN:FRAMe<n>:BITRate?**

Returns the primary bit rate.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<PrimaryBitRate> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**BUS<m>:CAN:FRAMe<n>:CSValue?**

Returns the CRC sequence value of the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<ChecksumValue> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and reset – are decimal values.
Range: 0 to 2097151
*RST: 0

**Usage:** Query only

**BUS<m>:CAN:FRAMe<n>:DLCValue?**

Returns the data length code of the selected frame - the number of data bytes in the frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<FrameDLCValue> Number of data bytes in decimal values.
Range: 0 to 15
*RST: 0

**Usage:** Query only

**BUS<m>:CAN:FRAMe<n>:IDTYpe?**

Returns the identifier type of the selected frame, the identifier format of data and remote frames.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<IdentifierType> ANY | B11 | B29
B11: standard format, 11 bit
B29: extended format, 29 bit
*RST: B11

**Usage:** Query only

**BUS<m>:CAN:FRAMe<n>:IDValue?**

Returns the identifier value of the selected frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<IdentifierValue> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and reset – are decimal values.

Range: 0 to 536870911
*RST: 0

**Usage:** Query only

---

**BUS<m>:CAN:FRAMe<n>:BSEPosition?**

Returns the location of a bit stuffing error.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<BitStuffErrorPos> Time when the error occured

Range: 0 to 5000
*RST: 0

**Usage:** Query only

---

**BUS<m>:CAN:FRAMe<n>:FERCause?**

Returns information on a form error, if the frame status query returned a FORM error.

See also: `BUS<m>:CAN:FRAMe<n>:STATus?` on page 1723

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<FormErrorCause> NONE | CRCDerror | ACKDerror | FSBE | RESerror

CRCD = CRC delimiter error
ACKD = ACK delimiter error
FSBE = Fixed stuff bit error (CAN-FD ISO only)
RESE = Reserved bit error

*RST: NONE

**Usage:** Query only

---

**BUS<m>:CAN:FRAMe<n>:SDEXport?**

Returns the symbolic data of the frame in export format.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<SymbolicData>

**Usage:** Query only

---

**BUS<m>:CAN:FDATa:FRAMe<n>:SCValue?**

Returns the stuff bit count modulo 8 value.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<StuffCount> Range: 0 to 7
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:CAN:FRAMe<n>:BYTE<o>:STATe?**

Returns the state of the specified byte.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the byte number.

**Return values:**
<State> OK | ERRor | UNDF
UNDF: Undefined
*RST: OK

**Usage:** Query only

---

**BUS<m>:CAN:FRAMe<n>:BYTE<o>:VALue?**

Returns the value of the specified byte.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the byte number.

**Return values:**

<Value> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and reset – are decimal values.

Range: 0 to 255
*RST: 0

**Usage:** Query only

### 24.17.6.4 Search settings

SEARch:TRIGger:CAN[:SSOFrame] ... 1732
SEARch:TRIGger:CAN:SFTYpe ... 1732
SEARch:TRIGger:CAN:SFIDentifier ... 1732
SEARch:TRIGger:CAN:SIDData ... 1732
SEARch:TRIGger:CAN:SERRor ... 1733
SEARch:TRIGger:CAN:FDATa:STANdard ... 1733
SEARch:TRIGger:CAN:FTYPe ... 1733
SEARch:TRIGger:CAN:ITYPe ... 1734
SEARch:TRIGger:CAN:ICONdition ... 1734
SEARch:TRIGger:CAN:IMAX ... 1734
SEARch:TRIGger:CAN:IMIN ... 1735
SEARch:TRIGger:CAN:DCONdition ... 1735
SEARch:TRIGger:CAN:DMIN ... 1735
SEARch:TRIGger:CAN:DMAX ... 1736
SEARch:TRIGger:CAN:DLCCondition ... 1736
SEARch:TRIGger:CAN:DLC ... 1736
SEARch:RESult:CAN:FRAMe<m>:NDBYtes? ... 1737
SEARch:TRIGger:CAN:FDATa:DPOPerator ... 1737
SEARch:TRIGger:CAN:FDATa:DPOSition ... 1738
SEARch:TRIGger:CAN:FDATa:DPTO ... 1738
SEARch:TRIGger:CAN:ACKerror ... 1738
SEARch:TRIGger:CAN:BITSterror ... 1738
SEARch:TRIGger:CAN:CRCerror ... 1739
SEARch:TRIGger:CAN:FORMerror ... 1739
SEARch:TRIGger:CAN:FDATa:SCERror ... 1739
SEARch:TRIGger:CAN:FDATa[:FDF] ... 1740
SEARch:TRIGger:CAN:FDATa:BRS ... 1740
SEARch:TRIGger:CAN:FDATa:ESI ... 1740

**SEARch:TRIGger:CAN[:SSOFrame]** <SearchName>,<FrameStart>
**SEARch:TRIGger:CAN[:SSOFrame]?** <SearchName>

Enables the search for a start of frame.

**Parameters:**
<FrameStart> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search Criteria" on page 576

**SEARch:TRIGger:CAN:SFTYpe** <SearchName>,<FrameType>
**SEARch:TRIGger:CAN:SFTYpe?** <SearchName>

Enables the search for a specified frame type.

**Parameters:**
<FrameType> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search Criteria" on page 576

**SEARch:TRIGger:CAN:SFIDentifier** <SearchName>,<Identifier>
**SEARch:TRIGger:CAN:SFIDentifier?** <SearchName>

Enables the search for frame indentifier.

**Parameters:**
<Identifier> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search Criteria" on page 576

**SEARch:TRIGger:CAN:SIDData** <SearchName>,<IdentifierData>
**SEARch:TRIGger:CAN:SIDData?** <SearchName>

Enables the search for identifier and data.

**Parameters:**
<IdentifierData> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search Criteria" on page 576

---

**SEARch:TRIGger:CAN:SERRor** <SearchName>,<ErrorCondition>
**SEARch:TRIGger:CAN:SERRor?** <SearchName>

Enables the search for a specified error.

**Parameters:**
<ErrorCondition> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search Criteria" on page 576

---

**SEARch:TRIGger:CAN:FDATa:STANdard** <SearchName>,<Standard>
**SEARch:TRIGger:CAN:FDATa:STANdard?** <SearchName>

Selects the CAN standard: CAN, CAN FD, or Any.

Use "Any" to search on either CAN or CAN-FD frame. In this case, the search configuration provides all possible settings, for CAN as well as for CAN FD. The setting is available in CAN FD option R&S RTP-K9.

**Parameters:**
<Standard> ANY | CAN | CANFd
*RST: CAN

**Parameters for setting and query:**
<SearchName> String with the search name

**Manual operation:** See "Standard" on page 578

---

**SEARch:TRIGger:CAN:FTYPe** <SearchName>,<FrameType>
**SEARch:TRIGger:CAN:FTYPe?** <SearchName>

Selects the CAN frame type to be searched for.

For data and remote frames, the identifier format has to be set with `SEARch:TRIGger:CAN:ITYPe` on page 1734.

**Parameters:**
<FrameType> ANY | DATA | REMote | ERRor | OVERload
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame type" on page 578

**SEARch:TRIGger:CAN:ITYPe** <SearchName>,<IdentifierType>
**SEARch:TRIGger:CAN:ITYPe?** <SearchName>

Selects the format of data and remote frames: 11 bit for CAN base frames, or 29 bits for CAN extended frames.

**Parameters:**
<IdentifierType> ANY | B11 | B29
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "ID type" on page 578

**SEARch:TRIGger:CAN:ICONdition** <SearchName>,<IdOperator>
**SEARch:TRIGger:CAN:ICONdition?** <SearchName>

Sets the operator to set a specific identifier or an identifier range.

**Parameters:**
<IdOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These condition require one identifier pattern to be set with `SEARch:TRIGger:CAN:IMIN`.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `SEARch:TRIGger:CAN:IMIN` and `SEARch:TRIGger:CAN:IMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Identifier setup: Condition, Identifier min, Identifier max" on page 578

**SEARch:TRIGger:CAN:IMAX** <SearchName>,<IdPatternTo>
**SEARch:TRIGger:CAN:IMAX?** <SearchName>

Sets the the end value of an identifier range if `SEARch:TRIGger:CAN:ICONdition` is set to `INRange` or `OORange`.

**Parameters:**
<IdPatternTo> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Identifier setup: Condition, Identifier min, Identifier max" on page 578

---

**SEARch:TRIGger:CAN:IMIN** <SearchName>,<IdPattern>
**SEARch:TRIGger:CAN:IMIN?** <SearchName>

Specifies a message identifier pattern, or sets the the start value of an identifier range.

**Parameters:**
<IdPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Identifier setup: Condition, Identifier min, Identifier max" on page 578

---

**SEARch:TRIGger:CAN:DCONdition** <SearchName>,<DataOperator>
**SEARch:TRIGger:CAN:DCONdition?** <SearchName>

Sets the operator to set a specific data pattern or a data pattern range.

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with `SEARch:TRIGger:CAN:DMIN`.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `SEARch:TRIGger:CAN:DMIN` and `SEARch:TRIGger:CAN:DMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: DLC, NDB, Condition, Data min, Data max" on page 579

---

**SEARch:TRIGger:CAN:DMIN** <SearchName>,<DataPattern>
**SEARch:TRIGger:CAN:DMIN?** <SearchName>

Specifies a data pattern, or sets the the start value of a data pattern range.

**Parameters:**
<DataPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: DLC, NDB, Condition, Data min, Data max" on page 579

---

**SEARch:TRIGger:CAN:DMAX** <SearchName>,<DataPatternTo>
**SEARch:TRIGger:CAN:DMAX?** <SearchName>

Sets the the end value of an data range if `SEARch:TRIGger:CAN:DCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<DataPatternTo> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: DLC, NDB, Condition, Data min, Data max" on page 579

---

**SEARch:TRIGger:CAN:DLCCondition** <SearchName>,<DLCOperator>
**SEARch:TRIGger:CAN:DLCCondition?** <SearchName>

Operator to set the data length code for search.

**Parameters:**
<DLCOperator> EQUal | GETHan

*RST: GETHan

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: DLC, NDB, Condition, Data min, Data max" on page 579

---

**SEARch:TRIGger:CAN:DLC** <SearchName>,<WordCount>
**SEARch:TRIGger:CAN:DLC?** <SearchName>

Sets the Data Length Code, the number of data bytes to be found. For complete definition, set also the operator with `SEARch:TRIGger:CAN:DLCCondition`.

**Parameters:**
<WordCount> Range: 1 to 8
Increment: 1
*RST: 1

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: DLC, NDB, Condition, Data min, Data max" on page 579

---

**SEARch:RESult:CAN:FRAMe<m>:NDBYtes?** <SearchName>

Returns the number of data bytes defined by DLC. DLC and NDB are different in CAN FD for DLCs > 8.

See also: "Data setup: DLC, NDB, Transfer, Condition, Data min, Data max" on page 572.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<NDBytes> Range: 1 to 64
Increment: 1
*RST: 1

**Usage:** Query only

**Manual operation:** See "Data setup: DLC, NDB, Condition, Data min, Data max" on page 579

---

**SEARch:TRIGger:CAN:FDATa:DPOPerator** <SearchName>,<DataPosOperator>
**SEARch:TRIGger:CAN:FDATa:DPOPerator?** <SearchName>

Sets the operator for the data position if DLC ≥ 9. You can define an exact position, or a position range. The setting is available in CAN FD option R&S RTP-K9.

**Parameters:**
<DataPosOperator> ANY | OFF | EQUal | GETHan | INRange | RANGe

**ANY = OFF**
The data position is not relevant for the search.

**EQUal | GETHan**
Equal, Greater or equal than. These conditions require one data position to be set with `SEARch:TRIGger:CAN:FDATa:DPOSition`.

**INRange = RANGe**
In range: Set the minimum and maximum value of the range with `SEARch:TRIGger:CAN:FDATa:DPOSition` and `SEARch:TRIGger:CAN:FDATa:DPTO`.

*RST: ANY

**Parameters for setting and query:**
<SearchName> String with the search name

**Manual operation:** See "Data position" on page 579

**SEARch:TRIGger:CAN:FDATa:DPOSition** <SearchName>,<DataPosition>
**SEARch:TRIGger:CAN:FDATa:DPOSition?** <SearchName>

Defines the first possible start position of the data pattern. The setting is available in CAN FD option R&S RTP-K9.

**Parameters:**
<DataPosition> Range: 1 to 57
Increment: 1
*RST: 1

**Parameters for setting and query:**
<SearchName> String with the search name

**Manual operation:** See "Data position" on page 579

**SEARch:TRIGger:CAN:FDATa:DPTO** <SearchName>,<DataPositionTo>
**SEARch:TRIGger:CAN:FDATa:DPTO?** <SearchName>

Defines the last possible start position of the data pattern if the position operator `SEARch:TRIGger:CAN:FDATa:DPOPerator` defines a range. The setting is available in CAN FD option R&S RTP-K9.

**Parameters:**
<DataPositionTo> Range: 8 to 64
Increment: 1
*RST: 8

**Parameters for setting and query:**
<SearchName> String with the search name

**Manual operation:** See "Data position" on page 579

**SEARch:TRIGger:CAN:ACKerror** <SearchName>,<AckError>
**SEARch:TRIGger:CAN:ACKerror?** <SearchName>

Searches for acknowledgement errors. An acknowledgement error occurs when the transmitter does not receive an acknowledgment - a dominant bit during the Ack Slot.

**Parameters:**
<AckError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 579

**SEARch:TRIGger:CAN:BITSterror** <SearchName>,<BitStuffError>
**SEARch:TRIGger:CAN:BITSterror?** <SearchName>

Searches for bit stuffing errors.

**Parameters:**
<BitStuffError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 579

---

**SEARch:TRIGger:CAN:CRCerror** <SearchName>,<ChecksumError>
**SEARch:TRIGger:CAN:CRCerror?** <SearchName>

Searches for errors in the Cyclic Redundancy Check.

**Parameters:**
<ChecksumError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 579

---

**SEARch:TRIGger:CAN:FORMerror** <SearchName>,<FormError>
**SEARch:TRIGger:CAN:FORMerror?** <SearchName>

Searches for form errors. A form error occurs when a fixed-form bit field contains one or more illegal bits.

**Parameters:**
<FormError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 579

---

**SEARch:TRIGger:CAN:FDATa:SCERror** <SearchName>,<StuffCountError>
**SEARch:TRIGger:CAN:FDATa:SCERror?** <SearchName>

Triggers on stuff count errors. A stuff count error occurs if th received stuff count value does not match the value calculated from the own stuff bit count.

Only relevant for CAN FD signals in ISO standard.

**Parameters:**
<StuffCountError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 579

---

**SEARch:TRIGger:CAN:FDATa[:FDF]** <SearchName>,<FDFBit>
**SEARch:TRIGger:CAN:FDATa[:FDF]?** <SearchName>

Sets the EDL bit (extended data length), which determines whether a frame is CAN or CAN-FD. The setting is available in CAN FD option R&S RTP-K9.

**Parameters:**
<FDFBit> ONE | ZERO | DC

ONE: CAN FD.
ZERO: CAN.
DC: don't care, the format is not relevant.

*RST: DC

**Parameters for setting and query:**
<SearchName> String with the search name

**Manual operation:** See "FD bits" on page 579

---

**SEARch:TRIGger:CAN:FDATa:BRS** <SearchName>,<BRSBit>
**SEARch:TRIGger:CAN:FDATa:BRS?** <SearchName>

Sets the bit rate switching bit for identifier and identifier + data searches. The setting is available in CAN FD option R&S RTP-K9.

**Parameters:**
<BRSBit> ONE | ZERO | DC

DC: Don't care

*RST: ONE

**Parameters for setting and query:**
<SearchName> String with the search name

**Manual operation:** See "FD bits" on page 579

---

**SEARch:TRIGger:CAN:FDATa:ESI** <SearchName>,<ESIBit>
**SEARch:TRIGger:CAN:FDATa:ESI?** <SearchName>

Sets the error state indicator bit for identifier and identifier + data searches. The setting is available in CAN FD option R&S RTP-K9.

**Parameters:**
<ESIBit> ONE | ZERO | DC

*RST: DC

**Parameters for setting and query:**
<SearchName> String with the search name

**Manual operation:** See "FD bits" on page 579

### 24.17.6.5 Search results

The search on decoded CAN data returns the same results as the queries for decode results.

In search result commands, you have to specify the <SearchName> parameter. It is a string parameter that contains the search definition name.

For a description of the returned values, see the corresponding commands in Chapter 24.17.6.3, "Decode results", on page 1722.

SEARch:RESult:CAN:FCOunt? 1741
SEARch:RESult:CAN:FRAMe<m>:ACKValue? 1741
SEARch:RESult:CAN:FRAMe<m>:BSEPosition? 1742
SEARch:RESult:CAN:FRAMe<m>:BYTE<n>:STATe? 1742
SEARch:RESult:CAN:FRAMe<m>:BYTE<n>:VALue? 1742
SEARch:RESult:CAN:FRAMe<m>:ACKState? 1743
SEARch:RESult:CAN:FRAMe<m>:CSSTate? 1743
SEARch:RESult:CAN:FRAMe<m>:DLCState? 1743
SEARch:RESult:CAN:FRAMe<m>:IDSTate? 1743
SEARch:RESult:CAN:FRAMe<m>:CSValue? 1743
SEARch:RESult:CAN:FRAMe<m>:DATA? 1743
SEARch:RESult:CAN:FRAMe<m>:DLCValue? 1744
SEARch:RESult:CAN:FRAMe<m>:FERCause? 1744
SEARch:RESult:CAN:FRAMe<m>:IDTYpe? 1744
SEARch:RESult:CAN:FRAMe<m>:IDValue? 1744
SEARch:RESult:CAN:FDATa:FRAMe<m>:SCValue? 1745
SEARch:RESult:CAN:FDATa:FRAMe<m>:STANdard? 1745
SEARch:RESult:CAN:FRAMe<m>:STARt? 1745
SEARch:RESult:CAN:FRAMe<m>:STATus? 1746
SEARch:RESult:CAN:FRAMe<m>:STOP? 1746
SEARch:RESult:CAN:FRAMe<m>:SYMBol? 1746
SEARch:RESult:CAN:FRAMe<m>:TYPE? 1747

---

**SEARch:RESult:CAN:FCOunt?** <SearchName>

**Query parameters:**
<SearchName>

**Return values:**

| | | |
|---|---|---|
| <Count> | Range: | 0 to 100000 |
| | Increment: | 1 |
| | *RST: | 0 |

**Usage:** Query only

---

**SEARch:RESult:CAN:FRAMe<m>:ACKValue?** <SearchName>

**Suffix:**

| | |
|---|---|
| <m> | * |

**Query parameters:**
<SearchName>

**Return values:**
<AckValue> Range: 0 to 1
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:CAN:FRAMe<m>:BSEPosition?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<BitStuffErrorPos> Range: 0 to 5000
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:CAN:FRAMe<m>:BYTE<n>:STATe?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | ERRor | UNDF
*RST: OK

**Usage:** Query only

---

**SEARch:RESult:CAN:FRAMe<m>:BYTE<n>:VALue?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Value> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:CAN:FRAMe<m>:ACKState?** <SearchName>
**SEARch:RESult:CAN:FRAMe<m>:CSSTate?** <SearchName>
**SEARch:RESult:CAN:FRAMe<m>:DLCState?** <SearchName>
**SEARch:RESult:CAN:FRAMe<m>:IDSTate?** <SearchName>
**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<IdentifierState> OK | ERRor | UNDF
*RST: OK

**Usage:** Query only

---

**SEARch:RESult:CAN:FRAMe<m>:CSValue?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<ChecksumValue> Range: 0 to 2097151
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:CAN:FRAMe<m>:DATA?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

**SEARch:RESult:CAN:FRAMe<m>:DLCValue?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameDLCValue> Range: 0 to 15
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:CAN:FRAMe<m>:FERCause?** <SearchName>

Returns information on a form error, if the frame status query returned a FORM error.

See also: SEARch:RESult:CAN:FRAMe<m>:STATus? on page 1746.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FormErrorCause> NONE | CRCDerror | ACKDerror | FSBE | RESerror
See BUS<m>:CAN:FRAMe<n>:FERCause? on page 1729.
*RST: NONE

**Usage:** Query only

**SEARch:RESult:CAN:FRAMe<m>:IDTYpe?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<IdentifierType> ANY | B11 | B29
*RST: B11

**Usage:** Query only

**SEARch:RESult:CAN:FRAMe<m>:IDValue?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<IdentifierValue> Range: 0 to 536870911
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:CAN:FDATa:FRAMe<m>:SCValue?** <SearchName>

Returns the stuff bit count modulo 8.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<StuffCount> Range: 0 to 7
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:CAN:FDATa:FRAMe<m>:STANdard?** <SearchName>

Returns the CAN protocol standard: CAN or CAN FD.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Standard> CAN | CANFd
*RST: CAN

**Usage:** Query only

---

**SEARch:RESult:CAN:FRAMe<m>:STARt?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameStart> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:CAN:FRAMe<m>:STATus?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameState> OK | FORM | BTST | CRC | CRCD | NOACk | ACKD | EOFD | CAERror | FCERor | INSufficient | SERRror | SFERror | SCERror | SAERror | SCAE | SCFE

See BUS<m>:CAN:FRAMe<n>:STATus? on page 1723.

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:CAN:FRAMe<m>:STOP?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameStop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:CAN:FRAMe<m>:SYMBol?** <SearchName>

Returns the symbolic label of the specified frame if the label list is enabled.

**Suffix:**
<m> *
Selects the number of the frame in the current acquisition, 1...n.

**Query parameters:**
<SearchName> String parameter that contains the search definition name

**Return values:**
<Label> Symbolic label (string)

**Usage:** Query only

---

**SEARch:RESult:CAN:FRAMe<m>:TYPE?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameType> DATA | REMote | ERR | OVLD
*RST: DATA

**Usage:** Query only

### 24.17.6.6 Symbolic trigger, decode and search


- Symbolic trigger ..... 1747
- Symbolic decode results ..... 1749
- Symbolic search ..... 1749


#### Symbolic trigger


TRIGger<m>:CAN:SYMBolic:MSGValue ..... 1747
TRIGger<m>:CAN:SYMBolic:TSIGnals ..... 1747
TRIGger<m>:CAN:SYMBolic:SIGValue ..... 1748
TRIGger<m>:CAN:SYMBolic:DMAX ..... 1748
TRIGger<m>:CAN:SYMBolic:DMIN ..... 1748
TRIGger<m>:CAN:SYMBolic:SGEValue ..... 1748


---

**TRIGger<m>:CAN:SYMBolic:MSGValue** <MessageName>

Sets the message to be triggered or searched for.

The setting is used by symbolic trigger and symbolic search. Thus, you always trigger on and search for the same message if symbolic trigger and symbolic search is used at the same time.

See also: `SEARch:TRIGger:CAN:SYMBolic:MSGValue` on page 1750

**Parameters:**
<MessageName> String with the symbolic message name

**Manual operation:** See "Message" on page 581

---

**TRIGger<m>:CAN:SYMBolic:TSIGnals** <TriggerOnSignal>

Enables the trigger on a specific signal value that is part of the selected message.

**Parameters:**
<TriggerOnSignal> ON | OFF
*RST: OFF

**Manual operation:** See "Signal trigger" on page 582

---

**TRIGger<m>:CAN:SYMBolic:SIGValue** <SignalName>

Sets the signal name to be triggered or searched for.

The setting is used by symbolic trigger and symbolic search. Thus, you always trigger on and search for the same signal if symbolic trigger and symbolic search is used at the same time.

See also: `SEARch:TRIGger:CAN:SYMBolic:SIGValue` on page 1751

**Parameters:**
<SignalName> String with the signal name as defined in the DBC file.

**Manual operation:** See "Signal" on page 582

---

**TRIGger<m>:CAN:SYMBolic:DMAX** <DataPatternTo>

Defines the maximum data value of the signal.

This value is required to specify a range if condition `INRange | OORange` is set with `TRIGger<m>:CAN:DCONdition`.

**Parameters:**
<DataPatternTo> Decimal representation of the data pattern
Range: -100E+24 to 100E+24
Increment: 1
*RST: 1

**Manual operation:** See "Value max" on page 582

---

**TRIGger<m>:CAN:SYMBolic:DMIN** <DataPattern>

Defines the minimum data value of the signal.

To set the condition, use `TRIGger<m>:CAN:DCONdition`.

**Parameters:**
<DataPattern> Decimal representation of the data pattern
Range: -100E+24 to 100E+24
Increment: 1
*RST: 0

**Manual operation:** See "Value, Value min" on page 582

---

**TRIGger<m>:CAN:SYMBolic:SGEValue** <SignalEnumValue>

Sets a symbolic data value for signals with enumerated values.

**Parameters:**

<SignalEnumValue> Numeric value according to the value definition in the DBC file

**Example:** Definition line in DBC file:

```
VAL_ 2175091489 Gear 0 "Idle" 1 "Gear_1" 2 "Gear_2" 3 "Gear_3"
4 "Gear_4" 5 "Gear_5" ;
```

Search for "Gear_4"

```
TRIGger:CAN:SYMBolic:SGEValue 4
```

**Manual operation:** See "Value, Value min" on page 582

### Symbolic decode results

**BUS<m>:CAN:FRAMe<n>:SDATa?**

Returns the complete symbolic data of the selected frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<SymbolicData> String with comma-separated list of symbolic data

**Example:**

```
BUS:CAN:FRAMe9:SDATa?
<-- [sym] 325 kW, 0x0A, 423 N, 174 l, Running, 90 degC, 0x06, 437 rpm
```

Returns the symbolic results of the 9th frame.

**Usage:** Query only

### Symbolic search

SEARch:RESult:CAN:FRAMe<m>:SDATa? ... 1749
SEARch:TRIGger:CAN:SSYMbolic ... 1750
SEARch:TRIGger:CAN:SYMBolic:MSGValue ... 1750
SEARch:TRIGger:CAN:SYMBolic:SSIGnals ... 1750
SEARch:TRIGger:CAN:SYMBolic:SIGValue ... 1751
SEARch:TRIGger:CAN:SYMBolic:DMIN ... 1751
SEARch:TRIGger:CAN:SYMBolic:DMAX ... 1751
SEARch:TRIGger:CAN:SYMBolic:SGEValue ... 1752

**SEARch:RESult:CAN:FRAMe<m>:SDATa?** <SearchName>

Returns the symbolic data of the selected result frame.

**Suffix:**

<m> *
Sets the index of the search result frame.

**Query parameters:**
<SearchName> String that contains the search definition name

**Return values:**
<SymbolicData> String with comma-separated list of symbolic data

**Example:**
```
SEARch:RESult:CAN:FRAMe:SDATa? 'Search1'
<-- [sym] 325 kW, 0x0A, 423 N, 174 l, Running, 90 degC, 0x06, 437 rpm
```
Returns the symbolic results of the first search result.

**Usage:** Query only

---

**SEARch:TRIGger:CAN:SSYMbolic** <SearchName>,<CheckSymbolic>
**SEARch:TRIGger:CAN:SSYMbolic?** <SearchName>

Enables the symbolic search and disables all other search criteria.

**Parameters:**
<CheckSymbolic> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName> String that contains the search definition name

**Manual operation:** See "Search Criteria" on page 576

---

**SEARch:TRIGger:CAN:SYMBolic:MSGValue** <SearchName>, <MessageName>
**SEARch:TRIGger:CAN:SYMBolic:MSGValue?** <SearchName>

Sets the message to be triggered or searched for.

The setting is used by symbolic trigger and symbolic search. Thus, you always trigger on and search for the same message if symbolic trigger and symbolic search is used at the same time.

See also: `TRIGger<m>:CAN:SYMBolic:MSGValue` on page 1747

**Parameters:**
<MessageName> String that contains the symbolic message name

**Parameters for setting and query:**
<SearchName> String that contains the search definition name

**Example:**
```
SEARch:TRIGger:CAN:SYMBolic:MSGValue "Search1","EngineData"
```

**Manual operation:** See "Message" on page 584

---

**SEARch:TRIGger:CAN:SYMBolic:SSIGnals** <SearchName>,<SymbolicSearch>
**SEARch:TRIGger:CAN:SYMBolic:SSIGnals?** <SearchName>

Enables the search for symbolic values if DBC label list file is loaded and applied. Symbolic search disables all other search criteria.

**Parameters:**
<SymbolicSearch> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName> String that contains the search definition name

**Manual operation:** See "Search signals" on page 584

---

**SEARch:TRIGger:CAN:SYMBolic:SIGValue** <SearchName>, <SignalName>
**SEARch:TRIGger:CAN:SYMBolic:SIGValue?** <SearchName>

Sets the signal name to be triggered or searched for.

The setting is used by symbolic trigger and symbolic search. Thus, you always trigger on and search for the same signal if symbolic trigger and symbolic search is used at the same time.

See also: `TRIGger<m>:CAN:SYMBolic:SIGValue` on page 1748

**Parameters:**
<SignalName> String that contains the symbolic signal name

**Parameters for setting and query:**
<SearchName> String that contains the search definition name

**Example:**
```
SEARch:TRIGger:CAN:SYMBolic:SIGValue "Search1","EngForce"
```

**Manual operation:** See "Signal" on page 584

---

**SEARch:TRIGger:CAN:SYMBolic:DMIN** <SearchName>,<DataPattern>
**SEARch:TRIGger:CAN:SYMBolic:DMIN?** <SearchName>

Defines the minimum data pattern in a symbolic search.

To set the condition, use `SEARch:TRIGger:CAN:DCONdition`.

**Parameters:**
<DataPattern> Range: -100E+24 to 100E+24
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName> String that contains the search definition name

**Manual operation:** See "Value, Value (min)" on page 584

---

**SEARch:TRIGger:CAN:SYMBolic:DMAX** <SearchName>,<DataPatternTo>
**SEARch:TRIGger:CAN:SYMBolic:DMAX?** <SearchName>

Defines the maximum data pattern of the signal in a symbolic search.

This value is required to specify a range if condition `INRange | OORange` is set with `SEARch:TRIGger:CAN:DCONdition` on page 1735.

**Parameters:**
<DataPatternTo> Range: -100E+24 to 100E+24
Increment: 1
*RST: 1

**Parameters for setting and query:**
<SearchName> String that contains the search definition name

**Manual operation:** See "Value (max)" on page 584

---

**SEARch:TRIGger:CAN:SYMBolic:SGEValue** <SearchName>, <SignalEnumValue>
**SEARch:TRIGger:CAN:SYMBolic:SGEValue?** <SearchName>

Sets a symbolic data value for signals with enumerated values.

**Parameters:**
<SignalEnumValue> Numeric value according to the value definition in the DBC file

**Parameters for setting and query:**
<SearchName> String that contains the search definition name

**Example:** Definition line in DBC file:

```
VAL_ 2175091489 Gear 0 "Idle" 1 "Gear_1" 2 "Gear_2"
3 "Gear_3" 4 "Gear_4" 5 "Gear_5" ;
```

Search for "Gear_3"

```
SEARch:TRIGger:CAN:SYMBolic:SGEValue "Search1",3
```

**Manual operation:** See "Value, Value (min)" on page 584

## 24.17.7 LIN (option R&S RTP-K3)


- Configuration....1752
- Trigger....1755
- Decode results....1761
- LIN search settings....1769
- LIN search results....1777


### 24.17.7.1 Configuration


BUS<m>:LIN:DATA:SOURce....1752
BUS<m>:LIN:DATA:THReshold....1753
BUS<m>:LIN:TECHnology....1753
BUS<m>:LIN:BITRate....1754
BUS<m>:LIN:POLarity....1754
BUS<m>:LIN:STANdard....1754


---

**BUS<m>:LIN:DATA:SOURce** <DataSource>

Sets the waveform of the data line.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<DataSource> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

See Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 587

---

**BUS<m>:LIN:DATA:THReshold** <Threshold>

Sets a user-defined threshold value.

Alternatively, you can set the threshold according to the signal technology with `BUS<m>:LIN:TECHnology`.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Threshold>
Range: -12 to 12
Increment: 0.1
*RST: 0
Default unit: V

**Manual operation:** See "Threshold" on page 588

---

**BUS<m>:LIN:TECHnology** <Technology>

Sets the threshold voltage as defined for various signal technologies.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Technology> V15 | V25 | V35 | V6 | V9 | MANual

**V15**
1.5 Volt (TTL)

**V25**
2.5 Volt (CMOS 5.0 V)

**V35 | V6 | V9**
3.5 V (7 V supply), 6.0 V (12 V supply), 9.0 V (18 V supply) respectively

**MANual**
Manual setting of user-defined values with `BUS<m>:LIN:DATA:THReshold`.

*RST: V35

**Manual operation:** See "Threshold" on page 588

---

**BUS<m>:LIN:BITRate** <Bitrate>

Sets the number of transmitted bits per second.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Bitrate> Range: 1000 to 20000
Increment: 1
*RST: 9600
Default unit: bps

**Manual operation:** See "Bit rate" on page 588

---

**BUS<m>:LIN:POLarity** <Polarity>

Defines the idle state of the bus. The idle state is the rezessive state and corresponds to a logic 1.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Polarity> IDLLow | IDLHigh
*RST: IDLHigh

**Manual operation:** See "Polarity" on page 588

---

**BUS<m>:LIN:STANdard** <Standard>

Selects the version of the LIN standard.

See also: "LIN standard" on page 587

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Standard> V1X | V2X | J2602 | AUTO
*RST: AUTO

**Manual operation:** See "LIN standard" on page 587

### 24.17.7.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:LIN:TYPE....1755
TRIGger<m>:LIN:ICONdition....1756
TRIGger<m>:LIN:IMIN....1756
TRIGger<m>:LIN:IMAX....1757
TRIGger<m>:LIN:DCONdition....1757
TRIGger<m>:LIN:DMIN....1757
TRIGger<m>:LIN:DMAX....1757
TRIGger<m>:LIN:BORDer....1758
TRIGger<m>:LIN:DLECondition....1758
TRIGger<m>:LIN:DLENgth....1759
TRIGger<m>:LIN:IDOR<n>:ENABle....1759
TRIGger<m>:LIN:IDOR<n>[:VALue]....1759
TRIGger<m>:LIN:SYERror....1759
TRIGger<m>:LIN:IPERror....1760
TRIGger<m>:LIN:CHKSerror....1760
TRIGger<m>:LIN:ERRPattern....1760
TRIGger<m>:LIN:CRCDatalen....1761
TRIGger<m>:LIN:STANdard....1761

---

**TRIGger<m>:LIN:TYPE** <Type>

Selects the trigger type for LIN analysis.

See: "Trigger type" on page 592.

**Parameters:**

<Type> SYNC | ID | IDOR | IDDT | WKFR | ERRC

**SYNC**
Start of the frame, triggers on the stop bit of the sync field.

**ID**
Sets the trigger to one specific identifier or an identifier range. To set the identifier, use `TRIGger<m>:LIN:ICONdition`, `TRIGger<m>:LIN:IMIN` on page 1756, and `TRIGger<m>:LIN:IMAX` on page 1757.

**IDOR**
Triggers on an OR combination with up to four identifier conditions. For each identifier condition, enable it with `TRIGger<m>:LIN:IDOR<n>:ENABle` and set the value with `TRIGger<m>:LIN:IDOR<n>[:VALue]`

**IDDT**
Combination of identifier and data conditions
To set the identifier condition, use `TRIGger<m>:LIN:ICONdition`, `TRIGger<m>:LIN:IMIN`, and `TRIGger<m>:LIN:IMAX`.
To set the data condition, use `TRIGger<m>:LIN:DCONdition`, `TRIGger<m>:LIN:DMIN`, and `TRIGger<m>:LIN:DMAX`.

**WKFR**
Wakeup frame

**ERRC**
Error condition. Define the error types with `TRIGger<m>:LIN:CHKSerror` on page 1760, `TRIGger<m>:LIN:IPERror`, and `TRIGger<m>:LIN:SYERror`

*RST: SYNC

**Manual operation:** See "Trigger type" on page 592

---

**TRIGger<m>:LIN:ICONdition** <IdOperator>

Sets the operator to set a specific identifier or an identifier range.

**Parameters:**
<IdOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These condition require one identifier pattern to be set with `TRIGger<m>:LIN:IMIN`

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `TRIGger<m>:LIN:IMIN` and `TRIGger<m>:LIN:IMAX`

*RST: EQUal

**Manual operation:** See "Identifier setup: Condition, Frame ID min, Frame ID max" on page 593

---

**TRIGger<m>:LIN:IMIN** <IdPattern>

Specifies a slave identifier pattern, or sets the the start value of an identifier range.

**Parameters:**
<IdPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Manual operation:** See "Identifier setup: Condition, Frame ID min, Frame ID max" on page 593

**TRIGger<m>:LIN:IMAX** <IdPatternTo>

Sets the the end value of an identifier range if `TRIGger<m>:LIN:ICONdition` is set to `INRange` or `OORange`.

**Parameters:**

<IdPatternTo> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Manual operation:** See "Identifier setup: Condition, Frame ID min, Frame ID max" on page 593

**TRIGger<m>:LIN:DCONdition** <DataOperator>

Sets the operator to set a specific data pattern or a data pattern range.

**Parameters:**

<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with `TRIGger<m>:LIN:DMIN`.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `TRIGger<m>:LIN:DMIN` and `TRIGger<m>:LIN:DMAX`

*RST: EQUal

**Manual operation:** See "Data setup: Data length, Transfer, Condition, Data min, Data max" on page 594

**TRIGger<m>:LIN:DMIN** <DataPattern>

Specifies a data pattern, or sets the the start value of a data pattern range.

**Parameters:**

<DataPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data setup: Data length, Transfer, Condition, Data min, Data max" on page 594

**TRIGger<m>:LIN:DMAX** <DataPatternTo>

Sets the the end value of an data range if `TRIGger<m>:LIN:DCONdition` is set to `INRange` or `OORange`.

**Parameters:**

<DataPatternTo> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data setup: Data length, Transfer, Condition, Data min, Data max" on page 594

---

**TRIGger<m>:LIN:BORDer** <Endianness>

Sets the byte order (endianness) of the data transfer.

According to the standard, LIN data is transmitted in little endian transfer order.

**Parameters:**

<Endianness> BENDian | LENDian

**BENDian**
Big endian, data is analyzed and evaluated in the order of reception.

**LENDian**
Little endian, the instrument reads the complete data, reverses the byte order of the data, and compares it with the specified data word.

*RST: BENDian

**Manual operation:** See "Data setup: Data length, Transfer, Condition, Data min, Data max" on page 594

---

**TRIGger<m>:LIN:DLECondition** <DLCOperator>

Operator to set the data length for triggering on LIN data.

For Big Endian transfer direction, you can trigger on a number of bytes less than the data length of the frame, that means, on the first bytes that are transmitted. For Little Endian transfer direction, the exact number of data bytes in the frame must be set.

Example: The data word to be sent is *12 34 56*, and it is sent little endian by the LIN node. With Data length ≥ 2 and Transfer = Big endian, you trigger on the data of the first two bytes, that is *56 34*. With Data length = 3 and Transfer = Little endian, you trigger on the required data word *12 34 56*.

The number of data bytes to be found is set with `TRIGger<m>:LIN:DLENgth` on page 1759.

See also:`TRIGger<m>:LIN:BORDer` on page 1758 .

**Parameters:**

<DLCOperator> EQUal | GETHan

For little endian transfer direction, EQUal must be set.

*RST: GETHan

**Manual operation:** See "Data setup: Data length, Transfer, Condition, Data min, Data max" on page 594

**TRIGger<m>:LIN:DLENgth** <WordCount>

Sets the length of the bit pattern to be found, in bytes. For "Big Endian" transfer direction, you can trigger on a number of bytes less than the data length of the frame, that means, on the beginning of the data pattern. For "Little Endian" transfer direction, the exact number of data bytes in the frame must be set.

For complete definition, set also the operator with `TRIGger<m>:LIN:DLECondition` on page 1758.

**Parameters:**

<WordCount>
Range: 1 to 8
Increment: 1
*RST: 1

**Manual operation:** See "Data setup: Data length, Transfer, Condition, Data min, Data max" on page 594

**TRIGger<m>:LIN:IDOR<n>:ENABle** <UseIdentifier>

Includes the indicated IDOR address in the "identifier OR" trigger condition.

**Suffix:**

<n>
1..4
Index of the identifier in an "identifier OR" condition

**Parameters:**

<UseIdentifier>
ON | OFF
*RST: OFF

**Manual operation:** See "Identifier OR setup: Monitor, Frame ID" on page 594

**TRIGger<m>:LIN:IDOR<n>[:VALue]** <IdPattern>

Defines the pattern of the indicated IDOR identifier in the "identifier OR" trigger condition.

**Suffix:**

<n>
1..4
Index of the identifier in an "identifier OR" condition

**Parameters:**

<IdPattern>
Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The parameter accepts the bit value X (don't care).

**Manual operation:** See "Identifier OR setup: Monitor, Frame ID" on page 594

**TRIGger<m>:LIN:SYERror** <SyncError>

Triggers if a synchronization error occurs.

The trigger type has to be set before: `TRIGger<m>:LIN:TYPE` to `ERRC`.

**Parameters:**
<SyncError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 595

---

**TRIGger<m>:LIN:IPERror** <IdParityError>

Triggers if an error occurs in the identifier parity bits. These are the bits 6 and 7 of the identifier.

The trigger type has to be set before: `TRIGger<m>:LIN:TYPE` to `ERRC`.

**Parameters:**
<IdParityError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 595

---

**TRIGger<m>:LIN:CHKSerror** <ChecksumError>

Triggers on checksum errors according to the LIN standard set with `BUS<m>:LIN:STANdard`.

The trigger type has to be set before: `TRIGger<m>:LIN:TYPE` to `ERRC`.

The frame identifier must be set with `TRIGger<m>:LIN:ERRPattern` on page 1760 and the data length with `TRIGger<m>:LIN:CRCDatalen` on page 1761.

**Parameters:**
<ChecksumError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 595

---

**TRIGger<m>:LIN:ERRPattern** <ErrorPattern>

Sets the frame identifier to trigger on a checksum error with `TRIGger<m>:LIN:CHKSerror` on page 1760.

**Parameters:**
<ErrorPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.
Possible values depend on `TRIGger<m>:LIN:CRCDatalen`.
Defining don't care bits 'X' in the ERRP bit string resets CRCDatalen to 0. When CRCDatalen is different than 0, then all the bits in ERRP must be 1 or 0, and X bits are set to 0.

**Example:**

```
FORM:BPAT STRG
TRIG1:LIN:ERRP '1X0'
TRIG1:LIN:ERRP?
<-- 1X0XXX
TRIGger:LIN:CRCDatalen?
<-- 0
TRIGger:LIN:CRCDatalen 4
TRIG1:LIN:ERRP?
<-- 100000
TRIG1:LIN:ERRP '00x11'
TRIG1:LIN:ERRP?
<-- 00X11X
TRIGger:LIN:CRCDatalen?
<-- 0
```

**Manual operation:** See "Error conditions" on page 595

---

**TRIGger<m>:LIN:CRCDatalen** <CRCDataLength>

Sets the number of data bytes to trigger on CRC errors (`TRIGger<m>:LIN:TYPE` is set to ERRC and `TRIGger<m>:LIN:CHKSerror` is set ON.)

**Parameters:**

<CRCDataLength> Values ≠0 restrict allowed bit values in `TRIGger<m>:LIN:ERRPattern` to 0 and 1.

Range: 0 to 8
Increment: 1
*RST: 0

**Manual operation:** See "Error conditions" on page 595

---

**TRIGger<m>:LIN:STANdard** <LINStandard>

Sets the LIN standard to trigger on CRC errors (`TRIGger<m>:LIN:TYPE` is set to ERRC and `TRIGger<m>:LIN:CHKSerror` is set ON.)

See also: "LIN standard" on page 587.

**Parameters:**

<LINStandard> V1X | V2X | J2602 | AUTO

*RST: AUTO

**Manual operation:** See "Error conditions" on page 595

### 24.17.7.3 Decode results

To load and acivate a label list, use:

- `BUS<m>:NEWList` on page 1639
- `BUS<m>:SYMBols` on page 1639

BUS<m>:LIN:FCOunt?....1762
BUS<m>:LIN:FRAMe<n>:STATus?....1762
BUS<m>:LIN:FRAMe<n>:STARt?....1763
BUS<m>:LIN:FRAMe<n>:STOP?....1763
BUS<m>:LIN:FRAMe<n>:SDATa?....1763
BUS<m>:LIN:FRAMe<n>:SDEXport?....1763
BUS<m>:LIN:FRAMe<n>:SYMBol?....1764
BUS<m>:LIN:FRAMe<n>:VERSion?....1764
BUS<m>:LIN:FRAMe<n>:DATA?....1764
BUS<m>:LIN:FRAMe<n>:IDSTate?....1765
BUS<m>:LIN:FRAMe<n>:IDValue?....1765
BUS<m>:LIN:FRAMe<n>:IDPValue?....1766
BUS<m>:LIN:FRAMe<n>:SYSTate?....1766
BUS<m>:LIN:FRAMe<n>:CSSTate?....1767
BUS<m>:LIN:FRAMe<n>:CSValue?....1767
BUS<m>:LIN:FRAMe<n>:BITRate?....1767
BUS<m>:LIN:FRAMe<n>:BYTE<o>:STATe?....1768
BUS<m>:LIN:FRAMe<n>:BYTE<o>:VALue?....1768

### BUS<m>:LIN:FCOunt?

Returns the number of decoded frames.

**Suffix:**

<m> 1..4
Selects the serial bus.

**Return values:**

<Count> Total number of decoded frames.

**Usage:** Query only

### BUS<m>:LIN:FRAMe<n>:STATus?

Returns the overall state of the selected frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<FrameState> OK | UART | CHCKsum | VERS | LENer | SPERror | PRERror | SYERror | WAKeup | CPERror | INSufficient | INComplete

UART: at least one UART error occured. LIN uses UART words without parity bit.
CHCKsum: checksum error
VERS: the version of the LIN standard is not valid
LENer: unexpected length
SPERror: stop error

PRERror: parity error in identifier
SYERror: synchronization error
WAKeup: the frame is a wakeup frame
CPERror: parity error and checksum error
INSufficient: the frame is not completely contained in the acquisition. The decoded part of the frame is valid.
INComplete: the frame is missing some parts

*RST: OK

**Usage:** Query only

---

**BUS<m>:LIN:FRAMe<n>:STARt?**
**BUS<m>:LIN:FRAMe<n>:STOP?**

Returns the start time and stop time of the selected frame, respectively.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<Start>, <Stop> Range: -100E+24 to 100E+24
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:LIN:FRAMe<n>:SDATa?**

Returns the complete symbolic data of the frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<SymbolicData>

**Usage:** Query only

---

**BUS<m>:LIN:FRAMe<n>:SDEXport?**

Returns the symbolic data of the frame in export format.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<SymbolicData>

**Usage:** Query only

---

**BUS<m>:LIN:FRAMe<n>:SYMBol?**

Returns the symbolic label of the specified frame if the label list is enabled.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the number of the frame in the current acquisition, 1...n.

**Return values:**
<Label> String with symbolic name of the identifier

**Example:** `BUS:LIN:FRAMe2:SYMBol?`
Response: `Temperature`

**Usage:** Query only

---

**BUS<m>:LIN:FRAMe<n>:VERSion?**

Returns the version of the LIN standard for the specified frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<FrameVersion> V1X | V2X | UNK
UNK: Unknown
*RST: UNK

**Usage:** Query only

---

**BUS<m>:LIN:FRAMe<n>:DATA?**

Returns the data bytes of the specified frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<Data> Comma-separated list of integer values (N, D1, D2,…, DN). N is the number of bytes in the frame, and D1…DN are the values of the bytes.

**Example:**
```
BUS:LIN:FRAMe4:DATA?
<-- 4,118,39,71,123
```

**Usage:** Query only

---

**BUS<m>:LIN:FRAMe<n>:IDSTate?**

Returns the identifier state of the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<IdentifierState> OK | STERror | SPERror | PRERror | UVAL | NOEXists | INSufficient

STERror: start error
SPERror: stop error
PRERror: parity error
UVAL: unexpected value
NOEXists: byte does not exist
INSufficient: the frame is not completely contained in the acquisition. The decoded part of the frame is valid.

*RST: OK

**Usage:** Query only

---

**BUS<m>:LIN:FRAMe<n>:IDValue?**

Returns the identifier value of the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<IdentifierValue> To set the value format, use FORMat:BPATtern.
The values below – range, increment and reset – are decimal values.

Range: 0 to 63
*RST: 0

**Usage:** Query only

---

**BUS<m>:LIN:FRAMe<n>:IDPValue?**

Returns the value of the identifier parity bits of the selected frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<IdtfParityValue> To set the value format, use FORMat:BPATtern.
The values below – range, increment and reset – are decimal values.

Range: 0 to 3
*RST: 0

**Usage:** Query only

---

**BUS<m>:LIN:FRAMe<n>:SYSTate?**

Returns the state of the sync field for the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<SyncState> OK | STERror | SPERror | UVAL | NOEXists | INSufficient

STERror: start error
SPERror: stop error
UVAL: unexpected value
NOEXists: byte does not exist
INSufficient: the frame is not completely contained in the acquisition. The decoded part of the frame is valid.

*RST: OK

**Usage:** Query only

**BUS<m>:LIN:FRAMe<n>:CSSTate?**

Returns the checksum state of the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<ChecksumState> OK | STERror | SPERror | UVAL | NOEXists | INSufficient
STERror: start error
SPERror: stop error
UVAL: unexpected value
NOEXists: byte does not exist
INSufficient: the frame is not completely contained in the acquisition. The decoded part of the frame is valid.
*RST: OK

**Usage:** Query only

**BUS<m>:LIN:FRAMe<n>:CSValue?**

Returns the checksum value of the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<ChecksumValue> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and reset – are decimal values.
Range: 0 to 255
*RST: 0

**Usage:** Query only

**BUS<m>:LIN:FRAMe<n>:BITRate?**

Returns the primary bit rate.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<PrimaryBitRate> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**BUS<m>:LIN:FRAMe<n>:BYTE<o>:STATe?**

Returns the state of the specified byte.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the byte number.

**Return values:**

<ByteState> OK | STERror | SPERror | UVAL | NOEXists | INSufficient
STERror: start error
SPERror: stop error
UVAL: unexpected value
NOEXists: byte does not exist
INSufficient: the frame is not completely contained in the acquisition. The decoded part of the frame is valid.
*RST: OK

**Usage:** Query only

---

**BUS<m>:LIN:FRAMe<n>:BYTE<o>:VALue?**

Returns the value of the specified byte.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the byte.

**Return values:**

<ByteValue> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and reset – are decimal values.

Range: 0 to 255
*RST: 0

**Usage:** Query only

### 24.17.7.4 LIN search settings

In search setup commands, you have to specify the <SearchName> parameter. It is a string parameter that contains the search definition name. The commands are similar to LIN trigger commands.

SEARch:TRIGger:LIN:SSOFrame....1769
SEARch:TRIGger:LIN:SFIDentifier....1770
SEARch:TRIGger:LIN:IDENtifieror....1770
SEARch:TRIGger:LIN:SIDData....1770
SEARch:TRIGger:LIN:SERRor....1770
SEARch:TRIGger:LIN:WUFRame....1771
SEARch:TRIGger:LIN:ICONdition....1771
SEARch:TRIGger:LIN:IMIN....1771
SEARch:TRIGger:LIN:IMAX....1772
SEARch:TRIGger:LIN:IDOR<m>:ENABle....1772
SEARch:TRIGger:LIN:IDOR<m>[:VALue]....1772
SEARch:TRIGger:LIN:DCONdition....1773
SEARch:TRIGger:LIN:DMIN....1773
SEARch:TRIGger:LIN:DMAX....1773
SEARch:TRIGger:LIN:BORDer....1774
SEARch:TRIGger:LIN:DLECondition....1774
SEARch:TRIGger:LIN:DLENgth....1775
SEARch:TRIGger:LIN:IPERror....1775
SEARch:TRIGger:LIN:SYERror....1775
SEARch:TRIGger:LIN:CHKSerror....1776
SEARch:TRIGger:LIN:ERRPattern....1776
SEARch:TRIGger:LIN:CRCDatalen....1776
SEARch:TRIGger:LIN:STANdard....1777

---

**SEARch:TRIGger:LIN:SSOFrame** <SearchName>,<FrameStart>
**SEARch:TRIGger:LIN:SSOFrame?** <SearchName>

Enables the search for the stop bit of the sync field, which marks the frame start.

**Parameters:**

<FrameStart> ON | OFF
*RST: OFF

**Parameters for setting and query:**

<SearchName>

**Manual operation:** See "Search types" on page 597

---

**SEARch:TRIGger:LIN:SFIDentifier** <SearchName>,<Identifier>
**SEARch:TRIGger:LIN:SFIDentifier?** <SearchName>

Enables the search for one specific identifier or an identifier range.

**Parameters:**
<Identifier> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search types" on page 597

---

**SEARch:TRIGger:LIN:IDENtifieror** <SearchName>,<IdentifierOr>
**SEARch:TRIGger:LIN:IDENtifieror?** <SearchName>

Enables the search for one to four address conditions.

**Parameters:**
<IdentifierOr> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search types" on page 597

---

**SEARch:TRIGger:LIN:SIDData** <SearchName>,<IdentifierData>
**SEARch:TRIGger:LIN:SIDData?** <SearchName>

Enables the search for a combination of identifier and data conditions.

**Parameters:**
<IdentifierData> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search types" on page 597

---

**SEARch:TRIGger:LIN:SERRor** <SearchName>,<ErrorCondition>
**SEARch:TRIGger:LIN:SERRor?** <SearchName>

Enables the search for various errors in the frame.

**Parameters:**
<ErrorCondition> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search types" on page 597

---

**SEARch:TRIGger:LIN:WUFRame** <SearchName>,<WakeUpFrame>
**SEARch:TRIGger:LIN:WUFRame?** <SearchName>

Enables the search for wakeup frames.

**Parameters:**
<WakeUpFrame> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search types" on page 597

---

**SEARch:TRIGger:LIN:ICONdition** <SearchName>,<IdOperator>
**SEARch:TRIGger:LIN:ICONdition?** <SearchName>

Sets the operator to define a specific identifier or an identifier range.

**Parameters:**
<IdOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These operators require one identifier pattern to be set with `SEARch:TRIGger:LIN:IMIN`

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `SEARch:TRIGger:LIN:IMIN` and `SEARch:TRIGger:LIN:IMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Identifier setup: Condition, Frame ID min, Frame ID max" on page 598

---

**SEARch:TRIGger:LIN:IMIN** <SearchName>,<IdPattern>
**SEARch:TRIGger:LIN:IMIN?** <SearchName>

Specifies a slave identifier pattern, or sets the the start value of an identifier range.

**Parameters:**
<IdPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Identifier setup: Condition, Frame ID min, Frame ID max" on page 598

---

**SEARch:TRIGger:LIN:IMAX** <SearchName>,<IdPatternTo>
**SEARch:TRIGger:LIN:IMAX?** <SearchName>

Sets the the end value of an identifier range if `SEARch:TRIGger:LIN:ICONdition` is set to `INRange` or `OORange`.

**Parameters:**
<IdPatternTo> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Identifier setup: Condition, Frame ID min, Frame ID max" on page 598

---

**SEARch:TRIGger:LIN:IDOR<m>:ENABle** <SearchName>,<UseIdentifier>
**SEARch:TRIGger:LIN:IDOR<m>:ENABle?** <SearchName>

Includes the indicated IDOR address in the "identifier OR" search.

**Suffix:**
<m> 1..4
Index of the identifier in an "identifier OR" condition

**Parameters:**
<UseIdentifier> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Identifier OR setup: Monitor, Frame ID" on page 598

---

**SEARch:TRIGger:LIN:IDOR<m>[:VALue]** <SearchName>,<IdPattern>
**SEARch:TRIGger:LIN:IDOR<m>[:VALue]?** <SearchName>

Defines the pattern of the indicated IDOR identifier in the "identifier OR" trigger condition.

**Suffix:**
<m> 1..4
Index of the identifier in an "identifier OR" condition

**Parameters:**
<IdPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Identifier OR setup: Monitor, Frame ID" on page 598

---

**SEARch:TRIGger:LIN:DCONdition** <SearchName>,<DataOperator>
**SEARch:TRIGger:LIN:DCONdition?** <SearchName>

Sets the operator to set a specific data pattern or a data pattern range.

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with `SEARch:TRIGger:LIN:DMIN`.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `SEARch:TRIGger:LIN:DMIN` and `SEARch:TRIGger:LIN:DMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: Condition, Data min, Data max, Data length, Transfer" on page 598

---

**SEARch:TRIGger:LIN:DMIN** <SearchName>,<DataPattern>
**SEARch:TRIGger:LIN:DMIN?** <SearchName>

Specifies a data pattern, or sets the the start value of a data pattern range.

**Parameters:**
<DataPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: Condition, Data min, Data max, Data length, Transfer" on page 598

---

**SEARch:TRIGger:LIN:DMAX** <SearchName>,<DataPatternTo>
**SEARch:TRIGger:LIN:DMAX?** <SearchName>

Sets the the end value of an identifier range if `SEARch:TRIGger:LIN:DCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<DataPatternTo> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: Condition, Data min, Data max, Data length, Transfer" on page 598

---

**SEARch:TRIGger:LIN:BORDer** <SearchName>,<Endianness>
**SEARch:TRIGger:LIN:BORDer?** <SearchName>

Sets the byte order (endianness) of the data transfer.

**Parameters:**
<Endianness> BENDian | LENDian

**BENDian**
Big endian, data is analyzed and evaluated in the order of reception.

**LENDian**
Little endian, the instrument reads the complete data, reverses the byte order of the data, and compares it with the specified data word.

*RST: BENDian

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: Condition, Data min, Data max, Data length, Transfer" on page 598

---

**SEARch:TRIGger:LIN:DLECondition** <SearchName>,<DLCOperator>
**SEARch:TRIGger:LIN:DLECondition?** <SearchName>

Operator to set the data length for search on LIN data.

For Big Endian transfer direction, you can trigger on a number of bytes less than the data length of the frame, that means, on the first bytes that are transmitted. For Little Endian transfer direction, the exact number of data bytes in the frame must be set.

Example: The data word to be sent is *12 34 56*, and it is sent little endian by the LIN node. With Data length ≥ 2 and Transfer = Big endian, you trigger on the data of the first two bytes, that is *56 34*. With Data length = 3 and Transfer = Little endian, you trigger on the required data word *12 34 56*.

The number of data bytes to be found is set with `SEARch:TRIGger:LIN:DLENgth`.

See also: `SEARch:TRIGger:LIN:BORDer` on page 1774.

**Parameters:**
<DLCOperator> EQUal | GETHan
For little endian transfer direction, EQUal must be set.
*RST: GETHan

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: Condition, Data min, Data max, Data length, Transfer" on page 598

---

**SEARch:TRIGger:LIN:DLENgth** <SearchName>,<WordCount>
**SEARch:TRIGger:LIN:DLENgth?** <SearchName>

Sets the length of the bit pattern to be found, in bytes.

For complete definition, set the operator using `SEARch:TRIGger:LIN:DLECondition`, and the transfer direction with `SEARch:TRIGger:LIN:BORDer`.

**Parameters:**
<WordCount> Range: 1 to 8
Increment: 1
*RST: 1

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: Condition, Data min, Data max, Data length, Transfer" on page 598

---

**SEARch:TRIGger:LIN:IPERror** <SearchName>,<IdParityError>
**SEARch:TRIGger:LIN:IPERror?** <SearchName>

Searches for errors in the identifier parity bits. These are the bits 6 and 7 of the identifier.

**Parameters:**
<IdParityError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 599

---

**SEARch:TRIGger:LIN:SYERror** <SearchName>,<SyncError>
**SEARch:TRIGger:LIN:SYERror?** <SearchName>

Searches for synchronization errors.

**Parameters:**
<SyncError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 599

---

**SEARch:TRIGger:LIN:CHKSerror** <SearchName>,<ChecksumError>
**SEARch:TRIGger:LIN:CHKSerror?** <SearchName>

Searches for checksum errors according to the LIN standard.

Use the following commands to configure the checksum error search:

- `SEARch:TRIGger:LIN:ERRPattern` on page 1776
- `SEARch:TRIGger:LIN:CRCDatalen` on page 1776
- `SEARch:TRIGger:LIN:STANdard` on page 1777

**Parameters:**
<ChecksumError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 599

---

**SEARch:TRIGger:LIN:ERRPattern** <SearchName>,<ErrorPattern>
**SEARch:TRIGger:LIN:ERRPattern?** <SearchName>

Sets the frame identifier to search for a checksum error.

**Parameters:**
<ErrorPattern> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 599

---

**SEARch:TRIGger:LIN:CRCDatalen** <SearchName>,<CRCDataLength>
**SEARch:TRIGger:LIN:CRCDatalen?** <SearchName>

Sets the number of data bytes search for CRC errors.

**Parameters:**
<CRCDataLength> Range: 0 to 8
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 599

---

**SEARch:TRIGger:LIN:STANdard** <SearchName>,<LINStandard>
**SEARch:TRIGger:LIN:STANdard?** <SearchName>

Sets the LIN standard to search for CRC errors.

**Parameters:**
<LINStandard> V1X | V2X | J2602 | AUTO
*RST: AUTO

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 599

### 24.17.7.5 LIN search results

The search on decoded LIN data returns the same results as the queries for decode results.

In search result commands, you have to specify the <SearchName> parameter. It is a string parameter that contains the search definition name.

For a description of the returned values, see the corresponding commands in Chapter 24.17.7.3, "Decode results", on page 1761.

SEARch:RESult:LIN:FCOunt? 1777
SEARch:RESult:LIN:FRAMe<m>:STATus? 1778
SEARch:RESult:LIN:FRAMe<m>:STARt? 1778
SEARch:RESult:LIN:FRAMe<m>:STOP? 1778
SEARch:RESult:LIN:FRAMe<m>:DATA? 1779
SEARch:RESult:LIN:FRAMe<m>:SDATa? 1779
SEARch:RESult:LIN:FRAMe<m>:SDEXport? 1779
SEARch:RESult:LIN:FRAMe<m>:CSSTate? 1779
SEARch:RESult:LIN:FRAMe<m>:CSValue? 1780
SEARch:RESult:LIN:FRAMe<m>:IDSTate? 1780
SEARch:RESult:LIN:FRAMe<m>:IDValue? 1780
SEARch:RESult:LIN:FRAMe<m>:IDPValue? 1781
SEARch:RESult:LIN:FRAMe<m>:SYMBol? 1781
SEARch:RESult:LIN:FRAMe<m>:SYSTate? 1781
SEARch:RESult:LIN:FRAMe<m>:VERSion? 1782
SEARch:RESult:LIN:FRAMe<m>:BYTE<n>:STATe? 1782
SEARch:RESult:LIN:FRAMe<m>:BYTE<n>:VALue? 1782

---

**SEARch:RESult:LIN:FCOunt?** <SearchName>

**Query parameters:**
<SearchName>

**Return values:**
<Count> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:LIN:FRAMe<m>:STATus?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | UART | CHCKsum | VERS | LENer | SPERror | PRERror | SYERror | WAKeup | CPERror | INSufficient | INComplete
*RST: OK

**Usage:** Query only

---

**SEARch:RESult:LIN:FRAMe<m>:STARt?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:LIN:FRAMe<m>:STOP?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:LIN:FRAMe<m>:DATA?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data> Data bytes in the Frame

**Usage:** Query only

**SEARch:RESult:LIN:FRAMe<m>:SDATa?** <SearchName>

Returns the symbolic data of the frame.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<SymbolicData>

**Usage:** Query only

**SEARch:RESult:LIN:FRAMe<m>:SDEXport?** <SearchName>

Returns the symbolic data of the frame for the export format.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<SymbolicData>

**Usage:** Query only

**SEARch:RESult:LIN:FRAMe<m>:CSSTate?** <SearchName>

Returns the status of the frame checksum.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<ChecksumState> OK | STERror | SPERror | PRERror | UVAL | NOEXists | INSufficient

STERror: StarT ERror, incorrect start bit
SPERror: StoP ERror, incorrect stop bit
PRERror:PaRity ERror, incorrect parity bit.
UVAL: unexpected value
NOEXists: byte does not exist

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:LIN:FRAMe<m>:CSValue?** <SearchName>

Returns the checksum value.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<ChecksumValue> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:LIN:FRAMe<m>:IDSTate?** <SearchName>

Returns the status of the identifier.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<IdentifierState> OK | STERror | SPERror | PRERror | UVAL | NOEXists | INSufficient

STERror: StarT ERror, incorrect start bit
SPERror: StoP ERror, incorrect stop bit
PRERror:PaRity ERror, incorrect parity bit.
UVAL: unexpected value
NOEXists: byte does not exist

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:LIN:FRAMe<m>:IDValue?** <SearchName>

Returns the value of the identifier.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<IdentifierValue> Range: 0 to 63
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:LIN:FRAMe<m>:IDPValue?** <SearchName>

Returns the value of the identifier parity bit.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<IdParityValue> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:LIN:FRAMe<m>:SYMBol?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Label>

**Usage:** Query only

---

**SEARch:RESult:LIN:FRAMe<m>:SYSTate?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<SyncState> OK | STERror | SPERror | PRERror | UVAL | NOEXists | INSufficient
*RST: OK

**Usage:** Query only

---

**SEARch:RESult:LIN:FRAMe<m>:VERSion?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Version> V1X | V2X | UNK
*RST: UNK

**Usage:** Query only

---

**SEARch:RESult:LIN:FRAMe<m>:BYTE<n>:STATe?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<ByteState> OK | STERror | SPERror | PRERror | UVAL | NOEXists | INSufficient
*RST: OK

**Usage:** Query only

---

**SEARch:RESult:LIN:FRAMe<m>:BYTE<n>:VALue?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<ByteValue> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

## 24.17.8 MIL-1553 (option R&S RTP-K6)


- Configuration....1783
- Trigger....1786
- Decode results....1797
- Search settings....1800
- Search results....1807


### 24.17.8.1 Configuration


BUS<m>:MILStd:SOURce....1783
BUS<m>:MILStd:MAXResponse:BITS....1783
BUS<m>:MILStd:MAXResponse:SELect....1784
BUS<m>:MILStd:MINGap:SELect....1784
BUS<m>:MILStd:MINGap:BITS....1784
BUS<m>:MILStd:POLarity....1784
BUS<m>:MILStd:PRESet....1785
BUS<m>:MILStd:THReshold:HIGH....1785
BUS<m>:MILStd:THReshold:LOW....1785


---

**BUS<m>:MILStd:SOURce** <SourceData>

Sets the channel for the signal source.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<SourceData> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

*RST: C1W1

**Manual operation:** See "Source" on page 602

---

**BUS<m>:MILStd:MAXResponse:BITS** <MaxResponseTime>

Sets the value for the maximum response time.

See also: `BUS<m>:MILStd:MAXResponse:SELect`.

**Suffix:**
<m> 1..4

**Parameters:**
<MaxResponseTime> Range: 2E-6 to 262E-6
Increment: 100E-9
*RST: 14E-6
Default unit: s

**Manual operation:** See "Check max response" on page 603

**BUS<m>:MILStd:MAXResponse:SELect** <MaxRespSel>

Enables the detection of the maximum response time between the last bit of a word and the following status word sync during decoding.

To specify the maximum response time, use `BUS<m>:MILStd:MAXResponse:BITS` on page 1783.

**Suffix:**
<m> 1..4

**Parameters:**
<MaxRespSelect> ON | OFF
*RST: ON

**Manual operation:** See "Check max response" on page 603

**BUS<m>:MILStd:MINGap:SELect** <MinGapSelect>

Enables the detection of the minimum idle time between the last bit of a message and the following command word sync during decoding.

To specify the minimum gap, use `BUS<m>:MILStd:MINGap:BITS`.

**Suffix:**
<m> 1..4

**Parameters:**
<MinGapSelect> ON | OFF
*RST: ON

**Manual operation:** See "Check min gap" on page 603

**BUS<m>:MILStd:MINGap:BITS** <MinGapTime>

Sets a value for the minimum gap.

See also: `BUS<m>:MILStd:MINGap:SELect`.

**Suffix:**
<m> 1..4

**Parameters:**
<MinGapTime> Range: 2E-6 to 262E-6
Increment: 100E-9
*RST: 4E-6
Default unit: s

**Manual operation:** See "Check min gap" on page 603

**BUS<m>:MILStd:POLarity** <Polarity>

Selects the wire on which the bus signal is measured.

**Suffix:**
<m> 1..4

**Parameters:**
<Polarity> NORMal | INVerted
*RST: NORMal

**Manual operation:** See "Polarity" on page 602

---

**BUS<m>:MILStd:PRESet** <Preset>

Sets the default threshold voltage.

**Suffix:**
<m> 1..4

**Parameters:**
<Preset> V05 | V2 | V5 | V7 | MANual
*RST: V5

**Manual operation:** See "Thresholds" on page 603

---

**BUS<m>:MILStd:THReshold:HIGH** <ThresholdHigh>

Sets the lower threshold level of the signal.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdHigh> Range: 0 to 14
Increment: 0.1
*RST: 5
Default unit: V

**Manual operation:** See "Thresholds" on page 603

---

**BUS<m>:MILStd:THReshold:LOW** <ThresholdLow>

Sets the lower threshold level of the signal.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdLow> Range: -14 to 0
Increment: 0.1
*RST: -5
Default unit: V

**Manual operation:** See "Thresholds" on page 603

### 24.17.8.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:MILStd:TYPE ..... 1787
TRIGger<m>:MILStd:DATA:RCONdition ..... 1787
TRIGger<m>:MILStd:CMD:RCONdition ..... 1787
TRIGger<m>:MILStd:CDST:RCONdition ..... 1787
TRIGger<m>:MILStd:DATA:RMIN ..... 1788
TRIGger<m>:MILStd:CMD:RMIN ..... 1788
TRIGger<m>:MILStd:CDST:RMIN ..... 1788
TRIGger<m>:MILStd:DATA:RMAX ..... 1788
TRIGger<m>:MILStd:CMD:RMAX ..... 1788
TRIGger<m>:MILStd:CDST:RMAX ..... 1788
TRIGger<m>:MILStd:CMD:CCONdition ..... 1788
TRIGger<m>:MILStd:CMD:CMAX ..... 1789
TRIGger<m>:MILStd:CMD:CMIN ..... 1789
TRIGger<m>:MILStd:CMD:SCONdition ..... 1789
TRIGger<m>:MILStd:CMD:SMAX ..... 1790
TRIGger<m>:MILStd:CMD:SMIN ..... 1790
TRIGger<m>:MILStd:CMD:TR ..... 1790
TRIGger<m>:MILStd:CDST:ICONdition ..... 1791
TRIGger<m>:MILStd:CDST:IMAX ..... 1791
TRIGger<m>:MILStd:CDST:IMIN ..... 1791
TRIGger<m>:MILStd:DATA:DCONdition ..... 1791
TRIGger<m>:MILStd:DATA:DMAX ..... 1792
TRIGger<m>:MILStd:DATA:DMIN ..... 1792
TRIGger<m>:MILStd:DATA:ICONdition ..... 1792
TRIGger<m>:MILStd:DATA:IMAX ..... 1793
TRIGger<m>:MILStd:DATA:IMIN ..... 1793
TRIGger<m>:MILStd:ERRor:MANChester ..... 1793
TRIGger<m>:MILStd:ERRor:PARity ..... 1793
TRIGger<m>:MILStd:ERRor:SYNC ..... 1794
TRIGger<m>:MILStd:MAXResponse:BITS ..... 1794
TRIGger<m>:MILStd:MAXResponse:SELect ..... 1794
TRIGger<m>:MILStd:MINGap:BITS ..... 1794
TRIGger<m>:MILStd:MINGap:SELect ..... 1795
TRIGger<m>:MILStd:STATus:BCReceived ..... 1795
TRIGger<m>:MILStd:STATus:BUSY ..... 1795
TRIGger<m>:MILStd:STATus:DBCaccept ..... 1795
TRIGger<m>:MILStd:STATus:INSTrument ..... 1795
TRIGger<m>:MILStd:STATus:MERRor ..... 1796
TRIGger<m>:MILStd:STATus:SREQuest ..... 1796

TRIGger<m>:MILStd:STATus:SUBSystem....1796
TRIGger<m>:MILStd:STATus:TERMinal....1796
TRIGger<m>:MILStd:TPSPecifier....1797

**TRIGger<m>:MILStd:TYPE** <Type>

Sets the trigger type for MIL-1553 analysis.

**Parameters:**

<Type> STYPe | WTYPe | DATA | CDST | CMD | STATword | ERR

**STYPe**
SyncTYPe: triggers on a sync impulse.

**WTYPe**
WordTYPe: triggers on the selected word type.

**DATA**
Triggers on a data word that can be specified.

**CDST**
CommanDSTatus word: triggers on a command word or on a status word that can be specified.

**CMD**
CoMmanD word: triggers on a command word or on a status word that can be specified.

**STATword**
STATus word: triggers on a status word that can be specified.

**ERR**
ERRor: triggers on any combination of protocol errors.

*RST: STYPe

**Manual operation:** See "Trigger Type" on page 605

**TRIGger<m>:MILStd:DATA:RCONdition** <RTAOperator>
**TRIGger<m>:MILStd:CMD:RCONdition** <RTAOperator>
**TRIGger<m>:MILStd:CDST:RCONdition** <RTAOperator>

Sets the operator to define a remote terminal address:

- DATA: for data words
- CMD: for command words
- CDST: for status words

**Parameters:**

<RTAOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with `TRIGger<m>:MILStd:CDST:RMIN`.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MILStd:CDST:RMIN` and `TRIGger<m>:MILStd:CDST:RMAX`.

*RST: EQUal

**Manual operation:** See "Remote Terminal Address" on page 607

---

**TRIGger<m>:MILStd:DATA:RMIN** <RTAPatternMin>
**TRIGger<m>:MILStd:CMD:RMIN** <RTAPatternMin>
**TRIGger<m>:MILStd:CDST:RMIN** <RTAPatternMin>

Specify a remote terminal address or set the the start value of a remote terminal address range:

- DATA: for data words
- CMD: for command words
- CDST: for status words

**Parameters:**
<RTAPatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Remote Terminal Address" on page 607

---

**TRIGger<m>:MILStd:DATA:RMAX** <RTAPatternMax>
**TRIGger<m>:MILStd:CMD:RMAX** <RTAPatternMax>
**TRIGger<m>:MILStd:CDST:RMAX** <RTAPatternMax>

Set the end value of a data range if `TRIGger<m>:MILStd:CDST:RCONdition` is set to `INRange` or `OORange`:

- DATA: for data words
- CMD: for command words
- CDST: for status words

**Parameters:**
<RTAPatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Remote Terminal Address" on page 607

---

**TRIGger<m>:MILStd:CMD:CCONdition** <WordCntOperator>

Sets the operator to set a specific data word count or mode code pattern.

**Parameters:**
<WordCntOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with `TRIGger<m>:MILStd:CMD:CMIN`.

**INRange | OORange**
In range/Out of range: Set the minimum and maximum value of the range with `TRIGger<m>:MILStd:CMD:CMIN` and `TRIGger<m>:MILStd:CMD:CMAX`.

*RST: EQUal

**Manual operation:** See "Data Word Count/Mode Code" on page 609

---

**TRIGger<m>:MILStd:CMD:CMAX** <WordCntPattMax>

Sets the end value of a data word count/mode code pattern if `TRIGger<m>:MILStd:CMD:CCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<WordCntPattMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data Word Count/Mode Code" on page 609

---

**TRIGger<m>:MILStd:CMD:CMIN** <WordCntPattMin>

Specifies a data word count/mode code pattern or sets the the start value of a pattern range.

**Parameters:**
<WordCntPattMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data Word Count/Mode Code" on page 609

---

**TRIGger<m>:MILStd:CMD:SCONdition** <SubaddrOperator>

Sets the operator to set a specific subaddress/mode pattern.

**Parameters:**
<SubaddrOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with`TRIGger<m>:MILStd:CMD:SMIN`.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MILStd:CMD:SMIN` and `TRIGger<m>:MILStd:CMD:SMAX`.

*RST: EQUal

**Manual operation:** See "Subaddress/ Mode" on page 609

---

**TRIGger<m>:MILStd:CMD:SMAX** <SubaddrPattMax>

Sets the end value of the subaddress range if `TRIGger<m>:MILStd:CMD:SCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<SubaddrPattMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Subaddress/ Mode" on page 609

---

**TRIGger<m>:MILStd:CMD:SMIN** <SubaddrPattMin>

Specifies a subaddress or sets the the start value of a subaddress range.

**Parameters:**
<SubaddrPattMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Subaddress/ Mode" on page 609

---

**TRIGger<m>:MILStd:CMD:TR** <TRFlag>

Triggers on a transmission mode.

**Parameters:**
<TRFlag> ONE | ZERO | DC

**ONE**
Transmit direction.

**ZERO**
Receive direction.

**DC**
Either directions.

*RST: DC

**Manual operation:** See "T/R (Transmit/receive)" on page 610

**TRIGger<m>:MILStd:CDST:ICONdition** <InfoOperator>

Sets the operator to set a specific info for the 9th to 19th bit of a command or status word.

**Parameters:**

<InfoOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with `TRIGger<m>:MILStd:CDST:IMIN`.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MILStd:CDST:IMIN` and `TRIGger<m>:MILStd:CDST:IMAX`.

*RST: EQUal

**Manual operation:** See "11-Bit Information" on page 609

**TRIGger<m>:MILStd:CDST:IMAX** <InfoPatternMax>

Sets the end value of the info range if `TRIGger<m>:MILStd:CDST:ICONdition` is set to `INRange`.

**Parameters:**

<InfoPatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "11-Bit Information" on page 609

**TRIGger<m>:MILStd:CDST:IMIN** <InfoPatternMin>

Specifies an info or sets the the start value of an info range.

**Parameters:**

<InfoPatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "11-Bit Information" on page 609

**TRIGger<m>:MILStd:DATA:DCONdition** <DataOperator>

Sets the operator to set a specific data pattern.

**Parameters:**

<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with `TRIGger<m>:MILStd:DATA:DMIN`.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MILStd:DATA:DMIN` and `TRIGger<m>:MILStd:DATA:DMAX`.

*RST: EQUal

**Manual operation:** See "Data Pattern" on page 608

---

**TRIGger<m>:MILStd:DATA:DMAX** <DataPatternMax>

Sets the end value of a data pattern range if `TRIGger<m>:MILStd:DATA:DCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<DataPatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data Pattern" on page 608

---

**TRIGger<m>:MILStd:DATA:DMIN** <DataPatternMin>

Specifies a data pattern or sets the the start value of a data pattern range.

**Parameters:**
<DataPatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data Pattern" on page 608

---

**TRIGger<m>:MILStd:DATA:ICONdition** <DataIdxOperator>

Sets the operator to set a specific range within this series of the data words that is considered for the analysis.

**Parameters:**
<DataIdxOperator> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal| LTHan | LETHan | GTHan | GETHan**
Equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with `TRIGger<m>:MILStd:DATA:IMIN`.

**INRange = RANGe**
In range: set the minimum and maximum value of the range with `TRIGger<m>:MILStd:DATA:IMIN` and `TRIGger<m>:MILStd:DATA:IMAX`.

*RST: INRange

**Manual operation:** See "Data Index" on page 608

---

**TRIGger<m>:MILStd:DATA:IMAX** <DataIndexMax>

Sets the end value of a data word series index range if `TRIGger<m>:MILStd:DATA:ICONdition` is set to `INRange`.

**Parameters:**
<DataIndexMax>
Range: 1 to 32
Increment: 1
*RST: 32

**Manual operation:** See "Data Index" on page 608

---

**TRIGger<m>:MILStd:DATA:IMIN** <DataIndexMin>

Specifies an index or sets the the start value of a data word series index range.

**Parameters:**
<DataIndexMin>
Range: 1 to 32
Increment: 1
*RST: 1

**Manual operation:** See "Data Index" on page 608

---

**TRIGger<m>:MILStd:ERRor:MANChester** <ManCodingError>

Triggers on an error of the manchester coding if `TRIGger<m>:MILStd:TYPE` is set to `ERRor`.

**Parameters:**
<ManCodingError>
ON | OFF
*RST: ON

**Manual operation:** See "Manchester Coding Error" on page 610

---

**TRIGger<m>:MILStd:ERRor:PARity** <ParityError>

Triggers on even parity if `TRIGger<m>:MILStd:TYPE` is set to `ERRor`.

**Parameters:**
<ParityError>
ON | OFF
*RST: ON

**Manual operation:** See "Parity Error" on page 610

**TRIGger<m>:MILStd:ERRor:SYNC** <SyncError>

Triggers on an error of the synchronization if `TRIGger<m>:MILStd:TYPE` is set to `ERRor`.

**Parameters:**
<SyncError> ON | OFF
*RST: ON

**Manual operation:** See "Synchronization Error" on page 610

**TRIGger<m>:MILStd:MAXResponse:BITS** <MaxResponseTime>

Sets the value for the maximum response time to be triggered on if `TRIGger<m>:MILStd:TYPE` is set to `ERRor`

Enable the error trigger on maximum response time with `TRIGger<m>:MILStd:MAXResponse:SELect`.

**Parameters:**
<MaxResponseTime> Range: 2E-6 to 262E-6
Increment: 100E-9
*RST: 14E-6
Default unit: s

**Manual operation:** See "Check max response" on page 603

**TRIGger<m>:MILStd:MAXResponse:SELect** <MaxRespSel>

Enables the trigger on exceeding the maximum response time if `TRIGger<m>:MILStd:TYPE` is set to `ERRor`. You can set the maximum time with: `TRIGger<m>:MILStd:MAXResponse:BITS`.

**Parameters:**
<MaxRespSel> ON | OFF
*RST: ON

**Manual operation:** See "Check max response" on page 603

**TRIGger<m>:MILStd:MINGap:BITS** <MinGapTime>

Sets the value for the minimum gap to be triggered on if `TRIGger<m>:MILStd:TYPE` is set to `ERRor`.

**Parameters:**
<MinGapTime> Range: 2E-6 to 262E-6
Increment: 100E-9
*RST: 4E-6
Default unit: s

**Manual operation:** See "Check min gap" on page 603

**TRIGger<m>:MILStd:MINGap:SELect** <MinGapSelect>

Enables triggering when the minimum gap is out of range if TRIGger<m>:MILStd:TYPE is set to ERRor. You can set the minimum gap with: TRIGger<m>:MILStd:MINGap:BITS.

**Parameters:**
<MinGapSelect> ON | OFF
*RST: ON

**Manual operation:** See "Check min gap" on page 603

**TRIGger<m>:MILStd:STATus:BCReceived** <BroadcastCmd>

Triggers on the state of the broadcast command received bit of the status word if TRIGger<m>:MILStd:TYPE is set to STATword.

**Parameters:**
<BroadcastCmd> ONE | ZERO | DC
*RST: DC

**Manual operation:** See "Status Flags" on page 610

**TRIGger<m>:MILStd:STATus:BUSY** <BusyFlag>

Triggers on the state of the busy bit of the status word if TRIGger<m>:MILStd:TYPE is set to STATword.

**Parameters:**
<BusyFlag> ONE | ZERO | DC
*RST: DC

**Manual operation:** See "Status Flags" on page 610

**TRIGger<m>:MILStd:STATus:DBCaccept** <DynBusControl>

Triggers on the state of the dynamic bus control accept bit of the status word if TRIGger<m>:MILStd:TYPE is set to STATword.

**Parameters:**
<DynBusControl> ONE | ZERO | DC
*RST: DC

**Manual operation:** See "Status Flags" on page 610

**TRIGger<m>:MILStd:STATus:INSTrument** <InstFlag>

Triggers on the state of the instrumentation bit of the status word if TRIGger<m>:MILStd:TYPE is set to STATword.

**Parameters:**
<InstFlag> ONE | ZERO | DC
*RST: ZERO

**Manual operation:** See "Status Flags" on page 610

---

**TRIGger<m>:MILStd:STATus:MERRor** <MessageError>

Triggers on the state of the message error bit of the status word if `TRIGger<m>:MILStd:TYPE` is set to `STATword`.

**Parameters:**
<MessageError> ONE | ZERO | DC
*RST: DC

**Manual operation:** See "Status Flags" on page 610

---

**TRIGger<m>:MILStd:STATus:SREQuest** <ServiceRequest>

Triggers on the state of the the service request bit of the status word if `TRIGger<m>:MILStd:TYPE` is set to `STATword`.

**Parameters:**
<ServiceRequest> ONE | ZERO | DC
*RST: DC

**Manual operation:** See "Status Flags" on page 610

---

**TRIGger<m>:MILStd:STATus:SUBSystem** <SubsystemFlag>

Triggers on the state of the subsystem flag bit of the status word if `TRIGger<m>:MILStd:TYPE` is set to `STATword`.

**Parameters:**
<SubsystemFlag> ONE | ZERO | DC
*RST: DC

**Manual operation:** See "Status Flags" on page 610

---

**TRIGger<m>:MILStd:STATus:TERMinal** <TerminalFlag>

Triggers on the state of the terminal flag bit of the status word if `TRIGger<m>:MILStd:TYPE` is set to `STATword`.

**Parameters:**
<TerminalFlag> ONE | ZERO | DC
*RST: DC

**Manual operation:** See "Status Flags" on page 610

**TRIGger<m>:MILStd:TPSPecifier** <TypeSpecifier>

Sets the sync impulse/ word type to be triggered on.

**Parameters:**

<TypeSpecifier> CSTatus | DATA | ALL
CSTatus: command/status word
*RST: ALL

**Manual operation:** See "Sync Type / Word Type" on page 607

### 24.17.8.3 Decode results

To load and acivate a label list, use:

- `BUS<m>:NEWList` on page 1639
- `BUS<m>:SYMBols` on page 1639

BUS<m>:MILStd:WCOunt? ..... 1797
BUS<m>:MILStd:WORD<n>:DATA? ..... 1797
BUS<m>:MILStd:WORD<n>:INFO? ..... 1798
BUS<m>:MILStd:WORD<n>:RTADdress? ..... 1798
BUS<m>:MILStd:WORD<n>:STARt? ..... 1798
BUS<m>:MILStd:WORD<n>:STATus? ..... 1799
BUS<m>:MILStd:WORD<n>:STOP? ..... 1799
BUS<m>:MILStd:WORD<n>:SYMBol? ..... 1800
BUS<m>:MILStd:WORD<n>:TYPE? ..... 1800

**BUS<m>:MILStd:WCOunt?**

Returns the number of decoded words.

**Suffix:**

<m> 1..4

**Return values:**

<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:MILStd:WORD<n>:DATA?**

Return the data bytes of the specified word.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**

<Data> 16-bit data of the specified word as a 2-byte bit pattern (B1, B2). The first byte B1 is the most significant byte.

**Example:**
```
BUS:MILStd:WORD4:DATA?
<-- #H08,#H49
```

**Usage:** Query only

---

**BUS<m>:MILStd:WORD<n>:INFO?**

Returns the info value for the specified word.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**

<FrameInfo> Range: 0 to 2047
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:MILStd:WORD<n>:RTADdress?**

Returns the RT address for the selected word.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**

<FrameRta> Range: 0 to 31
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:MILStd:WORD<n>:STARt?**

Return the start time of the selected word.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**
<FrameStart> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:MILStd:WORD<n>:STATus?**

Returns the overall state of the selected word.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**
<FrameState> OK | SYNC | MANC | PAR | GAP | RT | INComplete | UNKNown
OK: the word is valid.
SYNC: synchronization error occured.
MANC: manchester coding error occured.
PAR: parity error occured.
GAP: timing gap error occured.
RT: remote terminal error occured.
INComplete: the sequence is not completely contained in the acquisition
UNKNown: unknown frame type
*RST: OK

**Usage:** Query only

---

**BUS<m>:MILStd:WORD<n>:STOP?**

Return the stop time of the selected word.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**
<FrameStop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:MILStd:WORD<n>:SYMBol?**

Returns the label name of the word ID.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**
<Translation>

**Usage:** Query only

---

**BUS<m>:MILStd:WORD<n>:TYPE?**

Returns the type of the specified word.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**
<FrameType> UNKNown | DATA | CMD | STAT | CMST | IM

CMD: command word
CMST: command/status word
IM: inter message. Shows if there are gap errors or response timeout.

*RST: DATA

**Usage:** Query only

### 24.17.8.4 Search settings

The search remote commands are very similar to the trigger commands. Therefore, search coommands are described in short, for more details, see the corresponding trigger command in Chapter 24.17.8.2, "Trigger", on page 1786.

SEARch:TRIGger:MILStd:TYPE ..... 1801
SEARch:TRIGger:MILStd:DATA:RCONdition ..... 1802
SEARch:TRIGger:MILStd:CMD:RCONdition ..... 1802
SEARch:TRIGger:MILStd:CDST:RCONdition ..... 1802
SEARch:TRIGger:MILStd:DATA:RMIN ..... 1802
SEARch:TRIGger:MILStd:CMD:RMIN ..... 1802
SEARch:TRIGger:MILStd:CDST:RMIN ..... 1802

SEARch:TRIGger:MILStd:DATA:RMAX ... 1803
SEARch:TRIGger:MILStd:CMD:RMAX ... 1803
SEARch:TRIGger:MILStd:CDST:RMAX ... 1803
SEARch:TRIGger:MILStd:CDST:ICONdition ... 1803
SEARch:TRIGger:MILStd:CMD:CCONdition ... 1803
SEARch:TRIGger:MILStd:CMD:SCONdition ... 1803
SEARch:TRIGger:MILStd:DATA:DCONdition ... 1803
SEARch:TRIGger:MILStd:CDST:IMIN ... 1804
SEARch:TRIGger:MILStd:CMD:CMIN ... 1804
SEARch:TRIGger:MILStd:CMD:SMIN ... 1804
SEARch:TRIGger:MILStd:DATA:DMIN ... 1804
SEARch:TRIGger:MILStd:CDST:IMAX ... 1804
SEARch:TRIGger:MILStd:CMD:CMAX ... 1804
SEARch:TRIGger:MILStd:CMD:SMAX ... 1804
SEARch:TRIGger:MILStd:DATA:DMAX ... 1804
SEARch:TRIGger:MILStd:DATA:ICONdition ... 1805
SEARch:TRIGger:MILStd:DATA:IMIN ... 1805
SEARch:TRIGger:MILStd:DATA:IMAX ... 1805
SEARch:TRIGger:MILStd:CMD:TR ... 1806
SEARch:TRIGger:MILStd:ERRor:MANChester ... 1806
SEARch:TRIGger:MILStd:ERRor:PARity ... 1806
SEARch:TRIGger:MILStd:ERRor:SYNC ... 1806
SEARch:TRIGger:MILStd:ERRor:TIMing ... 1806
SEARch:TRIGger:MILStd:STATus:BCReceived ... 1806
SEARch:TRIGger:MILStd:STATus:BUSY ... 1806
SEARch:TRIGger:MILStd:STATus:DBCaccept ... 1806
SEARch:TRIGger:MILStd:STATus:INSTrument ... 1806
SEARch:TRIGger:MILStd:STATus:MERRor ... 1806
SEARch:TRIGger:MILStd:STATus:SREQuest ... 1807
SEARch:TRIGger:MILStd:STATus:SUBSystem ... 1807
SEARch:TRIGger:MILStd:STATus:TERMinal ... 1807
SEARch:TRIGger:MILStd:TPSPecifier ... 1807

---

**SEARch:TRIGger:MILStd:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:MILStd:TYPE?** <SearchName>

Sets the event to be searched for.

**Parameters:**

<Type> STYPe | WTYPe | DATA | CDST | CMD | STATword | ERR

See `TRIGger<m>:MILStd:TYPE` on page 1787

*RST: STYPe

**Parameters for setting and query:**

<SearchName>

**Manual operation:** See "Type" on page 613

**SEARch:TRIGger:MILStd:DATA:RCONdition** <SearchName>,<RTAOperator>
**SEARch:TRIGger:MILStd:DATA:RCONdition?** <SearchName>
**SEARch:TRIGger:MILStd:CMD:RCONdition** <SearchName>,<RTAOperator>
**SEARch:TRIGger:MILStd:CMD:RCONdition?** <SearchName>
**SEARch:TRIGger:MILStd:CDST:RCONdition** <SearchName>,<RTAOperator>
**SEARch:TRIGger:MILStd:CDST:RCONdition?** <SearchName>

Set the operator to define a remote terminal address:

- DATA: for data words
- CMD: for command words
- CDST: for status words

**Parameters:**

<RTAOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with `SEARch:TRIGger:MILStd:CDST:RMIN`.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:MILStd:CDST:RMIN` and `SEARch:TRIGger:MILStd:CDST:RMAX` on page 1803.

*RST: EQUal

**Parameters for setting and query:**

<SearchName> String parameter

**Manual operation:** See "Remote terminal address setup: Condition, RTA " on page 614

**SEARch:TRIGger:MILStd:DATA:RMIN** <SearchName>,<RTAPatternMin>
**SEARch:TRIGger:MILStd:DATA:RMIN?** <SearchName>
**SEARch:TRIGger:MILStd:CMD:RMIN** <SearchName>,<RTAPatternMin>
**SEARch:TRIGger:MILStd:CMD:RMIN?** <SearchName>
**SEARch:TRIGger:MILStd:CDST:RMIN** <SearchName>,<RTAPatternMin>
**SEARch:TRIGger:MILStd:CDST:RMIN?** <SearchName>

Specify a remote terminal address or set the the start value of a remote terminal address range:

- DATA: for data words
- CMD: for command words
- CDST: for status words

**Parameters:**

<RTAPatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Remote terminal address setup: Condition, RTA " on page 614

---

**SEARch:TRIGger:MILStd:DATA:RMAX** <SearchName>,<RTAPatternMax>
**SEARch:TRIGger:MILStd:DATA:RMAX?** <SearchName>
**SEARch:TRIGger:MILStd:CMD:RMAX** <SearchName>,<RTAPatternMax>
**SEARch:TRIGger:MILStd:CMD:RMAX?** <SearchName>
**SEARch:TRIGger:MILStd:CDST:RMAX** <SearchName>,<RTAPatternMax>
**SEARch:TRIGger:MILStd:CDST:RMAX?** <SearchName>

Set the end value of a data range if `SEARch:TRIGger:MILStd:CDST:RCONdition` is set to `INRange` or `OORange`:

- DATA: for data words
- CMD: for command words
- CDST: for status words

**Parameters:**
<RTAPatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Remote terminal address setup: Condition, RTA " on page 614

---

**SEARch:TRIGger:MILStd:CDST:ICONdition** <SearchName>,<InfoOperator>
**SEARch:TRIGger:MILStd:CDST:ICONdition?** <SearchName>
**SEARch:TRIGger:MILStd:CMD:CCONdition** <SearchName>,<WordCntOperator>
**SEARch:TRIGger:MILStd:CMD:CCONdition?** <SearchName>
**SEARch:TRIGger:MILStd:CMD:SCONdition** <SearchName>,<SubaddrOperator>
**SEARch:TRIGger:MILStd:CMD:SCONdition?** <SearchName>
**SEARch:TRIGger:MILStd:DATA:DCONdition** <SearchName>,<DataOperator>
**SEARch:TRIGger:MILStd:DATA:DCONdition?** <SearchName>

Sets the operator for the corresponding search:

- CDST:ICON - specific info for the 9th to 19th bit of a command or status word.
- CMD:CCON - specific data word count or mode code pattern in a command word
- CMD:SCON - specific subaddress/mode pattern in a command word
- DATA:DCON - data pattern in a data word

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange
*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data pattern setup: Condition, Data min, Data max" on page 614

---

**SEARch:TRIGger:MILStd:CDST:IMIN** <SearchName>,<InfoPatternMin>
**SEARch:TRIGger:MILStd:CDST:IMIN?** <SearchName>
**SEARch:TRIGger:MILStd:CMD:CMIN** <SearchName>,<WordCntPattMin>
**SEARch:TRIGger:MILStd:CMD:CMIN?** <SearchName>
**SEARch:TRIGger:MILStd:CMD:SMIN** <SearchName>,<SubaddrPattMin>
**SEARch:TRIGger:MILStd:CMD:SMIN?** <SearchName>
**SEARch:TRIGger:MILStd:DATA:DMIN** <SearchName>,<DataPatternMin>
**SEARch:TRIGger:MILStd:DATA:DMIN?** <SearchName>

Sets the pattern or the start value of a pattern range for the corresponding search:

- CDST:IMIN - specific info for the 9th to 19th bit of a command or status word.
- CMD:CMIN - specific data word count or mode code pattern in a command word
- CMD:SMIN - specific subaddress/mode pattern in a command word
- DATA:DMIN - data pattern in a data word

**Parameters:**
<DataPatternMin>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data pattern setup: Condition, Data min, Data max" on page 614

---

**SEARch:TRIGger:MILStd:CDST:IMAX** <SearchName>,<InfoPatternMax>
**SEARch:TRIGger:MILStd:CDST:IMAX?** <SearchName>
**SEARch:TRIGger:MILStd:CMD:CMAX** <SearchName>,<WordCntPattMax>
**SEARch:TRIGger:MILStd:CMD:CMAX?** <SearchName>
**SEARch:TRIGger:MILStd:CMD:SMAX** <SearchName>,<SubaddrPattMax>
**SEARch:TRIGger:MILStd:CMD:SMAX?** <SearchName>
**SEARch:TRIGger:MILStd:DATA:DMAX** <SearchName>,<DataPatternMax>
**SEARch:TRIGger:MILStd:DATA:DMAX?** <SearchName>

Sets the end value of a pattern range for the corresponding search:

- CDST:IMAX - specific info for the 9th to 19th bit of a command or status word.
- CMD:CMAX - specific data word count or mode code pattern in a command word
- CMD:SMAX - specific subaddress/mode pattern in a command word
- DATA:DMAX - data pattern in a data word

**Parameters:**
<DataPatternMax>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data pattern setup: Condition, Data min, Data max" on page 614

---

**SEARch:TRIGger:MILStd:DATA:ICONdition** <SearchName>,<DataIdxOperator>
**SEARch:TRIGger:MILStd:DATA:ICONdition?** <SearchName>

Sets the operator to set a range within a series of the data words that is considered for the search.

**Parameters:**
<DataIdxOperator> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

INRange = RANGe

*RST: INRange

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data index setup: Condition, Index min, Index max" on page 614

---

**SEARch:TRIGger:MILStd:DATA:IMIN** <SearchName>,<DataIndexMin>
**SEARch:TRIGger:MILStd:DATA:IMIN?** <SearchName>

Specifies an index or sets the the start value of a data word series index range.

**Parameters:**
<DataIndexMin> Range: 1 to 32
Increment: 1
*RST: 1

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data index setup: Condition, Index min, Index max" on page 614

---

**SEARch:TRIGger:MILStd:DATA:IMAX** <SearchName>,<DataIndexMax>
**SEARch:TRIGger:MILStd:DATA:IMAX?** <SearchName>

Sets the end value of a data word series index range if the operator is set to `INRange`.

**Parameters:**
<DataIndexMax> Range: 1 to 32
Increment: 1
*RST: 32

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data index setup: Condition, Index min, Index max" on page 614

**SEARch:TRIGger:MILStd:CMD:TR** <SearchName>,<TRFlag>
**SEARch:TRIGger:MILStd:CMD:TR?** <SearchName>

Searches for a transmission mode.

**Parameters:**
<TRFlag> ONE | ZERO | DC
*RST: DC

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "T/R (Transmit/receive)" on page 615

---

**SEARch:TRIGger:MILStd:ERRor:MANChester** <SearchName>,<ManCodingError>
**SEARch:TRIGger:MILStd:ERRor:MANChester?** <SearchName>
**SEARch:TRIGger:MILStd:ERRor:PARity** <SearchName>,<ParityError>
**SEARch:TRIGger:MILStd:ERRor:PARity?** <SearchName>
**SEARch:TRIGger:MILStd:ERRor:SYNC** <SearchName>,<SyncError>
**SEARch:TRIGger:MILStd:ERRor:SYNC?** <SearchName>
**SEARch:TRIGger:MILStd:ERRor:TIMing** <SearchName>,<MinGapSelect>
**SEARch:TRIGger:MILStd:ERRor:TIMing?** <SearchName>

Enables search for errors if `SEARch:TRIGger:MILStd:TYPE` is set to `ERRor`.

- MANChester: error of the manchester coding
- PARity: even parity (parity error)
- SYNC: error of the synchronization
- TIMing: Minimum gap is out of range

**Parameters:**
<SyncError>, <ParityError>, <ManCodingError>, <MinGapSelect> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 615

---

**SEARch:TRIGger:MILStd:STATus:BCReceived** <SearchName>,<BroadcastCmd>
**SEARch:TRIGger:MILStd:STATus:BCReceived?** <SearchName>
**SEARch:TRIGger:MILStd:STATus:BUSY** <SearchName>,<BusyFlag>
**SEARch:TRIGger:MILStd:STATus:BUSY?** <SearchName>
**SEARch:TRIGger:MILStd:STATus:DBCaccept** <SearchName>,<DynBusControl>
**SEARch:TRIGger:MILStd:STATus:DBCaccept?** <SearchName>
**SEARch:TRIGger:MILStd:STATus:INSTrument** <SearchName>,<InstFlag>
**SEARch:TRIGger:MILStd:STATus:INSTrument?** <SearchName>
**SEARch:TRIGger:MILStd:STATus:MERRor** <SearchName>,<MessageError>
**SEARch:TRIGger:MILStd:STATus:MERRor?** <SearchName>

**SEARch:TRIGger:MILStd:STATus:SREQuest** <SearchName>,<ServiceRequest>
**SEARch:TRIGger:MILStd:STATus:SREQuest?** <SearchName>
**SEARch:TRIGger:MILStd:STATus:SUBSystem** <SearchName>,<SubsystemFlag>
**SEARch:TRIGger:MILStd:STATus:SUBSystem?** <SearchName>
**SEARch:TRIGger:MILStd:STATus:TERMinal** <SearchName>,<TerminalFlag>
**SEARch:TRIGger:MILStd:STATus:TERMinal?** <SearchName>

Specifies the values (0, 1, X) of the status flags if `SEARch:TRIGger:MILStd:TYPE` is set to `STATword`.

**Parameters:**
<TerminalFlag> ONE | ZERO | DC
*RST: DC

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Status flags setup" on page 615

---

**SEARch:TRIGger:MILStd:TPSPecifier** <SearchName>,<TypeSpecifier>
**SEARch:TRIGger:MILStd:TPSPecifier?** <SearchName>

Sets the sync impulse/ word type to be searched for.

**Parameters:**
<TypeSpecifier> CSTatus | DATA | ALL
CSTatus: command/status word
*RST: ALL

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Sync Type / Word Type" on page 613

### 24.17.8.5 Search results

The search on decoded MIL-1553 data returns the same results as the queries for decode results.

In search result commands, you have to specify the <SearchName> parameter. It is a string parameter that contains the search definition name.

For a description of the returned values, see the corresponding commands in Chapter 24.17.8.3, "Decode results", on page 1797.

SEARch:RESult:MILStd:WCOunt? ..... 1808
SEARch:RESult:MILStd:WORD<m>:TYPE? ..... 1808
SEARch:RESult:MILStd:WORD<m>:STATus? ..... 1808
SEARch:RESult:MILStd:WORD<m>:STARt? ..... 1808
SEARch:RESult:MILStd:WORD<m>:STOP? ..... 1809
SEARch:RESult:MILStd:WORD<m>:SYMBol? ..... 1809

SEARch:RESult:MILStd:WORD<m>:RTADdress? ..... 1809
SEARch:RESult:MILStd:WORD<m>:DATA? ..... 1810
SEARch:RESult:MILStd:WORD<m>:INFO? ..... 1810

**SEARch:RESult:MILStd:WCOunt?** <SearchName>

**Query parameters:**
<SearchName>

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:MILStd:WORD<m>:TYPE?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameType> UNKNown | DATA | CMD | STAT | CMST | IM
*RST: DATA

**Usage:** Query only

**SEARch:RESult:MILStd:WORD<m>:STATus?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameState> OK | SYNC | MANC | PAR | GAP | RT | INComplete | UNKNown
*RST: OK

**Usage:** Query only

**SEARch:RESult:MILStd:WORD<m>:STARt?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**

<FrameStart> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:MILStd:WORD<m>:STOP?** <SearchName>

**Suffix:**

<m> *

**Query parameters:**

<SearchName>

**Return values:**

<FrameStop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:MILStd:WORD<m>:SYMBol?** <SearchName>

**Suffix:**

<m> *

**Query parameters:**

<SearchName>

**Return values:**

<Translation>

**Usage:** Query only

---

**SEARch:RESult:MILStd:WORD<m>:RTADdress?** <SearchName>

**Suffix:**

<m> *

**Query parameters:**

<SearchName>

**Return values:**

<FrameRta> Range: 0 to 31
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:MILStd:WORD<m>:DATA?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

**SEARch:RESult:MILStd:WORD<m>:INFO?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameInfo> Range: 0 to 2047
Increment: 1
*RST: 0

**Usage:** Query only

## 24.17.9 ARINC 429 (option R&S RTP-K7)


- Configuration....1810
- Trigger....1814
- Decode results....1818
- Search settings....1821
- Search results....1824


### 24.17.9.1 Configuration


BUS<m>:ARINc:SOURce....1811
BUS<m>:ARINc:BRValue....1811
BUS<m>:ARINc:BRMode....1811
BUS<m>:ARINc:MAXGap:SELect....1811
BUS<m>:ARINc:MAXGap:BITS....1812
BUS<m>:ARINc:MINGap:SELect....1812
BUS<m>:ARINc:MINGap:BITS....1812
BUS<m>:ARINc:POLarity....1813
BUS<m>:ARINc:PRESet....1813
BUS<m>:ARINc:THReshold:HIGH....1813
BUS<m>:ARINc:THReshold:LOW....1813

**BUS<m>:ARINc:SOURce** <SourceData>

Sets the channel for the signal source.

Reference and math waveforms are only available if the trigger source is one of the input channels but not the serial bus.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<SourceData> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: C1W1

**Manual operation:** See "Data" on page 618

**BUS<m>:ARINc:BRValue** <BitRateValue>

Sets the number of transmitted bits per second.

**Suffix:**
<m> 1..4

**Parameters:**
<BitRateValue> Range: 10000 to 110000
Increment: 100
*RST: 100000
Default unit: bps

**Manual operation:** See "Bit Rate" on page 619

**BUS<m>:ARINc:BRMode** <BitRateMode>

Sets the bit rate mode to high or low speed.

**Suffix:**
<m> 1..4

**Parameters:**
<BitRateMode> HIGH | LOW
*RST: HIGH

**Manual operation:** See "Bit Rate" on page 619

**BUS<m>:ARINc:MAXGap:SELect** <MaxGapSelect>

Enables the detection of the maximum gap time during decoding.

To specify the minimum gap time `BUS<m>:ARINc:MINGap:BITS`.

**Suffix:**
<m> 1..4

**Parameters:**
<MaxGapSelect> ON | OFF
*RST: OFF

**Manual operation:** See "Timing: Min gap, Max gap" on page 619

---

**BUS<m>:ARINc:MAXGap:BITS** <MaxGapBits>

Sets the value for the maximum gap between two words.

See also: `BUS<m>:ARINc:MAXGap:SELect`

**Suffix:**
<m> 1..4

**Parameters:**
<MaxGapBits> Range: 0 to 1000
Increment: 1
*RST: 100
Default unit: bit

**Manual operation:** See "Timing: Min gap, Max gap" on page 619

---

**BUS<m>:ARINc:MINGap:SELect** <MinGapSelect>

Enables the detection of the minimum idle time between two words during decoding.

To specify the minimum gap, use`BUS<m>:ARINc:MINGap:BITS`.

**Suffix:**
<m> 1..4

**Parameters:**
<MinGapSelect> ON | OFF
*RST: ON

**Manual operation:** See "Timing: Min gap, Max gap" on page 619

---

**BUS<m>:ARINc:MINGap:BITS** <MinGapBits>

Sets a value for the minimum timing gap between two words.

See also: `BUS<m>:ARINc:MINGap:SELect` on page 1812.

**Suffix:**
<m> 1..4

**Parameters:**
<MinGapBits> Range: 0 to 100
Increment: 1
*RST: 4
Default unit: bit

**Manual operation:** See "Timing: Min gap, Max gap" on page 619

**BUS<m>:ARINc:POLarity** <Polarity>

Sets the wire on which the bus signal is measured.

**Suffix:**
<m> 1..4

**Parameters:**
<Polarity> ALEG | BLEG
*RST: ALEG

**Manual operation:** See "Polarity" on page 618

**BUS<m>:ARINc:PRESet** <Preset>

Sets the default threshold voltage.

**Suffix:**
<m> 1..4

**Parameters:**
<Preset> V25 | V5 | MANual
*RST: V5

**Manual operation:** See "Thresholds" on page 618

**BUS<m>:ARINc:THReshold:HIGH** <ThresholdHigh>

Sets the high threshold level of the signal.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdHigh> Range: 0 to 12
Increment: 0.1
*RST: 5
Default unit: V

**Manual operation:** See "Thresholds" on page 618

**BUS<m>:ARINc:THReshold:LOW** <ThresholdLow>

Sets the low threshold level of the signal.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdLow> Range: -12 to 0
Increment: 0.1
*RST: -5
Default unit: V

**Manual operation:** See "Thresholds" on page 618

### 24.17.9.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:ARINc:TYPE....1814
TRIGger<m>:ARINc:DATA:CONDition....1814
TRIGger<m>:ARINc:DATA:MIN....1815
TRIGger<m>:ARINc:DATA:MAX....1815
TRIGger<m>:ARINc:ERRor:CODing....1815
TRIGger<m>:ARINc:ERRor:PARity....1815
TRIGger<m>:ARINc:LABel:CONDition....1816
TRIGger<m>:ARINc:LABel:MIN....1816
TRIGger<m>:ARINc:LABel:MAX....1816
TRIGger<m>:ARINc:MINGap:SELect....1816
TRIGger<m>:ARINc:MINGap:BITS....1817
TRIGger<m>:ARINc:MAXGap:SELect....1817
TRIGger<m>:ARINc:MAXGap:BITS....1817
TRIGger<m>:ARINc:SDI....1817
TRIGger<m>:ARINc:SSM....1817

---

**TRIGger<m>:ARINc:TYPE** <Type>

Sets the trigger type for ARINC 429 analysis.

**Parameters:**

<Type> STARt | STOP | LABel | ERRor

*RST: STARt

**Manual operation:** See "Trigger Type" on page 621

---

**TRIGger<m>:ARINc:DATA:CONDition** <DataOperator>

Sets the condition for the data. You can define an exact data patterm or a data range.

**Parameters:**

<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with `TRIGger<m>:ARINc:DATA:MIN`.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:ARINc:DATA:MIN` and `TRIGger<m>:ARINc:DATA:MAX`.

*RST: EQUal

**Manual operation:** See "Data setup: Condition, Data Min, Data Max" on page 623

---

**TRIGger<m>:ARINc:DATA:MIN** <DataMin>

Defines the minimum bit pattern for the data.

**Parameters:**
<DataMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data setup: Condition, Data Min, Data Max" on page 623

---

**TRIGger<m>:ARINc:DATA:MAX** <DataMax>

Sets the end value of a data pattern if `TRIGger<m>:ARINc:DATA:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<DataMax>

**Manual operation:** See "Data setup: Condition, Data Min, Data Max" on page 623

---

**TRIGger<m>:ARINc:ERRor:CODing** <CodingError>

Enables triggering when a coding error occurs.

**Parameters:**
<CodingError> ON | OFF

*RST: ON

**Manual operation:** See "Coding error" on page 623

---

**TRIGger<m>:ARINc:ERRor:PARity** <ParityError>

Enables triggering when a parity error occurs.

**Parameters:**
<ParityError> ON | OFF

*RST: ON

**Manual operation:** See "Parity Error" on page 623

**TRIGger<m>:ARINc:LABel:CONDition** <LabelOperator>

Sets the condition for the label. You can define an exact label or a label range.

**Parameters:**

<LabelOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with`TRIGger<m>:ARINc:LABel:MIN`.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with`TRIGger<m>:ARINc:LABel:MIN` and `TRIGger<m>:ARINc:LABel:MAX`.

*RST: EQUal

**Manual operation:** See "Label setup: Condition, Label Min, Label Max" on page 622

**TRIGger<m>:ARINc:LABel:MIN** <LabelMin>

Defines the minimum bit pattern for the label.

**Parameters:**

<LabelMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Label setup: Condition, Label Min, Label Max" on page 622

**TRIGger<m>:ARINc:LABel:MAX** <LabelMax>

Sets the end value of a label pattern if `TRIGger<m>:ARINc:LABel:CONDition` is set to `INRange` or `OORange`.

**Parameters:**

<LabelMax> Numeric or string pattern, see `TRIGger<m>:ARINc:LABel:MIN`

**Manual operation:** See "Label setup: Condition, Label Min, Label Max" on page 622

**TRIGger<m>:ARINc:MINGap:SELect** <MinGapSelect>

Enables triggering when the minimum gap is out of range. You can set the minimum gap with: `TRIGger<m>:ARINc:MINGap:BITS`.

**Parameters:**

<MinGapSelect> ON | OFF

*RST: ON

**Manual operation:** See "Timing: Min gap, Max gap" on page 619

**TRIGger<m>:ARINc:MINGap:BITS** <MinGapBits>

Sets the value for the minimum gap to be triggered on.

**Parameters:**
<MinGapBits> Range: 0 to 100
Increment: 1
*RST: 4
Default unit: bit

**Manual operation:** See "Timing: Min gap, Max gap" on page 619

**TRIGger<m>:ARINc:MAXGap:SELect** <MaxGapSelect>

Enables triggering when the maximum gap is out of range. You can set the maximum gap with: `TRIGger<m>:ARINc:MAXGap:BITS`.

**Parameters:**
<MaxGapSelect> ON | OFF
*RST: OFF

**Manual operation:** See "Timing: Min gap, Max gap" on page 619

**TRIGger<m>:ARINc:MAXGap:BITS** <MaxGapBits>

Sets the value for the maximum gap to be triggered on.

**Parameters:**
<MaxGapBits> Range: 0 to 1000
Increment: 1
*RST: 100
Default unit: bit

**Manual operation:** See "Timing: Min gap, Max gap" on page 619

**TRIGger<m>:ARINc:SDI** <SDI>

Sets the source/destination identifier (SDI) bits.

**Parameters:**
<SDI>

**Manual operation:** See "SDI / SSM" on page 623

**TRIGger<m>:ARINc:SSM** <SSM>

Sets the sign/status matrix (SSM) bits.

**Parameters:**
<SSM>

**Manual operation:** See "SDI / SSM" on page 623

### 24.17.9.3 Decode results

To load and acivate a label list, use:

- BUS<m>:NEWList on page 1639
- BUS<m>:SYMBols on page 1639

BUS<m>:ARINc:WCOunt?....1818
BUS<m>:ARINc:WORD<n>:DATA?....1818
BUS<m>:ARINc:WORD<n>:LABel?....1819
BUS<m>:ARINc:WORD<n>:PATTern?....1819
BUS<m>:ARINc:WORD<n>:SDI?....1819
BUS<m>:ARINc:WORD<n>:SSM?....1820
BUS<m>:ARINc:WORD<n>:STARt?....1820
BUS<m>:ARINc:WORD<n>:STATe?....1820
BUS<m>:ARINc:WORD<n>:STOP?....1821
BUS<m>:ARINc:WORD<n>:SYMBol?....1821

---

**BUS<m>:ARINc:WCOunt?**

Returns the number of decoded words.

**Suffix:**
<m> 1..4

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:ARINc:WORD<n>:DATA?**

Returns the data of the specified word.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<FrameData> 19-bit data field of the word as an integer
Range: 0 to 0
Increment: 1
*RST: 0

**Example:**
```
BUS:ARINc:WORD3:DATA?
<-- 148035
```

**Usage:** Query only

**BUS<m>:ARINc:WORD<n>:LABel?**

Returns the label of the specified word.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<FrameLabel> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:ARINc:WORD<n>:PATTern?**

Returns all 32 bits of the specified word.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *

**Return values:**
<Data> Comma-separated list of 4 bytes in big endian order. The format of each byte is defined by `FORMat:BPATtern`.

**Example:**
```
BUS2:ARINc:WORD3:PATTern?
--> #H75,#H11,#H55,#H82
FORMat:BPATtern DEC
BUS2:ARINc:WORD3:PATTern?
--> 117,17,85,130
```

**Usage:** Query only

**BUS<m>:ARINc:WORD<n>:SDI?**

Returns the source/destination identifier (SDI) bits of the specified word.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<SDI> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:ARINc:WORD<n>:SSM?**

Returns the sign/status matrix(SSM) bits of the specified word.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<SSM> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:ARINc:WORD<n>:STARt?**

Returns the start time of the specified word.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<FrameStart> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:ARINc:WORD<n>:STATe?**

Returns the overall state of the specified word.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the word.

**Return values:**
<FrameState> OK | CODE | GAP | PAR | INComplete | UNKNown
CODE: coding error occured.
GAP: timing gap error occured.
PAR: parity error occured.
INComplete: the sequence is not completely contained in the acquisition
UNKNown: unknown frame type
*RST: OK

**Usage:** Query only

---

**BUS<m>:ARINc:WORD<n>:STOP?**

Returns the end time of the specified word.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<FrameStop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:ARINc:WORD<n>:SYMBol?**

Returns the label name of the word ID.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Translation>

**Usage:** Query only

### 24.17.9.4 Search settings

SEARch:TRIGger:ARINc:TYPE ..... 1821
SEARch:TRIGger:ARINc:LABel:CONDition ..... 1822
SEARch:TRIGger:ARINc:DATA:CONDition ..... 1822
SEARch:TRIGger:ARINc:LABel:MIN ..... 1822
SEARch:TRIGger:ARINc:DATA:MIN ..... 1822
SEARch:TRIGger:ARINc:LABel:MAX ..... 1823
SEARch:TRIGger:ARINc:DATA:MAX ..... 1823
SEARch:TRIGger:ARINc:SDI ..... 1823
SEARch:TRIGger:ARINc:SSM ..... 1823
SEARch:TRIGger:ARINc:ERRor:CODing ..... 1823
SEARch:TRIGger:ARINc:ERRor:PARity ..... 1824
SEARch:TRIGger:ARINc:ERRor:TIMing ..... 1824

---

**SEARch:TRIGger:ARINc:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:ARINc:TYPE?** <SearchName>

Sets the search type.

**Parameters:**
<Type> STARt | STOP | LABel | ERRor
*RST: STARt

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Type" on page 627

---

**SEARch:TRIGger:ARINc:LABel:CONDition** <SearchName>,<LabelOperator>
**SEARch:TRIGger:ARINc:LABel:CONDition?** <SearchName>
**SEARch:TRIGger:ARINc:DATA:CONDition** <SearchName>,<DataOperator>
**SEARch:TRIGger:ARINc:DATA:CONDition?** <SearchName>

Set the condition for the label or data, respectively. You can define an exact value or a value range

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:ARINc:...:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range. with`TRIGger<m>:ARINc:LABel:MIN` and `TRIGger<m>:ARINc:LABel:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: Condition, Data min, Data max" on page 627

---

**SEARch:TRIGger:ARINc:LABel:MIN** <SearchName>,<LabelMin>
**SEARch:TRIGger:ARINc:LABel:MIN?** <SearchName>
**SEARch:TRIGger:ARINc:DATA:MIN** <SearchName>,<DataMin>
**SEARch:TRIGger:ARINc:DATA:MIN?** <SearchName>

Specifies a label or data bit pattern, or sets the the start value of a pattern range.

**Parameters:**
<DataMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Data setup: Condition, Data min, Data max" on page 627

**SEARch:TRIGger:ARINc:LABel:MAX** <SearchName>,<LabelMax>
**SEARch:TRIGger:ARINc:LABel:MAX?** <SearchName>
**SEARch:TRIGger:ARINc:DATA:MAX** <SearchName>,<DataMax>
**SEARch:TRIGger:ARINc:DATA:MAX?** <SearchName>

Set the end value of a label or data pattern if the condition is set to `INRange` or `OORange`.

**Parameters:**
<DataMax> Numeric or string pattern, see `SEARch:TRIGger:ARINc:LABel:MIN`

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data setup: Condition, Data min, Data max" on page 627

**SEARch:TRIGger:ARINc:SDI** <SearchName>,<SDI>
**SEARch:TRIGger:ARINc:SDI?** <SearchName>

Sets the source/destination identifier (SDI) bits.

**Parameters:**
<SDI>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "SDI / SSM setup: SDI, SSM" on page 627

**SEARch:TRIGger:ARINc:SSM** <SearchName>,<SSM>
**SEARch:TRIGger:ARINc:SSM?** <SearchName>

Sets the sign/status matrix (SSM) bits.

**Parameters:**
<SSM>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "SDI / SSM setup: SDI, SSM" on page 627

**SEARch:TRIGger:ARINc:ERRor:CODing** <SearchName>,<CodingError>
**SEARch:TRIGger:ARINc:ERRor:CODing?** <SearchName>

Enables the search for coding errors.

**Parameters:**
<CodingError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 628

---

**SEARch:TRIGger:ARINc:ERRor:PARity** <SearchName>,<ParityError>
**SEARch:TRIGger:ARINc:ERRor:PARity?** <SearchName>

Enables the search for parity errors.

**Parameters:**
<ParityError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 628

---

**SEARch:TRIGger:ARINc:ERRor:TIMing** <SearchName>,<MinGapSelect>
**SEARch:TRIGger:ARINc:ERRor:TIMing?** <SearchName>

Enables the search for timing errors, when the minimum gap is out of range.

**Parameters:**
<MinGapSelect> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 628

### 24.17.9.5 Search results

The search on decoded ARINC 429 data returns the same results as the queries for decode results.

In search result commands, you have to specify the <SearchName> parameter. It is a string parameter that contains the search definition name.

For a description of the returned values, see the corresponding commands in Chapter 24.17.9.3, "Decode results", on page 1818.

SEARch:RESult:ARINc:WORD<m>:LABel? 1825
SEARch:RESult:ARINc:WORD<m>:PATTern? 1825
SEARch:RESult:ARINc:WORD<m>:DATA? 1825
SEARch:RESult:ARINc:WORD<m>:SSM? 1825
SEARch:RESult:ARINc:WORD<m>:SYMBol? 1826
SEARch:RESult:ARINc:WCOunt? 1826
SEARch:RESult:ARINc:WORD<m>:STOP? 1826
SEARch:RESult:ARINc:WORD<m>:SDI? 1826
SEARch:RESult:ARINc:WORD<m>:STATe? 1827
SEARch:RESult:ARINc:WORD<m>:STARt? 1827

**SEARch:RESult:ARINc:WORD<m>:LABel?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Label> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:ARINc:WORD<m>:PATTern?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<DataValues>

**Usage:** Query only

**SEARch:RESult:ARINc:WORD<m>:DATA?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data> Range: 0 to 0
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:ARINc:WORD<m>:SSM?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<SSM> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:ARINc:WORD<m>:SYMBol?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Translation>

**Usage:** Query only

---

**SEARch:RESult:ARINc:WCOunt?** <SearchName>

**Query parameters:**
<SearchName>

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:ARINc:WORD<m>:STOP?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameStop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:ARINc:WORD<m>:SDI?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<SDI> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:ARINc:WORD<m>:STATe?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameState> OK | CODE | GAP | PAR | INComplete | UNKNown
*RST: OK

**Usage:** Query only

---

**SEARch:RESult:ARINc:WORD<m>:STARt?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameStart> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

## 24.17.10 Ethernet 10BASE-T and 100BASE-TX (option R&S RTP-K8)


- Configuration....1827
- Trigger....1830
- Decode results....1836
- Search settings....1842
- Search results....1847


### 24.17.10.1 Configuration

In all `BUS<m>:ETHernet` commands, the suffix <m> selects the serial bus.

BUS<m>:ETHernet:VARiant....1828
BUS<m>:ETHernet:SOURce....1828
BUS<m>:ETHernet:POLarity....1828
BUS<m>:ETHernet:THReshold:HIGH....1829
BUS<m>:ETHernet:THReshold:LOW....1829
BUS<m>:ETHernet:PRESet....1829
BUS<m>:ETHernet:BITRate....1830

**BUS<m>:ETHernet:VARiant** <Variant>

Selects the Ethernet protocol variant and transmission speed.

**Suffix:**
<m> 1..4

**Parameters:**
<Variant> B10T | B100TX | B100tx

**B10T**
Ethernet protocol variant 10BASE-T (10 Mbit/s)

**B100TX = B100tx**
Ethernet protocol variant 100BASE-TX (100 Mbit/s)

*RST: B10T

**Manual operation:** See "Ethernet Variant" on page 630

**BUS<m>:ETHernet:SOURce** <SourceData>

Selects the source channel for the data signal.

Usually, the source is one of the analog channels. Reference and math waveforms are available as source if the trigger source is one of the analog channels but not the serial bus.

Alternatively, digital channels can be used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.

For triggering on a serial bus, analog or digital channel sources are required.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceData> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

See Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C1W1

**Manual operation:** See "Source" on page 631

**BUS<m>:ETHernet:POLarity** <Polarity>

Defines the polarity of the data signal. This setting is only available in 10BASE-T.

**Suffix:**
<m> 1..4

**Parameters:**
<Polarity> NORMal | INVert

**NORMal**
Normal (non-inverted) data signal polarity

**INVert**
Inverted data signal polarity

*RST: NORMal

**Manual operation:** See "Polarity" on page 631

---

**BUS<m>:ETHernet:THReshold:HIGH** <ThresholdUpper>

Sets the upper threshold value for the signal digitization. If the signal value is higher than the this threshold, the signal state is considered high.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdUpper> Range: 0 to 10
Increment: 0.01
*RST: 0.5
Default unit: V

**Manual operation:** See "Thresholds" on page 631

---

**BUS<m>:ETHernet:THReshold:LOW** <ThresholdLower>

Sets the lower threshold value for the signal digitization. If the signal value is below this threshold, the signal state is considered low.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdLower> Range: -10 to 0
Increment: 0.01
*RST: -0.5
Default unit: V

**Manual operation:** See "Thresholds" on page 631

---

**BUS<m>:ETHernet:PRESet** <ThresholdPreset>

Sets the thresholds to predefined or individually definable voltage levels.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdPreset> T0 | T100 | TX0 | TX100 | MANual

**T0**
Sets the thresholds to the default values for 10BASE-T (0 meters): upper threshold to 1.25 V, lower threshold to -1.25 V

**T100**
Sets the thresholds to the default values for 10BASE-T (100 meters): upper threshold to 0.75 V, lower threshold to -0.75 V

**TX0**
Sets the thresholds to the default values for 100BASE-TX (0 meters): upper threshold to 0.5 V, lower threshold to -0.5 V

**TX100**
Sets the thresholds to the default values for 100BASE-TX (100 meters): upper threshold to 0.35 V, lower threshold to -0.35 V

**MANual**
Allows to set individual threshold voltage levels

*RST: T0

**Manual operation:** See "Thresholds" on page 631

---

**BUS<m>:ETHernet:BITRate** <BitRateValue>

Sets the bit rate value that defines the transmission speed in bits per second.

**Suffix:**
<m> 1..4

**Parameters:**
<BitRateValue> Range: 10000 to 150000000
Increment: 1000
*RST: 10000000
Default unit: bps

**Manual operation:** See "Bit rate" on page 631

### 24.17.10.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.

- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:ETHernet:TYPE ..... 1831
TRIGger<m>:ETHernet:PATTern ..... 1831
TRIGger<m>:ETHernet:ERRor:CRC ..... 1832
TRIGger<m>:ETHernet:ERRor:LENGth ..... 1832
TRIGger<m>:ETHernet:ERRor:PREamble ..... 1832
TRIGger<m>:ETHernet:FRAMe:CCONdition ..... 1832
TRIGger<m>:ETHernet:FRAMe:CMAX ..... 1833
TRIGger<m>:ETHernet:FRAMe:CMIN ..... 1833
TRIGger<m>:ETHernet:FRAMe:DCONdition ..... 1833
TRIGger<m>:ETHernet:FRAMe:DMAX ..... 1834
TRIGger<m>:ETHernet:FRAMe:DMIN ..... 1834
TRIGger<m>:ETHernet:FRAMe:SCONdition ..... 1834
TRIGger<m>:ETHernet:FRAMe:SMAX ..... 1835
TRIGger<m>:ETHernet:FRAMe:SMIN ..... 1835
TRIGger<m>:ETHernet:FRAMe:TCONdition ..... 1835
TRIGger<m>:ETHernet:FRAMe:TMAX ..... 1836
TRIGger<m>:ETHernet:FRAMe:TMIN ..... 1836

**TRIGger<m>:ETHernet:TYPE** <Type>

Selects the type of frame to be triggered on.

**Suffix:**
<m> 1..3

**Parameters:**
<Type> STARt | PATTern | ADVFrame | ADVCerror

**STARt**
Selects to trigger for the start of frame: search for the preamble and set the trigger instant thereafter.

**PATTern**
Selects to trigger for a bit pattern (data) to be specified in `TRIGger<m>:ETHernet:PATTern` on page 1831.

**ADVFrame**
Sets the trigger to the advanced frame. You can further define the values of the frame.

**ADVCerror**
Sets the trigger to advanced error. You can further enable the error conditions you want to trigger on.

*RST: STARt

**Manual operation:** See "Type" on page 635

**TRIGger<m>:ETHernet:PATTern** <DataPattern>

Specifies the bit pattern (or data) that is to be triggered.

**Suffix:**
<m> 1..3

**Parameters:**
<DataPattern>

**Manual operation:** See "Data Pattern" on page 637

---

**TRIGger<m>:ETHernet:ERRor:CRC** <ErrorCRC>

Enables the trigger on a mismatch of the cyclic redundancy check (CRC) value between the transmitting and receiving device.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorCRC> ON | OFF
*RST: ON

**Manual operation:** See "Error" on page 638

---

**TRIGger<m>:ETHernet:ERRor:LENGth** <ErrorLength>

Enables the trigger on an incorrect length of the sequence - when additional or missing bits are detected and the sequence of bits is not as expected.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorLength> ON | OFF
*RST: ON

**Manual operation:** See "Error" on page 638

---

**TRIGger<m>:ETHernet:ERRor:PREamble** <ErrorPreamble>

Enables the trigger on a frame with invalid preamble.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorPreamble> ON | OFF
*RST: ON

**Manual operation:** See "Error" on page 638

---

**TRIGger<m>:ETHernet:FRAMe:CCONdition** <CRCOperator>

Sets the operator to trigger on a specific frame check sequence value.

**Suffix:**
<m> 1..3

**Parameters:**
<CRCOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater Than, Greater than or equal. These conditions require a destination address to be set with `TRIGger<m>:ETHernet:FRAMe:CMIN`.

**INRange | OORange**
In range, Out of range. Set the minimum and maximum value of the range with `TRIGger<m>:ETHernet:FRAMe:CMIN` and `TRIGger<m>:ETHernet:FRAMe:CMAX`.

*RST: EQUal

**Manual operation:** See "Frame check" on page 638

---

**TRIGger<m>:ETHernet:FRAMe:CMAX** <CRCPatternMax>

Sets the end value of the frame check sequence range, if `TRIGger<m>:ETHernet:FRAMe:CCONdition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<CRCPatternMax>

**Manual operation:** See "Frame check" on page 638

---

**TRIGger<m>:ETHernet:FRAMe:CMIN** <CRCPatternMin>

Specifies a CRC pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<CRCPatternMin>

**Manual operation:** See "Frame check" on page 638

---

**TRIGger<m>:ETHernet:FRAMe:DCONdition** <DestAddrOptor>

Sets the destination address operator to trigger on a specific pattern or a range.

**Suffix:**
<m> 1..3

**Parameters:**
<DestAddrOptor> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater Than, Greater than or equal. These conditions require a destination address to be set with `TRIGger<m>:ETHernet:FRAMe:DMIN`.

**INRange | OORange**
In range, Out of range. Set the minimum and maximum value of the range with `TRIGger<m>:ETHernet:FRAMe:DMIN` and `TRIGger<m>:ETHernet:FRAMe:DMAX`.

*RST: EQUal

**Manual operation:** See "Destination Address" on page 637

---

**TRIGger<m>:ETHernet:FRAMe:DMAX** <DestAddrPattMax>

Sets the end value of a destination address range, if `TRIGger<m>:ETHernet:FRAMe:DMAX` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<DestAddrPattMax>

**Manual operation:** See "Destination Address" on page 637

---

**TRIGger<m>:ETHernet:FRAMe:DMIN** <DestAddrPattMin>

Specifies a destination address pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<DestAddrPattMin>

**Manual operation:** See "Destination Address" on page 637

---

**TRIGger<m>:ETHernet:FRAMe:SCONdition** <SrcAddrOperator>

Sets source address operator to trigger on a specific pattern or a range.

**Suffix:**
<m> 1..3

**Parameters:**
<SrcAddrOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater Than, Greater than or equal. These conditions require a destination address to be set with `TRIGger<m>:ETHernet:FRAMe:SMIN`.

**INRange | OORange**
In range, Out of range. Set the minimum and maximum value of the range with `TRIGger<m>:ETHernet:FRAMe:SMIN` and `TRIGger<m>:ETHernet:FRAMe:SMAX`.

*RST: EQUal

**Manual operation:** See "Source Address" on page 637

---

**TRIGger<m>:ETHernet:FRAMe:SMAX** <SrcAddrPattMax>

Sets the end value of a range, if the data condition is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<SrcAddrPattMax>

**Manual operation:** See "Source Address" on page 637

---

**TRIGger<m>:ETHernet:FRAMe:SMIN** <SrcAddrPattMin>

Specifies a source address pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<SrcAddrPattMin>

**Manual operation:** See "Source Address" on page 637

---

**TRIGger<m>:ETHernet:FRAMe:TCONdition** <TypeOperator>

Sets the type operator to trigger on a specific length/type value pattern or a range.

**Suffix:**
<m> 1..3

**Parameters:**
<TypeOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater Than, Greater than or equal. These conditions require a destination address to be set with `TRIGger<m>:ETHernet:FRAMe:TMIN`.

**INRange | OORange**
In range, Out of range. Set the minimum and maximum value of the range with `TRIGger<m>:ETHernet:FRAMe:TMIN` and `TRIGger<m>:ETHernet:FRAMe:TMAX`.

*RST: EQUal

**Manual operation:** See "Length/ Type" on page 638

---

**TRIGger<m>:ETHernet:FRAMe:TMAX** <TypePatternMax>

Sets the end value of a length/type range, if `TRIGger<m>:ETHernet:FRAMe:TCONdition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<TypePatternMax>

**Manual operation:** See "Length/ Type" on page 638

---

**TRIGger<m>:ETHernet:FRAMe:TMIN** <TypePatternMin>

Specifies a length/type value, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<TypePatternMin>

**Manual operation:** See "Length/ Type" on page 638

### 24.17.10.3 Decode results

In all `BUS<m>:ETHernet:WORD<n>` commands, the suffix <m> selects the serial bus and the suffix <n> selects the word number in the decode table.

As an example, a set of query commands for bus #1 and word #1 is shown in the following, together with examples for results of these queries:

- `BUS1:ETH:WCOunt?` !2
- `BUS1:ETH:WORD1:STATe?` !OK
- `BUS1:ETH:WORD1:STARt?` !-0.000135
- `BUS1:ETH:WORD1:STOP?` !-6.62e-5
- `BUS1:ETH:WORD1:DEST?` !FF:FF:FF:FF:FF:FF
- `BUS1:ETH:WORD1:SRC?` !0F:0E:0D:0C:0B:0A

- `BUS1:ETH:WORD1:DATA?` ![60]45003c3e6210...
- `BUS1:ETH:WORD1:TYPE?` !2048
- `BUS1:ETH:WORD1:CRC?` !-1821935433
- `BUS1:ETH:WORD1:SSYM?` !
- `BUS1:ETH:WORD1:DSYM?` !BroadCast
- `BUS1:ETH:WORD1:BYTE1:VAL?` !69
- `BUS1:ETH:WORD1:BYTE2:VAL?` !0

BUS<m>:ETHernet:WCOunt? ..... 1837
BUS<m>:ETHernet:WORD<n>:FTYPe? ..... 1837
BUS<m>:ETHernet:WORD<n>:STATe? ..... 1838
BUS<m>:ETHernet:WORD<n>:STARt? ..... 1838
BUS<m>:ETHernet:WORD<n>:STOP? ..... 1838
BUS<m>:ETHernet:WORD<n>:DESTaddress? ..... 1839
BUS<m>:ETHernet:WORD<n>:SRCaddress? ..... 1839
BUS<m>:ETHernet:WORD<n>:TYPE? ..... 1839
BUS<m>:ETHernet:WORD<n>:DATA? ..... 1840
BUS<m>:ETHernet:WORD<n>:CRC? ..... 1840
BUS<m>:ETHernet:WORD<n>:DSYMbol? ..... 1840
BUS<m>:ETHernet:WORD<n>:SSYMbol? ..... 1841
BUS<m>:ETHernet:WORD<n>:BITRate? ..... 1841
BUS<m>:ETHernet:WORD<n>:BYTE<o>:VALue? ..... 1841
BUS<m>:ETHernet:WORD<n>:NUMWords? ..... 1842

### BUS<m>:ETHernet:WCOunt?

Returns the frame count for the selected serial bus, i.e. the number of frames in the current acquisition. The result corresponds to the number of rows in the result table.

**Suffix:**
<m> 1..4

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

### BUS<m>:ETHernet:WORD<n>:FTYPe?

Returns the frame type of the selected word in the current acquisition.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<FrameType> MAC | IDLE | SLEep | EOS | UNKNown
*RST: MAC

**Usage:** Query only

---

**BUS<m>:ETHernet:WORD<n>:STATe?**

Returns the frame state of the selected word in the current acquisition.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<State> OK | ERR_PREAMBLE | ERR_LENGTH | UNCorrelated | INComplete

**OK**
No error detected

**ERR_PREAMBLE**
Error in the preamble of the selected word

**ERR_SFD**
Error in the start frame delimiter (SFD). The value of a correct SFD byte is 171. The SFD is transmitted LSB first.

**ERR_LENGTH**
Error in the number of bits in the selected word

*RST: OK

**Usage:** Query only

---

**BUS<m>:ETHernet:WORD<n>:STARt?**

Returns the frame start time of the selected word in the current acquisition.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
Default unit: s

**Usage:** Query only

---

**BUS<m>:ETHernet:WORD<n>:STOP?**

Returns the frame stop time of the selected word in the current acquisition.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
Default unit: s

**Usage:** Query only

---

**BUS<m>:ETHernet:WORD<n>:DESTaddress?**

Returns the destination address of the specified word.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<DestAddress> String parameter

**Usage:** Query only

---

**BUS<m>:ETHernet:WORD<n>:SRCaddress?**

Returns the source address of the specified word.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<SrcAddress> String parameter

**Usage:** Query only

---

**BUS<m>:ETHernet:WORD<n>:TYPE?**

The sub-protocol (e.g. HTML, video, etc.) determines what meaning this field has. Since the content of this data area is not decoded, the interpretation of the TYPE field is ambivalent.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Type> Returns the word type (specific for the sub-protocol), or the length of the selected word.
Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

### BUS<m>:ETHernet:WORD<n>:DATA?

Returns the number of word bytes in brackets [.] followed by the first six word bytes of data in hexadecimal format.

Use `BUS<m>:ETHernet:WORD<n>:BYTE<o>:VALue?` to access the word bytes.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Data> String parameter

**Example:**
```
BUS:ETHernet:WORD3:DATA?
<-- '[60]FF00FFFF1234'
```

**Usage:** Query only

### BUS<m>:ETHernet:WORD<n>:CRC?

Returns the Cyclic Redundancy Code (CRC, or frame check) checksum of the selected word.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<CRC> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

### BUS<m>:ETHernet:WORD<n>:DSYMbol?

Returns the symbolic label (or translation) of the destination address of the specified word, if the label list is enabled.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<DestTranslation> String parameter

**Usage:** Query only

**BUS<m>:ETHernet:WORD<n>:SSYMbol?**

Returns the symbolic label (or translation) of the source address of the specified word, if the label list is enabled.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<SrcTranslation> String parameter

**Usage:** Query only

**BUS<m>:ETHernet:WORD<n>:BITRate?**

Returns the primary bit rate.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<PrimaryBitRate> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

**BUS<m>:ETHernet:WORD<n>:BYTE<o>:VALue?**

BYTE returns all data of up to 1982 bytes (not just the first 5 or 6 bytes). This is also visible in the data table under "Show details".

**Suffix:**
<m> 1..4

<n> *

<o> *
Selects the byte number.

**Return values:**
<FrameByteValue> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:ETHernet:WORD<n>:NUMWords?**

Returns the number of words in the selected frame. The result corresponds to the "Number of Words" column in the results table.

**Suffix:**
<m> 1..4

<n> *
Frame index

**Return values:**
<NumWords> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

### 24.17.10.4 Search settings

SEARch:TRIGger:ETHernet:TYPE....1842
SEARch:TRIGger:ETHernet:FRAMe:DCONdition....1843
SEARch:TRIGger:ETHernet:FRAMe:DMIN....1843
SEARch:TRIGger:ETHernet:FRAMe:DMAX....1843
SEARch:TRIGger:ETHernet:FRAMe:SCONdition....1844
SEARch:TRIGger:ETHernet:FRAMe:SMIN....1844
SEARch:TRIGger:ETHernet:FRAMe:SMAX....1844
SEARch:TRIGger:ETHernet:FRAMe:TCONdition....1845
SEARch:TRIGger:ETHernet:FRAMe:TMIN....1845
SEARch:TRIGger:ETHernet:FRAMe:TMAX....1846
SEARch:TRIGger:ETHernet:FRAMe:CCONdition....1846
SEARch:TRIGger:ETHernet:FRAMe:CMIN....1846
SEARch:TRIGger:ETHernet:FRAMe:CMAX....1847
SEARch:TRIGger:ETHernet:ERRor:PREamble....1847
SEARch:TRIGger:ETHernet:ERRor:LENGth....1847

**SEARch:TRIGger:ETHernet:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:ETHernet:TYPE?** <SearchName>

Selects, if it searched on a MAC frame or on an error.

**Parameters:**
<Type> FRAMe | ERRor
*RST: FRAMe

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Type" on page 641

**SEARch:TRIGger:ETHernet:FRAMe:DCONdition** <SearchName>,<DestAddrOptor>
**SEARch:TRIGger:ETHernet:FRAMe:DCONdition?** <SearchName>

Defines the operator to search a specific destination address within a frame.

**Parameters:**
<DestAddrOptor> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater Than, Greater than or equal. These conditions require a destination address to be set with `SEARch:TRIGger:ETHernet:FRAMe:DMIN`.

**INRange | OORange**
In range, Out of range. Set the minimum and maximum value of the range with `SEARch:TRIGger:ETHernet:FRAMe:DMIN` and `SEARch:TRIGger:ETHernet:FRAMe:DMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Destination address" on page 642

**SEARch:TRIGger:ETHernet:FRAMe:DMIN** <SearchName>,<DestAddrPatt>
**SEARch:TRIGger:ETHernet:FRAMe:DMIN?** <SearchName>

Defines a destination address, or sets the start value of a destination address range.

**Parameters:**
<DestAddrPatt> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Destination address" on page 642

**SEARch:TRIGger:ETHernet:FRAMe:DMAX** <SearchName>,<DestAddrPatt>
**SEARch:TRIGger:ETHernet:FRAMe:DMAX?** <SearchName>

Sets the end value of a destination address range, if `SEARch:TRIGger:ETHernet:FRAMe:DCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<DestAddrPatt> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Destination address" on page 642

---

**SEARch:TRIGger:ETHernet:FRAMe:SCONdition**
<SearchName>,<SrcAddrOperator>
**SEARch:TRIGger:ETHernet:FRAMe:SCONdition?** <SearchName>

Defines the operator to search a specific source address within a frame.

**Parameters:**
<SrcAddrOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater Than, Greater than or equal. These conditions require a destination address to be set with `SEARch:TRIGger:ETHernet:FRAMe:SMIN`.

**INRange | OORange**
In range, Out of range. Set the minimum and maximum value of the range with `SEARch:TRIGger:ETHernet:FRAMe:SMIN` and `SEARch:TRIGger:ETHernet:FRAMe:SMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Source address" on page 642

---

**SEARch:TRIGger:ETHernet:FRAMe:SMIN** <SearchName>,<SrcAddrPatt>
**SEARch:TRIGger:ETHernet:FRAMe:SMIN?** <SearchName>

Defines a source address, or sets the start value of a source address range.

**Parameters:**
<SrcAddrPatt> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Source address" on page 642

---

**SEARch:TRIGger:ETHernet:FRAMe:SMAX** <SearchName>,<SrcAddrPatt>
**SEARch:TRIGger:ETHernet:FRAMe:SMAX?** <SearchName>

Sets the end value of a source address range, if `SEARch:TRIGger:ETHernet:FRAMe:SCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<SrcAddrPatt> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Source address" on page 642

---

**SEARch:TRIGger:ETHernet:FRAMe:TCONdition** <SearchName>,<TypeOperator>
**SEARch:TRIGger:ETHernet:FRAMe:TCONdition?** <SearchName>

Defines the operator to search for a specific frame length or type.

**Parameters:**
<TypeOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater Than, Greater than or equal. These conditions require a pattern to be set with `SEARch:TRIGger:ETHernet:FRAMe:TMIN`.

**INRange | OORange**
In range, Out of range. Set the minimum and maximum value of the range with `SEARch:TRIGger:ETHernet:FRAMe:TMIN` and `SEARch:TRIGger:ETHernet:FRAMe:TMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Length/Type" on page 642

---

**SEARch:TRIGger:ETHernet:FRAMe:TMIN** <SearchName>,<TypePatternMin>
**SEARch:TRIGger:ETHernet:FRAMe:TMIN?** <SearchName>

Defines a frame length/type, or sets the start value for a range of frame lengths/types.

**Parameters:**
<TypePatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Length/Type" on page 642

**SEARch:TRIGger:ETHernet:FRAMe:TMAX** <SearchName>,<TypePatternMax>
**SEARch:TRIGger:ETHernet:FRAMe:TMAX?** <SearchName>

Sets the end value of a range of frame lengths/types, if `SEARch:TRIGger:ETHernet:FRAMe:TCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<TypePatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Length/Type" on page 642

**SEARch:TRIGger:ETHernet:FRAMe:CCONdition** <SearchName>,<CRCOperator>
**SEARch:TRIGger:ETHernet:FRAMe:CCONdition?** <SearchName>

Defines the operator to search for a Cyclic Redundancy Code (CRC, or frame check) error condition within a frame.

**Parameters:**
<CRCOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater Than, Greater than or equal. These conditions require a CRC pattern to be set with `SEARch:TRIGger:ETHernet:FRAMe:CMIN`.

**INRange | OORange**
In range, Out of range. Set the minimum and maximum value of the range with `SEARch:TRIGger:ETHernet:FRAMe:CMIN` and `SEARch:TRIGger:ETHernet:FRAMe:CMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Frame check" on page 642

**SEARch:TRIGger:ETHernet:FRAMe:CMIN** <SearchName>,<CRCPatternMin>
**SEARch:TRIGger:ETHernet:FRAMe:CMIN?** <SearchName>

Defines a CRC error condition pattern, or sets the start value of such a pattern.

**Parameters:**
<CRCPatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Frame check" on page 642

---

**SEARch:TRIGger:ETHernet:FRAMe:CMAX** <SearchName>,<CRCPatternMax>
**SEARch:TRIGger:ETHernet:FRAMe:CMAX?** <SearchName>

Sets the end value of a CRC error condition pattern, if `SEARch:TRIGger:ETHernet:FRAMe:CCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<CRCPatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Frame check" on page 642

---

**SEARch:TRIGger:ETHernet:ERRor:PREamble** <SearchName>,<ErrorPreamble>
**SEARch:TRIGger:ETHernet:ERRor:PREamble?** <SearchName>

Defines, whether a search for any preamble error shall be activated or not.

**Parameters:**
<ErrorPreamble> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Preamble Error" on page 643

---

**SEARch:TRIGger:ETHernet:ERRor:LENGth** <SearchName>,<ErrorLength>
**SEARch:TRIGger:ETHernet:ERRor:LENGth?** <SearchName>

Defines, whether a search for any word length error (too few or too many bits per word) shall be activated or not.

**Parameters:**
<ErrorLength> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Length Error" on page 643

### 24.17.10.5 Search results

To show the results on the screen, use the following commands:

- `BUS<m>:RESult` on page 1638

In all `SEARch:RESult:ETHernet:WORD<m>` commands, the suffix <m> selects the word number in the list of search results.

SEARch:RESult:ETHernet:WCOunt? 1848
SEARch:RESult:ETHernet:WORD<m>:STATe? 1848
SEARch:RESult:ETHernet:WORD<m>:STARt? 1849
SEARch:RESult:ETHernet:WORD<m>:STOP? 1849
SEARch:RESult:ETHernet:WORD<m>:DESTaddress? 1849
SEARch:RESult:ETHernet:WORD<m>:SRCaddress? 1850
SEARch:RESult:ETHernet:WORD<m>:TYPE? 1850
SEARch:RESult:ETHernet:WORD<m>:FTYPe? 1850
SEARch:RESult:ETHernet:WORD<m>:DATA? 1850
SEARch:RESult:ETHernet:WORD<m>:CRC? 1851
SEARch:RESult:ETHernet:WORD<m>:DSYMbol? 1851
SEARch:RESult:ETHernet:WORD<m>:SSYMbol? 1851
SEARch:RESult:ETHernet:WORD<m>:BYTE<n>:VALue? 1852

**SEARch:RESult:ETHernet:WCOunt?** <SearchName>

Returns the number of decoded words within the search result.

**Query parameters:**

<SearchName> String parameter

**Return values:**

<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:ETHernet:WORD<m>:STATe?** <SearchName>

Returns the frame state of the selected word within the search result.

**Suffix:**

<m> *

**Query parameters:**

<SearchName> String parameter

**Return values:**

<State> OK | ERR_PREAMBLE | ERR_LENGTH

**OK**
No error detected

**ERR_PREAMBLE**
Error in the preamble of the selected word

**ERR_SFD**
Error in the start frame delimiter (SFD). The value of a correct SFD byte is 171. The SFD is transmitted LSB first.

**ERR_LENGTH**
Error in the number of bits in the selected word

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:ETHernet:WORD<m>:STARt?** <SearchName>

Returns the frame start time of the selected word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName> String parameter

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:ETHernet:WORD<m>:STOP?** <SearchName>

Returns the frame stop time of the selected word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName> String parameter

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:ETHernet:WORD<m>:DESTaddress?** <SearchName>

Returns the destination address of the specified word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName> String parameter

**Return values:**
<DestAddress> String parameter

**Usage:** Query only

**SEARch:RESult:ETHernet:WORD<m>:SRCaddress?** <SearchName>

Returns the source address of the specified word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName> String parameter

**Return values:**
<SourceAddress> String parameter

**Usage:** Query only

**SEARch:RESult:ETHernet:WORD<m>:TYPE?** <SearchName>

The sub-protocol (e.g. HTML, video, etc.) determines what meaning this field has. Since the content of this data area is not decoded, the interpretation of the TYPE field is ambivalent. The query either returns the word type (specific for the sub-protocol), or the length of the selected word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName> String parameter

**Return values:**
<Type> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:ETHernet:WORD<m>:FTYPe?** <SearchName>

Returns the frame type of the specified frame.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameType> MAC | IDLE | SLEep | EOS | UNKNown
*RST: MAC

**Usage:** Query only

**SEARch:RESult:ETHernet:WORD<m>:DATA?** <SearchName>

Returns the data bytes of the specified word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName> String parameter

**Return values:**
<Data> String parameter

**Usage:** Query only

---

**SEARch:RESult:ETHernet:WORD<m>:CRC?** <SearchName>

Returns the Cyclic Redundancy Code (CRC, or frame check) checksum of the selected word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName> String parameter

**Return values:**
<CRC> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:ETHernet:WORD<m>:DSYMbol?** <SearchName>

Returns the symbolic label (or translation) of the destination address of the specified word within the search result, if the label list is enabled.

**Suffix:**
<m> *

**Query parameters:**
<SearchName> String parameter

**Return values:**
<DestTranslation> String parameter

**Usage:** Query only

---

**SEARch:RESult:ETHernet:WORD<m>:SSYMbol?** <SearchName>

Returns the symbolic label (or translation) of the source address of the specified word within the search result, if the label list is enabled.

**Suffix:**
<m> *

**Query parameters:**
<SearchName> String parameter

**Return values:**
<SrcTranslation> String parameter

**Usage:** Query only

---

**SEARch:RESult:ETHernet:WORD<m>:BYTE<n>:VALue?** <SearchName>

BYTE returns all data of up to 1982 bytes (not just the first 5 or 6 bytes). This is also visible in the data table under "Show details".

**Suffix:**
<m> *

<n> *
Selects the byte number.

**Query parameters:**
<SearchName> String parameter

**Return values:**
<FrameByteValue> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

## 24.17.11 Ethernet 100BASE-T1 (option R&S RTP-57)


- Configuration....1852
- Trigger....1856
- Decode results....1864
- Search settings....1870
- Search results....1878


### 24.17.11.1 Configuration

In all `BUS<m>:HBTO` commands, the suffix <m> selects the serial bus.


BUS<m>:HBTO:ATTN....1853
BUS<m>:HBTO:FDAM....1853
BUS<m>:HBTO:FDAP....1853
BUS<m>:HBTO:FDIF....1853
BUS<m>:HBTO:FTYP....1854
BUS<m>:HBTO:RDAM....1854
BUS<m>:HBTO:RDAP....1854
BUS<m>:HBTO:RDIF....1854
BUS<m>:HBTO:RTYP....1855
BUS<m>:HBTO:THReshold....1855
BUS<m>:HBTO:POLarity....1855
BUS<m>:HBTO:MODE....1855
BUS<m>:HBTO:SYMRate....1856

**BUS<m>:HBTO:ATTN** <Attenuation>

Sets the attenuation factor. It is used to de-amplify the reverse signal before subtracting it from the forward signal.

**Suffix:**
<m> 1..4

**Parameters:**
<Attenuation> Range: -100 to 0
Increment: 0.1
*RST: -26
Default unit: dB

**Manual operation:** See "Reverse attenuation" on page 647

**BUS<m>:HBTO:FDAM** <SourceDAminus>

Selects the DA- source of the provided forward single ended signal, if `BUS<m>:HBTO:FTYP` is set to `SINGle`.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceDAminus> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: C2W1

**Manual operation:** See "Forward" on page 647

**BUS<m>:HBTO:FDAP** <SourceDAplus>

Selects the DA+ source of the provided forward single ended signal, if `BUS<m>:HBTO:FTYP` is set to `SINGle`.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceDAplus> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: C1W1

**Manual operation:** See "Forward" on page 647

**BUS<m>:HBTO:FDIF** <SrcDiff>

Sets the source of the provided forward differential signal, if `BUS<m>:HBTO:FTYP` is set to `DIFFerential`.

**Suffix:**
<m> 1..4

**Parameters:**
<SrcDiff> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: C1W1

**Manual operation:** See "Forward" on page 647

---

**BUS<m>:HBTO:FTYP** <SignalType>

Sets the type of forward signal measurement.

**Suffix:**
<m> 1..4

**Parameters:**
<SignalType> DIFFerential | SINGle
*RST: DIFFerential

**Manual operation:** See "Forward" on page 647

---

**BUS<m>:HBTO:RDAM** <SrcRevDAminus>

Selects the DA- source of the provided reversed single ended signal, if `BUS<m>:HBTO:RTYP` is set to `SINGle`.

**Suffix:**
<m> 1..4

**Parameters:**
<SrcRevDAminus> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

**Manual operation:** See "Reverse" on page 647

---

**BUS<m>:HBTO:RDAP** <SourceRevDAplus>

Selects the DA+ source of the provided reversed single ended signal, if `BUS<m>:HBTO:RTYP` is set to `SINGle`.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceRevDAplus> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

**Manual operation:** See "Reverse" on page 647

---

**BUS<m>:HBTO:RDIF** <SrcRevDiff>

Sets the source of the provided reversed differential signal, if `BUS<m>:HBTO:RTYP` is set to `DIFFerential`.

**Suffix:**
<m> 1..4

**Parameters:**
<SrcRevDiff> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

**Manual operation:** See "Reverse" on page 647

---

**BUS<m>:HBTO:RTYP** <SignalRevType>

Sets the type of reverse signal measurement.

**Suffix:**
<m> 1..4

**Parameters:**
<SignalRevType> DISabled | DIFFerential | SINGle
*RST: DISabled

**Manual operation:** See "Reverse" on page 647

---

**BUS<m>:HBTO:THReshold** <ThresholdUpper>

Sets an user-defined threshold value.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdUpper> Range: 0 to 2
Increment: 1E-3
*RST: 0.4
Default unit: V

**Manual operation:** See "Thresholds" on page 647

---

**BUS<m>:HBTO:POLarity** <Polarity>

Selects the polarity of the data signal.

**Suffix:**
<m> 1..4

**Parameters:**
<Polarity> NORMal | INVert
*RST: NORMal

**Manual operation:** See "Polarity" on page 647

---

**BUS<m>:HBTO:MODE** <Mode>

Selects the direction of the full-duplex signal you want to look at.

**Suffix:**
<m> 1..4

**Parameters:**
<Mode> MASTer | SLAVe | AUTO
*RST: AUTO

**Manual operation:** See "Operation mode" on page 648

---

**BUS<m>:HBTO:SYMRate** <SymbolRate>

Defines the rate of ternary symbols.

**Suffix:**
<m> 1..4

**Parameters:**
<SymbolRate> Range: 10 to 150
Increment: 1E-3
*RST: 66.6667
Default unit: MSymb/s

**Manual operation:** See "Symbol rate" on page 648

### 24.17.11.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:HBTO:TYPE ..... 1857
TRIGger<m>:HBTO:DADDress:CONDition ..... 1857
TRIGger<m>:HBTO:DADDress:MIN ..... 1858
TRIGger<m>:HBTO:DADDress:MAX ..... 1858
TRIGger<m>:HBTO:SADDress:CONDition ..... 1858
TRIGger<m>:HBTO:SADDress:MIN ..... 1859
TRIGger<m>:HBTO:SADDress:MAX ..... 1859
TRIGger<m>:HBTO:LENGth:CONDition ..... 1859
TRIGger<m>:HBTO:LENGth:MIN ..... 1860
TRIGger<m>:HBTO:LENGth:MAX ..... 1860
TRIGger<m>:HBTO:CRC:CONDition ..... 1860
TRIGger<m>:HBTO:CRC:MIN ..... 1861
TRIGger<m>:HBTO:CRC:MAX ..... 1861
TRIGger<m>:HBTO:DATA:DCONdition ..... 1861
TRIGger<m>:HBTO:DATA:DMIN ..... 1862
TRIGger<m>:HBTO:DATA:DMAX ..... 1862

TRIGger<m>:HBTO:DATA:ICONdition....1862
TRIGger<m>:HBTO:DATA:IMIN....1863
TRIGger<m>:HBTO:DATA:IMAX....1863
TRIGger<m>:HBTO:ERRor:PREamble....1863
TRIGger<m>:HBTO:ERRor:CRC....1863
TRIGger<m>:HBTO:ERRor:SFD....1864

**TRIGger<m>:HBTO:TYPE** <Type>

Selects the type of frame to be triggered on.

**Suffix:**
<m> 1..3

**Parameters:**
<Type> STARt | MAC | IDLE | ERRor

**STARt**
Start of frame.

**MAC**
MAC frame. This frame contains information that define how to go about transmitting and receiving frames.

**IDLE**
IDLE frame. This frame is used for clock synchronization.

**ERRor**
Error frame. Thi frame contains erroneous bits.

*RST: STARt

**Manual operation:** See "Type" on page 651

**TRIGger<m>:HBTO:DADDress:CONDition** <DestAddrOptor>

Sets the condition for the destination address. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<DestAddrOptor> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:HBTO:DADDress:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:HBTO:DADDress:MIN` and `TRIGger<m>:HBTO:DADDress:MAX`.

*RST: EQUal

**Manual operation:** See "Destination address" on page 652

---

**TRIGger<m>:HBTO:DADDress:MIN** <DestAddrPattMin>

Specifies the destination address bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<DestAddrPattMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Destination address" on page 652

---

**TRIGger<m>:HBTO:DADDress:MAX** <DestAddrPattMax>

Sets the end value of the destination address range if `TRIGger<m>:HBTO:DADDress:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<DestAddrPattMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Destination address" on page 652

---

**TRIGger<m>:HBTO:SADDress:CONDition** <SrcAddrOperator>

Sets the condition for the source address. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<SrcAddrOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:HBTO:SADDress:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:HBTO:SADDress:MIN` and `TRIGger<m>:HBTO:SADDress:MAX`.

*RST: EQUal

**Manual operation:** See "Source address" on page 653

---

**TRIGger<m>:HBTO:SADDress:MIN** <SrcAddrPattMin>

Specifies the source address bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<SrcAddrPattMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Source address" on page 653

---

**TRIGger<m>:HBTO:SADDress:MAX** <SrcAddrPattMax>

Sets the end value of the source address range if `TRIGger<m>:HBTO:SADDress:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<SrcAddrPattMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Source address" on page 653

---

**TRIGger<m>:HBTO:LENGth:CONDition** <TypeOperator>

Sets the condition for the length / type. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<TypeOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:HBTO:LENGth:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:HBTO:LENGth:MIN` and `TRIGger<m>:HBTO:LENGth:MAX`.

*RST: EQUal

**Manual operation:** See "Length/Type" on page 653

---

**TRIGger<m>:HBTO:LENGth:MIN** <TypePatternMin>

Specifies the length / type bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<TypePatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Length/Type" on page 653

---

**TRIGger<m>:HBTO:LENGth:MAX** <TypePatternMax>

Sets the end value of the length / type range if `TRIGger<m>:HBTO:LENGth:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<TypePatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Length/Type" on page 653

---

**TRIGger<m>:HBTO:CRC:CONDition** <CRCOperator>

Sets the condition for the frame check. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<CRCOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:HBTO:CRC:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:HBTO:CRC:MIN` and `TRIGger<m>:HBTO:CRC:MAX`.

*RST: EQUal

**Manual operation:** See "Frame check" on page 654

---

**TRIGger<m>:HBTO:CRC:MIN** <CRCPatternMin>

Specifies the frame check bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<CRCPatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Frame check" on page 654

---

**TRIGger<m>:HBTO:CRC:MAX** <CRCPatternMax>

Sets the end value of the frame check range if `TRIGger<m>:HBTO:CRC:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<CRCPatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Frame check" on page 654

---

**TRIGger<m>:HBTO:DATA:DCONdition** <DataOperator>

Sets the condition for the data. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:HBTO:DATA:DMIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:HBTO:DATA:DMIN` and `TRIGger<m>:HBTO:DATA:DMAX`.

*RST: EQUal

**Manual operation:** See "Data" on page 654

---

**TRIGger<m>:HBTO:DATA:DMIN** <DataMin>

Specifies the data bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data" on page 654

---

**TRIGger<m>:HBTO:DATA:DMAX** <DataMax>

Sets the end value of the data range if `TRIGger<m>:HBTO:DATA:DCONdition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<DataMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data" on page 654

---

**TRIGger<m>:HBTO:DATA:ICONdition** <DataIdxOperator>

Sets the condition for the index. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataIdxOperator> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:HBTO:DATA:IMIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:HBTO:DATA:IMIN` and `TRIGger<m>:HBTO:DATA:IMAX`.

*RST: INRange

**Manual operation:** See "Index" on page 655

---

**TRIGger<m>:HBTO:DATA:IMIN** <DataIndexMin>

Specifies the index bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataIndexMin> Range: 1 to 65535
Increment: 1
*RST: 1

**Manual operation:** See "Index" on page 655

---

**TRIGger<m>:HBTO:DATA:IMAX** <DataIndexMax>

Sets the end value of the index range if `TRIGger<m>:HBTO:DATA:ICONdition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<DataIndexMax> Range: 1 to 65535
Increment: 1
*RST: 0

**Manual operation:** See "Index" on page 655

---

**TRIGger<m>:HBTO:ERRor:PREamble** <ErrorPreamble>

Enables / disables trigger on preamble error.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorPreamble> ON | OFF
*RST: ON

**Manual operation:** See "Preamble Error" on page 655

---

**TRIGger<m>:HBTO:ERRor:CRC** <ErrorCRC>

Enables / disables trigger on Cyclic Redundancy Check (CRC) error.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorCRC> ON | OFF
*RST: ON

**Manual operation:** See "CRC Error" on page 655

---

**TRIGger<m>:HBTO:ERRor:SFD** <ErrorSFD>

Enables / disables trigger on start frame delimiter (SFD) error.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorSFD> ON | OFF
*RST: ON

**Manual operation:** See "SFD Error" on page 655

### 24.17.11.3 Decode results

To show the results on the screen, use the following commands:

- `BUS<m>:RESult` on page 1638

In all `BUS<m>:HBTO:RESult:FRAMe<n>:WORD<o>` commands, the suffix <m> selects the serial bus, suffix <n> selects the frame in the decode table and suffix <o> selects the word in the selected frame.

BUS<m>:HBTO:RESult:FCOunt? 1864
BUS<m>:HBTO:RESult:FRAMe<n>:FTYPe? 1865
BUS<m>:HBTO:RESult:FRAMe<n>:STATe? 1865
BUS<m>:HBTO:RESult:FRAMe<n>:STARt? 1866
BUS<m>:HBTO:RESult:FRAMe<n>:STOP? 1866
BUS<m>:HBTO:RESult:FRAMe<n>:DESTaddress? 1866
BUS<m>:HBTO:RESult:FRAMe<n>:SRCaddress? 1867
BUS<m>:HBTO:RESult:FRAMe<n>:TYPE? 1867
BUS<m>:HBTO:RESult:FRAMe<n>:DATA? 1867
BUS<m>:HBTO:RESult:FRAMe<n>:CRC? 1867
BUS<m>:HBTO:RESult:FRAMe<n>:NUMWords? 1868
BUS<m>:HBTO:RESult:FRAMe<n>:DSYMbol? 1868
BUS<m>:HBTO:RESult:FRAMe<n>:SSYMbol? 1868
BUS<m>:HBTO:RESult:FRAMe<n>:BITRate? 1868
BUS<m>:HBTO:RESult:FRAMe<n>:WORD<o>:TYPE? 1869
BUS<m>:HBTO:RESult:FRAMe<n>:WORD<o>:VALue? 1869
BUS<m>:HBTO:RESult:FRAMe<n>:WORD<o>:VSTR? 1869

---

**BUS<m>:HBTO:RESult:FCOunt?**

Returns the number of decoded frames.

**Suffix:**
<m> 1..4

**Return values:**

<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:HBTO:RESult:FRAMe<n>:FTYPe?**

Returns the type of frame for the selected frame.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<FrameType> IDLE | MAC | FILLer | UNKNown

**IDLE**
IDLE frame. This frame is used for clock synchronization.

**MAC**
MAC frame. This frame contains information that define how to go about transmitting and receiving frames.

**FILLer**
Filler frame. The frame is used to maintain transmission activity.

**UNKNown**
No meaningful frame can be determined.

*RST: MAC

**Usage:** Query only

---

**BUS<m>:HBTO:RESult:FRAMe<n>:STATe?**

Returns the state of the frame.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<State> OK | ERR_PREAMBLE | ERR_SFD | ERR_LENGTH | ERR_CRC | UNCorrelated | INComplete

**OK**
Valid frame.

**ERR_PREAMBLE**
Erroneous frame due to preamble error.

**ERR_SFD**
Erroneous frame due to SFD error.

**ERR_LENGTH**
Erroneous frame due to length / type error.

**ERR_CRC**
Erroneous frame due to CRC error.

*RST: OK

**Usage:** Query only

---

**BUS<m>:HBTO:RESult:FRAMe<n>:STARt?**

Returns the start time of the selected frame.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:HBTO:RESult:FRAMe<n>:STOP?**

Returns the end time of the selected frame.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:HBTO:RESult:FRAMe<n>:DESTaddress?**

Returns the destination address of the selected frame.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<DestAddress>

**Usage:** Query only

**BUS<m>:HBTO:RESult:FRAMe<n>:SRCaddress?**

Returns the source address of the selected frame.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<SrcAddress>

**Usage:** Query only

**BUS<m>:HBTO:RESult:FRAMe<n>:TYPE?**

Returns the value of length / type field of the selected frame.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<TypeLen> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:HBTO:RESult:FRAMe<n>:DATA?**

Returns the data for the selected frame, corresponds to the Data column in the decode results table.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Data>

**Usage:** Query only

**BUS<m>:HBTO:RESult:FRAMe<n>:CRC?**

Returns the CRC (frame check) checksum of the selected frame.

**Suffix:**
<m> 1..4

<n> *

**Return values:**

<CRC> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:HBTO:RESult:FRAMe<n>:NUMWords?**

Returns the number of decoded words for the selected frame.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<NumWords> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:HBTO:RESult:FRAMe<n>:DSYMbol?**

Returns the destination symbols of the selected frame.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<DestTranslation>

**Usage:** Query only

---

**BUS<m>:HBTO:RESult:FRAMe<n>:SSYMbol?**

Returns the source symbols of the selected frame.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<SrcTranslation>

**Usage:** Query only

---

**BUS<m>:HBTO:RESult:FRAMe<n>:BITRate?**

Returns the primary bit rate.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<PrimaryBitRate> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**BUS<m>:HBTO:RESult:FRAMe<n>:WORD<o>:TYPE?**

Returns the data type for the selected word.

**Suffix:**
<m> 1..4

<n> *

<o> *

**Return values:**
<WordType>

**Usage:** Query only

---

**BUS<m>:HBTO:RESult:FRAMe<n>:WORD<o>:VALue?**

Returns the data value for the selected word.

**Suffix:**
<m> 1..4

<n> *

<o> *

**Return values:**
<WordValue> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:HBTO:RESult:FRAMe<n>:WORD<o>:VSTR?**

Returns the string equivalent of data value for the selected cell.

**Suffix:**
<m> 1..4

<n> *

<o> *

**Return values:**
<WordValueString> String parameter

**Usage:** Query only

### 24.17.11.4 Search settings

SEARch:TRIGger:HBTO:TYPE ... 1870
SEARch:TRIGger:HBTO:DADDress:CONDition ... 1871
SEARch:TRIGger:HBTO:DADDress:MIN ... 1871
SEARch:TRIGger:HBTO:DADDress:MAX ... 1871
SEARch:TRIGger:HBTO:SADDress:CONDition ... 1872
SEARch:TRIGger:HBTO:SADDress:MIN ... 1872
SEARch:TRIGger:HBTO:SADDress:MAX ... 1873
SEARch:TRIGger:HBTO:LENGth:CONDition ... 1873
SEARch:TRIGger:HBTO:LENGth:MIN ... 1873
SEARch:TRIGger:HBTO:LENGth:MAX ... 1874
SEARch:TRIGger:HBTO:CRC:CONDition ... 1874
SEARch:TRIGger:HBTO:CRC:MIN ... 1874
SEARch:TRIGger:HBTO:CRC:MAX ... 1875
SEARch:TRIGger:HBTO:DATA:DCONdition ... 1875
SEARch:TRIGger:HBTO:DATA:DMIN ... 1875
SEARch:TRIGger:HBTO:DATA:DMAX ... 1876
SEARch:TRIGger:HBTO:DATA:ICONdition ... 1876
SEARch:TRIGger:HBTO:DATA:IMIN ... 1877
SEARch:TRIGger:HBTO:DATA:IMAX ... 1877
SEARch:TRIGger:HBTO:ERRor:PREamble ... 1877
SEARch:TRIGger:HBTO:ERRor:CRC ... 1877
SEARch:TRIGger:HBTO:ERRor:SFD ... 1878

---

**SEARch:TRIGger:HBTO:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:HBTO:TYPE?** <SearchName>

Selects the type of frame to be searched for.

**Parameters:**
<Type> STARt | MAC | IDLE | ERRor

**STARt**
Start of frame.

**MAC**
MAC frame. This frame contains information that define how to go about transmitting and receiving frames.

**IDLE**
IDLE frame. This frame is used for clock synchronization.

**ERRor**
Error frame. Thi frame contains erroneous bits.

*RST: STARt

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Type" on page 658

---

**SEARch:TRIGger:HBTO:DADDress:CONDition** <SearchName>,<DestAddrOptor>
**SEARch:TRIGger:HBTO:DADDress:CONDition?** <SearchName>

Sets the condition for the destination address. You can define an exact value or a value range.

**Parameters:**
<DestAddrOptor> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:HBTO:DADDress:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:HBTO:DADDress:MIN` and `SEARch:TRIGger:HBTO:DADDress:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Destination address" on page 659

---

**SEARch:TRIGger:HBTO:DADDress:MIN** <SearchName>,<DestAddrPattMin>
**SEARch:TRIGger:HBTO:DADDress:MIN?** <SearchName>

Specifies the destination address bit pattern, or sets the start value of a pattern range.

**Parameters:**
<DestAddrPattMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Destination address" on page 659

---

**SEARch:TRIGger:HBTO:DADDress:MAX** <SearchName>,<DestAddrPattMax>
**SEARch:TRIGger:HBTO:DADDress:MAX?** <SearchName>

Sets the end value of the destination address range if `SEARch:TRIGger:HBTO:DADDress:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<DestAddrPattMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Destination address" on page 659

---

**SEARch:TRIGger:HBTO:SADDress:CONDition** <SearchName>,<SrcAddrOperator>
**SEARch:TRIGger:HBTO:SADDress:CONDition?** <SearchName>

Sets the condition for the source address. You can define an exact value or a value range.

**Parameters:**
<SrcAddrOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:HBTO:SADDress:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:HBTO:SADDress:MIN` and `SEARch:TRIGger:HBTO:SADDress:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Source address" on page 659

---

**SEARch:TRIGger:HBTO:SADDress:MIN** <SearchName>,<SrcAddrPattMin>
**SEARch:TRIGger:HBTO:SADDress:MIN?** <SearchName>

Specifies the source address bit pattern, or sets the start value of a pattern range.

**Parameters:**
<SrcAddrPattMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Source address" on page 659

**SEARch:TRIGger:HBTO:SADDress:MAX** <SearchName>,<SrcAddrPattMax>
**SEARch:TRIGger:HBTO:SADDress:MAX?** <SearchName>

Sets the end value of the source address range if `SEARch:TRIGger:HBTO:SADDress:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<SrcAddrPattMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Source address" on page 659

**SEARch:TRIGger:HBTO:LENGth:CONDition** <SearchName>,<TypeOperator>
**SEARch:TRIGger:HBTO:LENGth:CONDition?** <SearchName>

Sets the condition for the length / type. You can define an exact value or a value range.

**Parameters:**
<TypeOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:HBTO:LENGth:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:HBTO:LENGth:MIN` and `SEARch:TRIGger:HBTO:LENGth:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Length/Type" on page 659

**SEARch:TRIGger:HBTO:LENGth:MIN** <SearchName>,<TypePatternMin>
**SEARch:TRIGger:HBTO:LENGth:MIN?** <SearchName>

Specifies the length / type bit pattern, or sets the start value of a pattern range.

**Parameters:**
<TypePatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Length/Type" on page 659

---

**SEARch:TRIGger:HBTO:LENGth:MAX** <SearchName>,<TypePatternMax>
**SEARch:TRIGger:HBTO:LENGth:MAX?** <SearchName>

Sets the end value of the length / type range if `SEARch:TRIGger:HBTO:LENGth:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<TypePatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Length/Type" on page 659

---

**SEARch:TRIGger:HBTO:CRC:CONDition** <SearchName>,<CRCOperator>
**SEARch:TRIGger:HBTO:CRC:CONDition?** <SearchName>

Sets the condition for the frame check. You can define an exact value or a value range.

**Parameters:**
<CRCOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:HBTO:CRC:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:HBTO:CRC:MIN` and `SEARch:TRIGger:HBTO:CRC:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame check" on page 660

---

**SEARch:TRIGger:HBTO:CRC:MIN** <SearchName>,<CRCPatternMin>
**SEARch:TRIGger:HBTO:CRC:MIN?** <SearchName>

Specifies the frame check bit pattern, or sets the start value of a pattern range.

**Parameters:**
<CRCPatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame check" on page 660

---

**SEARch:TRIGger:HBTO:CRC:MAX** <SearchName>,<CRCPatternMax>
**SEARch:TRIGger:HBTO:CRC:MAX?** <SearchName>

Sets the end value of the frame check range if `SEARch:TRIGger:HBTO:CRC:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<CRCPatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame check" on page 660

---

**SEARch:TRIGger:HBTO:DATA:DCONdition** <SearchName>,<DataOperator>
**SEARch:TRIGger:HBTO:DATA:DCONdition?** <SearchName>

Sets the condition for the data. You can define an exact value or a value range.

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:HBTO:DATA:DMIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:HBTO:DATA:DMIN` and `SEARch:TRIGger:HBTO:DATA:DMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data" on page 660

---

**SEARch:TRIGger:HBTO:DATA:DMIN** <SearchName>,<DataMin>
**SEARch:TRIGger:HBTO:DATA:DMIN?** <SearchName>

Specifies the data bit pattern, or sets the start value of a pattern range.

**Parameters:**
<DataMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data" on page 660

---

**SEARch:TRIGger:HBTO:DATA:DMAX** <SearchName>,<DataMax>
**SEARch:TRIGger:HBTO:DATA:DMAX?** <SearchName>

Sets the end value of the data range if `SEARch:TRIGger:HBTO:DATA:DCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<DataMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data" on page 660

---

**SEARch:TRIGger:HBTO:DATA:ICONdition** <SearchName>,<DataIdxOperator>
**SEARch:TRIGger:HBTO:DATA:ICONdition?** <SearchName>

Sets the condition for the index. You can define an exact value or a value range.

**Parameters:**
<DataIdxOperator> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:HBTO:DATA:IMIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:HBTO:DATA:IMIN` and `SEARch:TRIGger:HBTO:DATA:IMAX`.

*RST: INRange

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Index" on page 661

**SEARch:TRIGger:HBTO:DATA:IMIN** <SearchName>,<DataIndexMin>
**SEARch:TRIGger:HBTO:DATA:IMIN?** <SearchName>

Specifies the index bit pattern, or sets the start value of a pattern range.

**Parameters:**
<DataIndexMin> Range: 1 to 0
Increment: 1
*RST: 1

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Index" on page 661

**SEARch:TRIGger:HBTO:DATA:IMAX** <SearchName>,<DataIndexMax>
**SEARch:TRIGger:HBTO:DATA:IMAX?** <SearchName>

Sets the end value of the index range if `SEARch:TRIGger:HBTO:DATA:ICONdition` is set to `INRange` or `OORange`.

**Parameters:**
<DataIndexMax> Range: 1 to 0
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Index" on page 661

**SEARch:TRIGger:HBTO:ERRor:PREamble** <SearchName>,<ErrorPreamble>
**SEARch:TRIGger:HBTO:ERRor:PREamble?** <SearchName>

Enables / disables search for preamble error.

**Parameters:**
<ErrorPreamble> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Preamble Error" on page 661

**SEARch:TRIGger:HBTO:ERRor:CRC** <SearchName>,<ErrorCRC>
**SEARch:TRIGger:HBTO:ERRor:CRC?** <SearchName>

Enables / disables trigger on Cyclic Redundancy Check (CRC) error.

**Parameters:**
<ErrorCRC> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "CRC Error" on page 661

---

**SEARch:TRIGger:HBTO:ERRor:SFD** <SearchName>,<ErrorSFD>
**SEARch:TRIGger:HBTO:ERRor:SFD?** <SearchName>

Enables / disables search for start frame delimiter (SFD) error.

**Parameters:**
<ErrorSFD> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "SFD Error" on page 661

### 24.17.11.5 Search results

In all `SEARch:RESult:HBTO:FRAMe<m>` commands, the suffix <m> selects the frame number in the list of search results.

SEARch:RESult:HBTO:FCOunt? ..... 1878
SEARch:RESult:HBTO:FRAMe<m>:TYPE? ..... 1879
SEARch:RESult:HBTO:FRAMe<m>:STATe? ..... 1879
SEARch:RESult:HBTO:FRAMe<m>:STARt? ..... 1879
SEARch:RESult:HBTO:FRAMe<m>:STOP? ..... 1880
SEARch:RESult:HBTO:FRAMe<m>:DESTaddress? ..... 1880
SEARch:RESult:HBTO:FRAMe<m>:SRCaddress? ..... 1880
SEARch:RESult:HBTO:FRAMe<m>:DATA? ..... 1881
SEARch:RESult:HBTO:FRAMe<m>:CRC? ..... 1881
SEARch:RESult:HBTO:FRAMe<m>:NUMWords? ..... 1881
SEARch:RESult:HBTO:FRAMe<m>:FTYPe? ..... 1882
SEARch:RESult:HBTO:FRAMe<m>:DSYMbol? ..... 1882
SEARch:RESult:HBTO:FRAMe<m>:SSYMbol? ..... 1882
SEARch:RESult:HBTO:FRAMe<m>:WORD<n>:TYPE? ..... 1883
SEARch:RESult:HBTO:FRAMe<m>:WORD<n>:VALue? ..... 1883

---

**SEARch:RESult:HBTO:FCOunt?** <SearchName>

Returns the number of decoded frames within the search result.

**Query parameters:**
<SearchName>

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:HBTO:FRAMe<m>:TYPE?** <SearchName>

Returns the value of length / type field of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<TypeLen> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:HBTO:FRAMe<m>:STATe?** <SearchName>

Returns the state of the frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | ERR_PREAMBLE | ERR_SFD | ERR_LENGTH | ERR_CRC | UNCorrelated | INComplete

**OK**
Valid frame.

**ERR_PREAMBLE**
Erroneous frame due to preamble error.

**ERR_SFD**
Erroneous frame due to SFD error.

**ERR_LENGTH**
Erroneous frame due to length / type error.

**ERR_CRC**
Erroneous frame due to CRC error.

*RST: OK

**Usage:** Query only

**SEARch:RESult:HBTO:FRAMe<m>:STARt?** <SearchName>

Returns the start time of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:HBTO:FRAMe<m>:STOP?** <SearchName>

Returns the end time of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:HBTO:FRAMe<m>:DESTaddress?** <SearchName>

Returns the destination address of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<DestAddress>

**Usage:** Query only

---

**SEARch:RESult:HBTO:FRAMe<m>:SRCaddress?** <SearchName>

Returns the source address of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<SourceAddress>

**Usage:** Query only

---

**SEARch:RESult:HBTO:FRAMe<m>:DATA?** <SearchName>

Returns the data for the selected frame, corresponds to the Data column in the decode results table.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

---

**SEARch:RESult:HBTO:FRAMe<m>:CRC?** <SearchName>

Returns the CRC checksum of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<CRC> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:HBTO:FRAMe<m>:NUMWords?** <SearchName>

Returns the number of decoded words for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<NumWords> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:HBTO:FRAMe<m>:FTYPe?** <SearchName>

Returns the type of frame for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameType> IDLE | MAC | FILLer | UNKNown

**IDLE**
IDLE frame. This frame is used for clock synchronization.

**MAC**
MAC frame. This frame contains information that define how to go about transmitting and receiving frames.

**FILLer**
Filler frame. The frame is used to maintain transmission activity.

**UNKNown**
No meaningful frame can be determined.

*RST: MAC

**Usage:** Query only

**SEARch:RESult:HBTO:FRAMe<m>:DSYMbol?** <SearchName>

Returns the destination symbols of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<DestTranslation>

**Usage:** Query only

**SEARch:RESult:HBTO:FRAMe<m>:SSYMbol?** <SearchName>

Returns the source symbols of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<SrcTranslation>

**Usage:** Query only

**SEARch:RESult:HBTO:FRAMe<m>:WORD<n>:TYPE?** <SearchName>

Returns the data type of the selected word within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<WordType> String parameter

**Usage:** Query only

**SEARch:RESult:HBTO:FRAMe<m>:WORD<n>:VALue?** <SearchName>

Returns the data value of the selected word within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<WordValue> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

## 24.17.12 Ethernet 1000BASE-T1 (option R&S RTP-K58)


- Configuration....1883
- Decode results....1887
- Trigger....1893
- Search settings....1902
- Search results....1910


### 24.17.12.1 Configuration

In all `BUS<m>:TBTO` commands, the suffix <m> selects the serial bus.

**BUS<m>:TBTO:ATTN** <Attenuation>

Sets the attenuation factor. It is used to de-amplify the reverse signal before subtracting it from the forward signal.

**Suffix:**
<m> 1..4

**Parameters:**
<Attenuation> Range: -100 to 0
Increment: 0.1
*RST: -26
Default unit: dB

**Manual operation:** See "Reverse attenuation" on page 665

---

**BUS<m>:TBTO:FDAM** <SourceDAminus>

Selects the DA- source of the provided forward single ended signal, if `BUS<m>:TBTO:FTYP` is set to `SINGle`.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceDAminus> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

**Manual operation:** See "Forward" on page 664

---

**BUS<m>:TBTO:FDAP** <SourceDAplus>

Selects the DA+ source of the provided forward single ended signal, if `BUS<m>:TBTO:FTYP` is set to `SINGle`.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceDAplus> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

**Manual operation:** See "Forward" on page 664

---

**BUS<m>:TBTO:FDIF** <SrcDiff>

Sets the source of the provided forward differential signal, if `BUS<m>:TBTO:FTYP` is set to `DIFFerential`.

**Suffix:**
<m> 1..4

**Parameters:**
<SrcDiff> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

**Manual operation:** See "Forward" on page 664

**BUS<m>:TBTO:FTYP** <SignalType>

Sets the type of forward signal measurement.

**Suffix:**
<m> 1..4

**Parameters:**
<SignalType> DIFFerential | SINGle
*RST: DIFFerential

**Manual operation:** See "Forward" on page 664

**BUS<m>:TBTO:RDAM** <SrcRevDAminus>

Selects the DA- source of the provided reversed single ended signal, if `BUS<m>:TBTO:RTYP` is set to `SINGle`.

**Suffix:**
<m> 1..4

**Parameters:**
<SrcRevDAminus> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

**Manual operation:** See "Reverse" on page 664

**BUS<m>:TBTO:RDAP** <SourceRevDAplus>

Selects the DA+ source of the provided reversed single ended signal, if `BUS<m>:TBTO:RTYP` is set to `SINGle`.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceRevDAplus> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

**Manual operation:** See "Reverse" on page 664

**BUS<m>:TBTO:RDIF** <SrcRevDiff>

Sets the source of the provided reversed differential signal, if `BUS<m>:TBTO:RTYP` is set to `DIFFerential`.

**Suffix:**
<m> 1..4

**Parameters:**
<SrcRevDiff> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

**Manual operation:** See "Reverse" on page 664

**BUS<m>:TBTO:RTYP** <SignalRevType>

Sets the type of reverse signal measurement.

**Suffix:**
<m> 1..4

**Parameters:**
<SignalRevType> DISabled | DIFFerential | SINGle
*RST: DISabled

**Manual operation:** See "Reverse" on page 664

**BUS<m>:TBTO:MODE** <Mode>

Selects the direction of the full-duplex signal you want to look at.

**Suffix:**
<m> 1..4

**Parameters:**
<Mode> MASTer | SLAVe | AUTO
*RST: AUTO

**Manual operation:** See "Operation mode" on page 665

**BUS<m>:TBTO:POLarity** <Polarity>

Selects the polarity of the data signal.

**Suffix:**
<m> 1..4

**Parameters:**
<Polarity> NORMal | INVert
*RST: NORMal

**Manual operation:** See "Polarity" on page 665

**BUS<m>:TBTO:SYMRate** <SymbolRate>

Defines the rate of ternary symbols.

**Suffix:**
<m> 1..4

**Parameters:**
<SymbolRate> Range: 10 to 1500
Increment: 1E-3
*RST: 750
Default unit: MSymb/s

### 24.17.12.2 Decode results

To show the results on the screen, use the following commands:

- BUS<m>:RESult on page 1638

BUS<m>:TBTO:RESult:FCOunt? ..... 1887
BUS<m>:TBTO:RESult:FRAMe<n>:BITRate? ..... 1887
BUS<m>:TBTO:RESult:FRAMe<n>:CRC? ..... 1888
BUS<m>:TBTO:RESult:FRAMe<n>:DATA? ..... 1888
BUS<m>:TBTO:RESult:FRAMe<n>:DESTaddress? ..... 1888
BUS<m>:TBTO:RESult:FRAMe<n>:DSYMbol? ..... 1889
BUS<m>:TBTO:RESult:FRAMe<n>:FTYPe? ..... 1889
BUS<m>:TBTO:RESult:FRAMe<n>:NUMWords? ..... 1889
BUS<m>:TBTO:RESult:FRAMe<n>:SRCaddress? ..... 1890
BUS<m>:TBTO:RESult:FRAMe<n>:SSYMbol? ..... 1890
BUS<m>:TBTO:RESult:FRAMe<n>:STARt? ..... 1890
BUS<m>:TBTO:RESult:FRAMe<n>:STATe? ..... 1891
BUS<m>:TBTO:RESult:FRAMe<n>:STOP? ..... 1891
BUS<m>:TBTO:RESult:FRAMe<n>:TYPE? ..... 1892
BUS<m>:TBTO:RESult:FRAMe<n>:WORD<o>:TYPE? ..... 1892
BUS<m>:TBTO:RESult:FRAMe<n>:WORD<o>:VALue? ..... 1892

---

**BUS<m>:TBTO:RESult:FCOunt?**

Returns the number of decoded frames.

**Suffix:**

<m> 1..4
Selects the serial bus.

**Return values:**

<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:TBTO:RESult:FRAMe<n>:BITRate?**

Returns the primary bit rate.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<PrimaryBitRate> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**BUS<m>:TBTO:RESult:FRAMe<n>:CRC?**

Returns the CRC (frame check) checksum of the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<CRC> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:TBTO:RESult:FRAMe<n>:DATA?**

Returns the data for the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<Data>

**Usage:** Query only

---

**BUS<m>:TBTO:RESult:FRAMe<n>:DESTaddress?**

Returns the destination address of the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<DestAddress>

**Usage:** Query only

**BUS<m>:TBTO:RESult:FRAMe<n>:DSYMbol?**

Returns the destination symbols of the selected frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<DestTranslation>

**Usage:** Query only

**BUS<m>:TBTO:RESult:FRAMe<n>:FTYPe?**

Returns the type of the selected frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<FrameType> IDLE | MAC | OAM | RSFEC | ZEROTSYM | UNKNown

**IDLE**
IDLE frame. This frame is used for clock synchronization.

**MAC**
MAC frame. This frame contains information that define how to go about transmitting and receiving frames.

**OAM**
Operations, administration, and maintenance frame.

**RSFEC**
Reed Solomon forward error correction check frame.

**ZEROTSYM**
Zero terrnary symbol.

**UNKNown**
No meaningful frame can be determined.

*RST: MAC

**Usage:** Query only

**BUS<m>:TBTO:RESult:FRAMe<n>:NUMWords?**

Returns the number of decoded words for the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<NumWords> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:TBTO:RESult:FRAMe<n>:SRCaddress?**

Returns the source address of the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<SrcAddress>

**Usage:** Query only

---

**BUS<m>:TBTO:RESult:FRAMe<n>:SSYMbol?**

Returns the source symbols of the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<SrcTranslation>

**Usage:** Query only

---

**BUS<m>:TBTO:RESult:FRAMe<n>:STARt?**

Returns the start time of the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:TBTO:RESult:FRAMe<n>:STATe?**

Returns the state of the frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<State> OK | ERRFEC | ERRZERO | ERROOR | ERRCRC | INComplete

**OK**
Valid frame.

**ERRFEC**
Reed-Solomon forward error correction.

**ERRZERO**
Zero ternary symbol pair error.

**ERROOR**
Values are out of range error.

**ERR_CRC**
Erroneous frame due to CRC error.

**INComplete**
Incomplete frame.

*RST: OK

**Usage:** Query only

---

**BUS<m>:TBTO:RESult:FRAMe<n>:STOP?**

Returns the end time of the selected frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:TBTO:RESult:FRAMe<n>:TYPE?**

Returns the value of length / type field of the selected frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<TypeLen> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:TBTO:RESult:FRAMe<n>:WORD<o>:TYPE?**

Returns the data type for the selected word.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the word.

**Return values:**

<WordType>

**Usage:** Query only

---

**BUS<m>:TBTO:RESult:FRAMe<n>:WORD<o>:VALue?**

Returns the data value for the selected word.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the word.

**Return values:**
<WordValue> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

### 24.17.12.3 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:TBTO:TRGLevel ..... 1894
TRIGger<m>:TBTO:TYPE ..... 1894
TRIGger<m>:TBTO:CRC:CONDition ..... 1894
TRIGger<m>:TBTO:CRC:MAX ..... 1895
TRIGger<m>:TBTO:CRC:MIN ..... 1895
TRIGger<m>:TBTO:DADDress:CONDition ..... 1895
TRIGger<m>:TBTO:DADDress:MAX ..... 1896
TRIGger<m>:TBTO:DADDress:MIN ..... 1896
TRIGger<m>:TBTO:DATA:DCONdition ..... 1896
TRIGger<m>:TBTO:DATA:DMAX ..... 1897
TRIGger<m>:TBTO:DATA:DMIN ..... 1897
TRIGger<m>:TBTO:DATA:ICONdition ..... 1897
TRIGger<m>:TBTO:DATA:IMAX ..... 1898
TRIGger<m>:TBTO:DATA:IMIN ..... 1898
TRIGger<m>:TBTO:ERRor:CRC ..... 1898
TRIGger<m>:TBTO:ERRor:FEC ..... 1899
TRIGger<m>:TBTO:ERRor:OOR ..... 1899
TRIGger<m>:TBTO:ERRor:ZERO ..... 1899
TRIGger<m>:TBTO:LENGth:CONDition ..... 1899
TRIGger<m>:TBTO:LENGth:MAX ..... 1900
TRIGger<m>:TBTO:LENGth:MIN ..... 1900
TRIGger<m>:TBTO:SADDress:CONDition ..... 1901
TRIGger<m>:TBTO:SADDress:MAX ..... 1901
TRIGger<m>:TBTO:SADDress:MIN ..... 1901

**TRIGger<m>:TBTO:TRGLevel** <TriggerLevel>

Sets the trigger level.

**Suffix:**
<m> 1..3

**Parameters:**
<TriggerLevel> Range: -0.5 to 0.5
Increment: 1E-3
*RST: 0.01
Default unit: V

**Manual operation:** See "Trigger level" on page 669

**TRIGger<m>:TBTO:TYPE** <Type>

Selects the type of frame to be triggered on.

**Suffix:**
<m> 1..3

**Parameters:**
<Type> STARt | MAC | IDLE | ERRor

**STARt**
Start of frame.

**MAC**
MAC frame. This frame contains information that define how to go about transmitting and receiving frames.

**IDLE**
IDLE frame. This frame is used for clock synchronization.

**ERRor**
Error frame. You can enable the error to be triggered on:
`TRIGger<m>:TBTO:ERRor:CRC`
`TRIGger<m>:TBTO:ERRor:FEC`
`TRIGger<m>:TBTO:ERRor:OOR`
`TRIGger<m>:TBTO:ERRor:ZERO`

*RST: STARt

**Manual operation:** See "Type" on page 668

**TRIGger<m>:TBTO:CRC:CONDition** <CRCOperator>

Sets the condition for the frame check. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<CRCOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:TBTO:CRC:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:TBTO:CRC:MIN` and `TRIGger<m>:TBTO:CRC:MAX`.

*RST: EQUal

**Manual operation:** See "Frame check" on page 671

---

**TRIGger<m>:TBTO:CRC:MAX** <CRCPatternMax>

Sets the end value of the frame check range if `TRIGger<m>:TBTO:CRC:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<CRCPatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Frame check" on page 671

---

**TRIGger<m>:TBTO:CRC:MIN** <CRCPatternMin>

Specifies the frame check bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<CRCPatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Frame check" on page 671

---

**TRIGger<m>:TBTO:DADDress:CONDition** <DestAddrOptor>

Sets the condition for the destination address. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<DestAddrOptor> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:TBTO:DADDress:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:TBTO:DADDress:MIN` and `TRIGger<m>:TBTO:DADDress:MAX`.

*RST: EQUal

**Manual operation:** See "Destination address" on page 669

---

**TRIGger<m>:TBTO:DADDress:MAX** <DestAddrPattMax>

Sets the end value of the destination address range if `TRIGger<m>:TBTO:DADDress:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<DestAddrPattMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Destination address" on page 669

---

**TRIGger<m>:TBTO:DADDress:MIN** <DestAddrPattMin>

Specifies the destination address bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<DestAddrPattMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Destination address" on page 669

---

**TRIGger<m>:TBTO:DATA:DCONdition** <DataOperator>

Sets the condition for the data. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:TBTO:DATA:DMIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:TBTO:DATA:DMIN` and `TRIGger<m>:TBTO:DATA:DMAX`.

*RST: EQUal

**Manual operation:** See "Data" on page 671

---

**TRIGger<m>:TBTO:DATA:DMAX** <DataMax>

Sets the end value of the data range if `TRIGger<m>:TBTO:DATA:DCONdition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<DataMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Data" on page 671

---

**TRIGger<m>:TBTO:DATA:DMIN** <DataMin>

Specifies the data bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Data" on page 671

---

**TRIGger<m>:TBTO:DATA:ICONdition** <DataIdxOperator>

Sets the condition for the index. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataIdxOperator> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:TBTO:DATA:IMIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:TBTO:DATA:IMIN` and `TRIGger<m>:TBTO:DATA:IMAX`.

*RST: INRange

**Manual operation:** See "Index" on page 671

---

**TRIGger<m>:TBTO:DATA:IMAX** <DataIndexMax>

Sets the end value of the index range if `TRIGger<m>:TBTO:DATA:ICONdition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<DataIndexMax> Range: 1 to 0
Increment: 1
*RST: 0

**Manual operation:** See "Index" on page 671

---

**TRIGger<m>:TBTO:DATA:IMIN** <DataIndexMin>

Specifies the index bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataIndexMin> Range: 1 to 0
Increment: 1
*RST: 1

**Manual operation:** See "Index" on page 671

---

**TRIGger<m>:TBTO:ERRor:CRC** <ErrorCRC>

Enables triggering on cyclic redundancy check (CRC) error.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorCRC> ON | OFF
*RST: ON

**Manual operation:** See "CRC error" on page 672

---

**TRIGger<m>:TBTO:ERRor:FEC** <ErrorFEC>

Enables triggering, if more than 22 RS-FEC errors are detected.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorFEC> ON | OFF
*RST: ON

**Manual operation:** See "RS-FEC error" on page 672

---

**TRIGger<m>:TBTO:ERRor:OOR** <ErrorOOR>

Enables triggering if values are out of range (OOR). For example, a preamble error, or invalid symbols in control blocks.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorOOR> ON | OFF
*RST: ON

**Manual operation:** See "Out of range error" on page 672

---

**TRIGger<m>:TBTO:ERRor:ZERO** <ErrorZERO>

Enables triggering, if a zero ternary symbol pair (0-0) is detected.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorZERO> ON | OFF
*RST: ON

**Manual operation:** See "Zero terrnary symbol" on page 672

---

**TRIGger<m>:TBTO:LENGth:CONDition** <TypeOperator>

Sets the condition for the length / type. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<TypeOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:TBTO:LENGth:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:TBTO:LENGth:MIN` and `TRIGger<m>:TBTO:LENGth:MAX`.

*RST: EQUal

**Manual operation:** See "Length/Type" on page 670

---

**TRIGger<m>:TBTO:LENGth:MAX** <TypePatternMax>

Sets the end value of the length / type range if `TRIGger<m>:TBTO:LENGth:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<TypePatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Length/Type" on page 670

---

**TRIGger<m>:TBTO:LENGth:MIN** <TypePatternMin>

Specifies the length / type bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<TypePatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Length/Type" on page 670

**TRIGger<m>:TBTO:SADDress:CONDition** <SrcAddrOperator>

Sets the condition for the source address. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<SrcAddrOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:TBTO:SADDress:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:TBTO:SADDress:MIN` and `TRIGger<m>:TBTO:SADDress:MAX`.

*RST: EQUal

**Manual operation:** See "Source address" on page 670

**TRIGger<m>:TBTO:SADDress:MAX** <SrcAddrPattMax>

Sets the end value of the source address range, if `TRIGger<m>:TBTO:SADDress:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<SrcAddrPattMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Source address" on page 670

**TRIGger<m>:TBTO:SADDress:MIN** <SrcAddrPattMin>

Specifies the source address bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<SrcAddrPattMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Source address" on page 670

### 24.17.12.4 Search settings

SEARch:TRIGger:TBTO:CRC:CONDition ..... 1902
SEARch:TRIGger:TBTO:CRC:MAX ..... 1903
SEARch:TRIGger:TBTO:CRC:MIN ..... 1903
SEARch:TRIGger:TBTO:DADDress:CONDition ..... 1903
SEARch:TRIGger:TBTO:DADDress:MAX ..... 1904
SEARch:TRIGger:TBTO:DADDress:MIN ..... 1904
SEARch:TRIGger:TBTO:DATA:DCONdition ..... 1904
SEARch:TRIGger:TBTO:DATA:DMAX ..... 1905
SEARch:TRIGger:TBTO:DATA:DMIN ..... 1905
SEARch:TRIGger:TBTO:DATA:ICONdition ..... 1905
SEARch:TRIGger:TBTO:DATA:IMAX ..... 1906
SEARch:TRIGger:TBTO:DATA:IMIN ..... 1906
SEARch:TRIGger:TBTO:ERRor:CRC ..... 1906
SEARch:TRIGger:TBTO:ERRor:FEC ..... 1907
SEARch:TRIGger:TBTO:ERRor:OOR ..... 1907
SEARch:TRIGger:TBTO:ERRor:ZERO ..... 1907
SEARch:TRIGger:TBTO:LENGth:CONDition ..... 1908
SEARch:TRIGger:TBTO:LENGth:MAX ..... 1908
SEARch:TRIGger:TBTO:LENGth:MIN ..... 1908
SEARch:TRIGger:TBTO:SADDress:CONDition ..... 1909
SEARch:TRIGger:TBTO:SADDress:MAX ..... 1909
SEARch:TRIGger:TBTO:SADDress:MIN ..... 1909
SEARch:TRIGger:TBTO:TYPE ..... 1910

---

**SEARch:TRIGger:TBTO:CRC:CONDition** <SearchName>,<CRCOperator>
**SEARch:TRIGger:TBTO:CRC:CONDition?** <SearchName>

Sets the condition for the frame check. You can define an exact value or a value range.

**Parameters:**

<CRCOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:TBTO:CRC:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:TBTO:CRC:MIN` and `SEARch:TRIGger:TBTO:CRC:MAX`.

*RST: EQUal

**Parameters for setting and query:**

<SearchName>

**Manual operation:** See "Frame check" on page 677

**SEARch:TRIGger:TBTO:CRC:MAX** <SearchName>,<CRCPatternMax>
**SEARch:TRIGger:TBTO:CRC:MAX?** <SearchName>

Sets the end value of the frame check range if `SEARch:TRIGger:TBTO:CRC:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<CRCPatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame check" on page 677

**SEARch:TRIGger:TBTO:CRC:MIN** <SearchName>,<CRCPatternMin>
**SEARch:TRIGger:TBTO:CRC:MIN?** <SearchName>

Specifies the frame check bit pattern, or sets the start value of a pattern range.

**Parameters:**
<CRCPatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame check" on page 677

**SEARch:TRIGger:TBTO:DADDress:CONDition** <SearchName>,<DestAddrOptor>
**SEARch:TRIGger:TBTO:DADDress:CONDition?** <SearchName>

Sets the condition for the destination address. You can define an exact value or a value range.

**Parameters:**
<DestAddrOptor> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:TBTO:DADDress:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:TBTO:DADDress:MIN` and `SEARch:TRIGger:TBTO:DADDress:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Destination address" on page 676

---

**SEARch:TRIGger:TBTO:DADDress:MAX** <SearchName>,<DestAddrPattMax>
**SEARch:TRIGger:TBTO:DADDress:MAX?** <SearchName>

Sets the end value of the destination address range if `SEARch:TRIGger:TBTO:DADDress:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<DestAddrPattMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Destination address" on page 676

---

**SEARch:TRIGger:TBTO:DADDress:MIN** <SearchName>,<DestAddrPattMin>
**SEARch:TRIGger:TBTO:DADDress:MIN?** <SearchName>

Specifies the destination address bit pattern, or sets the start value of a pattern range.

**Parameters:**
<DestAddrPattMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Destination address" on page 676

---

**SEARch:TRIGger:TBTO:DATA:DCONdition** <SearchName>,<DataOperator>
**SEARch:TRIGger:TBTO:DATA:DCONdition?** <SearchName>

Sets the condition for the data. You can define an exact value or a value range.

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:TBTO:DATA:DMIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:TBTO:DATA:DMIN` and `SEARch:TRIGger:TBTO:DATA:DMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data" on page 678

---

**SEARch:TRIGger:TBTO:DATA:DMAX** <SearchName>,<DataMax>
**SEARch:TRIGger:TBTO:DATA:DMAX?** <SearchName>

Sets the end value of the data range if `SEARch:TRIGger:TBTO:DATA:DCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<DataMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data" on page 678

---

**SEARch:TRIGger:TBTO:DATA:DMIN** <SearchName>,<DataMin>
**SEARch:TRIGger:TBTO:DATA:DMIN?** <SearchName>

Specifies the data bit pattern, or sets the start value of a pattern range.

**Parameters:**
<DataMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data" on page 678

---

**SEARch:TRIGger:TBTO:DATA:ICONdition** <SearchName>,<DataIdxOperator>
**SEARch:TRIGger:TBTO:DATA:ICONdition?** <SearchName>

Sets the search condition for the index. You can define an exact value or a value range.

**Parameters:**
<DataIdxOperator> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:TBTO:DATA:IMIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:TBTO:DATA:IMIN` and `SEARch:TRIGger:TBTO:DATA:IMAX`.

*RST: INRange

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Index" on page 678

---

**SEARch:TRIGger:TBTO:DATA:IMAX** <SearchName>,<DataIndexMax>
**SEARch:TRIGger:TBTO:DATA:IMAX?** <SearchName>

Sets the end value of the index range to be searched for, if `SEARch:TRIGger:TBTO:DATA:IMAX` is set to `INRange` or `OORange`.

**Parameters:**
<DataIndexMax>
Range: 1 to 0
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Index" on page 678

---

**SEARch:TRIGger:TBTO:DATA:IMIN** <SearchName>,<DataIndexMin>
**SEARch:TRIGger:TBTO:DATA:IMIN?** <SearchName>

Specifies the index bit pattern, or sets the start value of a pattern range.

**Parameters:**
<DataIndexMin>
Range: 1 to 0
Increment: 1
*RST: 1

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Index" on page 678

---

**SEARch:TRIGger:TBTO:ERRor:CRC** <SearchName>,<ErrorCRC>
**SEARch:TRIGger:TBTO:ERRor:CRC?** <SearchName>

Enables search for cyclic redundancy check (CRC) error.

**Parameters:**
<ErrorCRC> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "CRC error" on page 679

---

**SEARch:TRIGger:TBTO:ERRor:FEC** <SearchName>,<ErrorFEC>
**SEARch:TRIGger:TBTO:ERRor:FEC?** <SearchName>

Enables search for RS-FEC errors.

**Parameters:**
<ErrorFEC> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "RS-FEC error" on page 679

---

**SEARch:TRIGger:TBTO:ERRor:OOR** <SearchName>,<ErrorOOR>
**SEARch:TRIGger:TBTO:ERRor:OOR?** <SearchName>

Enables search for values that are out of range.

**Parameters:**
<ErrorOOR> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Out of range error" on page 679

---

**SEARch:TRIGger:TBTO:ERRor:ZERO** <SearchName>,<ErrorZERO>
**SEARch:TRIGger:TBTO:ERRor:ZERO?** <SearchName>

Enables search for zero ternary symbol pairs error.

**Parameters:**
<ErrorZERO> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Zero terrnary symbol" on page 679

**SEARch:TRIGger:TBTO:LENGth:CONDition** <SearchName>,<TypeOperator>
**SEARch:TRIGger:TBTO:LENGth:CONDition?** <SearchName>

Sets the condition for the length/ type. You can define an exact value or a value range.

**Parameters:**
<TypeOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, not equal, less than, less or equal than, greater than, greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:TBTO:LENGth:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:TBTO:LENGth:MIN` and `SEARch:TRIGger:TBTO:LENGth:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Length/Type" on page 677

---

**SEARch:TRIGger:TBTO:LENGth:MAX** <SearchName>,<TypePatternMax>
**SEARch:TRIGger:TBTO:LENGth:MAX?** <SearchName>

Sets the end value of the length / type range if `SEARch:TRIGger:TBTO:LENGth:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<TypePatternMax>

**Parameters for setting and query:**
<SearchName> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Length/Type" on page 677

---

**SEARch:TRIGger:TBTO:LENGth:MIN** <SearchName>,<TypePatternMin>
**SEARch:TRIGger:TBTO:LENGth:MIN?** <SearchName>

Specifies the length/ type bit pattern, or sets the start value of a pattern range.

**Parameters:**
<TypePatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Length/Type" on page 677

---

**SEARch:TRIGger:TBTO:SADDress:CONDition** <SearchName>,<SrcAddrOperator>
**SEARch:TRIGger:TBTO:SADDress:CONDition?** <SearchName>

Sets the condition for the source address. You can define an exact value or a value range.

**Parameters:**

<SrcAddrOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, not equal, less than, less or equal than, greater than, greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:TBTO:SADDress:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:TBTO:SADDress:MIN` and`SEARch:TRIGger:TBTO:SADDress:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Source address" on page 677

---

**SEARch:TRIGger:TBTO:SADDress:MAX** <SearchName>,<SrcAddrPattMax>
**SEARch:TRIGger:TBTO:SADDress:MAX?** <SearchName>

Sets the end value of the source address range if `SEARch:TRIGger:TBTO:SADDress:CONDition` is set to `INRange` or `OORange`.

**Parameters:**

<SrcAddrPattMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Source address" on page 677

---

**SEARch:TRIGger:TBTO:SADDress:MIN** <SearchName>,<SrcAddrPattMin>
**SEARch:TRIGger:TBTO:SADDress:MIN?** <SearchName>

Specifies the source address bit pattern, or sets the start value of a pattern range.

**Parameters:**
<SrcAddrPattMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Source address" on page 677

---

**SEARch:TRIGger:TBTO:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:TBTO:TYPE?** <SearchName>

Selects the type of frame to be searched for.

**Parameters:**
<Type> STARt | MAC | IDLE | ERRor

**STARt**
Start of frame.

**MAC**
MAC frame. This frame contains information that define how to go about transmitting and receiving frames.

**IDLE**
IDLE frame. This frame is used for clock synchronization.

**ERRor**
Error frame. This frame contains erroneous bits.

*RST: STARt

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Type" on page 675

### 24.17.12.5 Search results

SEARch:RESult:TBTO:FCOunt? 1911
SEARch:RESult:TBTO:FRAMe<m>:CRC? 1911
SEARch:RESult:TBTO:FRAMe<m>:DATA? 1911
SEARch:RESult:TBTO:FRAMe<m>:DESTaddress? 1911
SEARch:RESult:TBTO:FRAMe<m>:DSYMbol? 1912
SEARch:RESult:TBTO:FRAMe<m>:FTYPe? 1912
SEARch:RESult:TBTO:FRAMe<m>:NUMWords? 1913
SEARch:RESult:TBTO:FRAMe<m>:SRCaddress? 1913
SEARch:RESult:TBTO:FRAMe<m>:SSYMbol? 1913
SEARch:RESult:TBTO:FRAMe<m>:STARt? 1914
SEARch:RESult:TBTO:FRAMe<m>:STATe? 1914
SEARch:RESult:TBTO:FRAMe<m>:STOP? 1914
SEARch:RESult:TBTO:FRAMe<m>:TYPE? 1915

SEARch:RESult:TBTO:FRAMe<m>:WORD<n>:TYPE? ..... 1915
SEARch:RESult:TBTO:FRAMe<m>:WORD<n>:VALue? ..... 1915
SEARch:RESult:TBTO:FRAMe<m>:WORD<n>:VSTR? ..... 1916

**SEARch:RESult:TBTO:FCOunt?** <SearchName>

Returns the number of decoded frames within the search result.

**Query parameters:**
<SearchName>

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:TBTO:FRAMe<m>:CRC?** <SearchName>

Returns the CRC checksum of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<CRC> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:TBTO:FRAMe<m>:DATA?** <SearchName>

Returns the data for the selected frame.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

**SEARch:RESult:TBTO:FRAMe<m>:DESTaddress?** <SearchName>

Returns the destination address of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<DestAddrstr>

**Usage:** Query only

---

**SEARch:RESult:TBTO:FRAMe<m>:DSYMbol?** <SearchName>

Returns the destination symbols of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<DestTranslation>

**Usage:** Query only

---

**SEARch:RESult:TBTO:FRAMe<m>:FTYPe?** <SearchName>

Returns the type of frame for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameType> IDLE | MAC | OAM | RSFEC | BH | CTLADDR | CTLCODE | ZEROTSYM | UNKNown

**IDLE**
IDLE frame. This frame is used for clock synchronization.

**MAC**
MAC frame. This frame contains information that define how to go about transmitting and receiving frames.

**OAM**
Operations, administration and maintenance (OAM) frame.

**RSFEC**
Reed Solomon forward error correction check frame.

**BH**
BASE-H fame.

**CTLADDR**
Control address frame.

**CTLCODE**
Control code frame.

**ZEROTSYM**
Zero terrnary symbol frame.

**UNKNown**
No meaningful frame can be determined.

*RST: MAC

**Usage:** Query only

---

**SEARch:RESult:TBTO:FRAMe<m>:NUMWords?** <SearchName>

Returns the number of decoded words for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<NumWords> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:TBTO:FRAMe<m>:SRCaddress?** <SearchName>

Returns the source address of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<SrcAddrstr>

**Usage:** Query only

---

**SEARch:RESult:TBTO:FRAMe<m>:SSYMbol?** <SearchName>

Returns the source symbols of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<SrcTranslation>

**Usage:** Query only

---

**SEARch:RESult:TBTO:FRAMe<m>:STARt?** <SearchName>

Returns the start time of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:TBTO:FRAMe<m>:STATe?** <SearchName>

Returns the state of the frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | ERRFEC | ERRZERO | ERROOR | ERRCRC | INComplete

**OK**
Valid frame.

**ERRFEC**
Reed-Solomon forward error correction.

**ERRZERO**
Zero ternary symbol pair error.

**ERROOR**
Values are out of range error.

**ERR_CRC**
Erroneous frame due to CRC error.

**INComplete**
Incomplete frame.

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:TBTO:FRAMe<m>:STOP?** <SearchName>

Returns the end time of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:TBTO:FRAMe<m>:TYPE?** <SearchName>

Returns the value of length / type field of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<TypeLen> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:TBTO:FRAMe<m>:WORD<n>:TYPE?** <SearchName>

Returns the data type of the selected word within the search result.

**Suffix:**
<m> *
Selects the frame.

<n> *
Selects the word.

**Query parameters:**
<SearchName>

**Return values:**
<WordType>

**Usage:** Query only

---

**SEARch:RESult:TBTO:FRAMe<m>:WORD<n>:VALue?** <SearchName>

Returns the data value of the selected word within the search result.

**Suffix:**
<m> *
Selects the frame.

<n> *
Selects the word.

**Query parameters:**
<SearchName>

**Return values:**
<WordValue> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:TBTO:FRAMe<m>:WORD<n>:VSTR?** <SearchName>

Returns the string equivalent of data value for the selected cell.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<WordValueString>

**Usage:** Query only

## 24.17.13 RFFE (option R&S RTP-K40)

For programming examples, see Chapter 24.3.7.1, "RFFE (option R&S RTP-K40)", on page 1187.

### 24.17.13.1 Configuration

In all BUS<m>:RFFE commands, the suffix <m> selects the serial bus.

For programming examples, see "Configuring RFFE bus" on page 1187.

BUS<m>:RFFE:CLOCk:SOURce....1917
BUS<m>:RFFE:DATA:SOURce....1917
BUS<m>:RFFE:PRESet....1917
BUS<m>:RFFE:COUPling....1918
BUS<m>:RFFE:DATA:THReshold:HIGH....1918
BUS<m>:RFFE:DATA:THReshold:LOW....1918
BUS<m>:RFFE:DATA:THReshold:HYSTeresis....1918
BUS<m>:RFFE:CLOCk:THReshold:HIGH....1919

BUS<m>:RFFE:CLOCk:THReshold:LOW ..... 1919
BUS<m>:RFFE:CLOCk:THReshold:HYSTeresis ..... 1919
BUS<m>:RFFE:GFILter ..... 1920
BUS<m>:RFFE:GFWidth ..... 1920
BUS<m>:RFFE:MINGap:SELect ..... 1920
BUS<m>:RFFE:MINGap:TIME ..... 1921
BUS<m>:RFFE:RDMD ..... 1921
BUS<m>:RFFE:VERSion ..... 1921

**BUS<m>:RFFE:CLOCk:SOURce** <SourceClock>

Sets the source of the RFFE clock line.

For triggering on a serial bus, analog or digital input channels are required.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceClock> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

*RST: C2W1

**Manual operation:** See "SCLK Source" on page 684

**BUS<m>:RFFE:DATA:SOURce** <SourceData>

Sets the source of the RFFE data line. For triggering on a serial bus, analog or digital input channels are required.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceData> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

*RST: C1W1

**Manual operation:** See "SDATA Source" on page 685

**BUS<m>:RFFE:PRESet** <ThresholdPreset>

Sets default threshold voltages for 1.2 V or 1.8 V bus. Set the value to MAN if you want to define individual thresholds.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdPreset> V12 | V18 | MANual

*RST: V12

**BUS<m>:RFFE:COUPling** <ThresCpl>

If enabled, the SDATA thresholds are set to the SCLK threshold values.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresCpl> ON | OFF
*RST: ON

**BUS<m>:RFFE:DATA:THReshold:HIGH** <ThresDatPos>

Set the positive going threshold ($V_{TP}$) for the data signal.

Note that the high and low thresholds and the hysteresis values are dependent, so it is sufficient to define two values for each line.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresDatPos> Range: 0 to 2.5
Increment: 0.1
*RST: 0.72
Default unit: V

**Manual operation:** See "Thresholds" on page 685

**BUS<m>:RFFE:DATA:THReshold:LOW** <ThresDatNeg>

Set the negative going threshold ($V_{TN}$) for the data signal.

Note that the high and low thresholds and the hysteresis values are dependent, so it is sufficient to define two values for each line.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresDatNeg> Range: 0 to 2.5
Increment: 0.1
*RST: 0.48
Default unit: V

**Manual operation:** See "Thresholds" on page 685

**BUS<m>:RFFE:DATA:THReshold:HYSTeresis** <ThresDatHyst>

Set the hysteresises between $V_{TP}$ and $V_{TN}$ for the data signal.

Note that the high and low thresholds and the hysteresis values are dependent, so it is sufficient to define two values for each line.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresDatHyst> Range: 0 to 1
Increment: 0.1
*RST: 0.2
Default unit: V

**Manual operation:** See "Thresholds" on page 685

---

**BUS<m>:RFFE:CLOCk:THReshold:HIGH** <ThresClkPos>

Set the positive going threshold ($V_{TP}$) for the clock signal.

Note that the high and low thresholds and the hysteresis values are dependent, so it is sufficient to define two values for each line.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresClkPos> Range: 0 to 2.5
Increment: 0.1
*RST: 0.72
Default unit: V

**Manual operation:** See "Thresholds" on page 685

---

**BUS<m>:RFFE:CLOCk:THReshold:LOW** <ThresClkNeg>

Set the negative going threshold ($V_{TN}$) for the clock signal.

Note that the high and low thresholds and the hysteresis values are dependent, so it is sufficient to define two values for each line.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresClkNeg> Range: 0 to 2.5
Increment: 0.1
*RST: 0.48
Default unit: V

**Manual operation:** See "Thresholds" on page 685

---

**BUS<m>:RFFE:CLOCk:THReshold:HYSTeresis** <ThresClkHyst>

Set the hysteresis between $V_{TP}$ and $V_{TN}$ for the clock signal.

Note that the high and low thresholds and the hysteresis values are dependent, so it is sufficient to define two values for each line.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresClkHyst> Range: 0 to 1
Increment: 0.1
*RST: 0.2
Default unit: V

**Manual operation:** See "Thresholds" on page 685

---

**BUS<m>:RFFE:GFILter** <GlitchFilter>

Enables the glitch filter on the SCLK and SDATA lines to improve decode accuracy.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<GlitchFilter> ON | OFF
*RST: OFF

**Manual operation:** See "Glitch filter" on page 686

---

**BUS<m>:RFFE:GFWidth** <GlitchFilterWdt>

Sets the maximum glitch width to be ignored.

**Suffix:**
<m> 1..4

**Parameters:**
<GlitchFilterWdt> Range: 10E-12 to 10E-9
Increment: 10E-12
*RST: 1E-9
Default unit: s

**Manual operation:** See "Glitch filter" on page 686

---

**BUS<m>:RFFE:MINGap:SELect** <MinGapSelect>

If the minimum gap is enabled, the instrument detects the specified gap.

Set the minimum gap time using `BUS<m>:RFFE:MINGap:TIME`.

**Suffix:**
<m> 1..4

**Parameters:**
<MinGapSelect> ON | OFF
*RST: OFF

**Manual operation:** See "Timing" on page 686

**BUS<m>:RFFE:MINGap:TIME** <MinGapTime>

Sets the minimum idle time between the Bus Park Cycle (BP) and Sequence Start Condition (SSC).

The setting is only relevant if `BUS<m>:RFFE:MINGap:SELect` is enabled.

**Suffix:**
<m> 1..4

**Parameters:**
<MinGapTime> Range: 1E-9 to 10E-6
Increment: 1E-9
*RST: 10E-9
Default unit: s

**Manual operation:** See "Timing" on page 686

**BUS<m>:RFFE:RDMD** <ReadMode>

Selects, if the standard or synchronous read (`SREAD`) mode is used.

**Suffix:**
<m> 1..4

**Parameters:**
<ReadMode> STRD | SREAD
*RST: STRD

**Manual operation:** See "Read mode" on page 684

**BUS<m>:RFFE:VERSion** <Version>

Selects the RFFE version.

**Suffix:**
<m> 1..4

**Parameters:**
<Version> V1DX | V2D0 | V2D1 | V3D0
`V1DX`: 1.x
`V2D0`: 2.0
`V2D1`: 2.1
*RST: V1DX

**Manual operation:** See "Version" on page 684

### 24.17.13.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

For programming examples, see "Triggering on RFFE bus" on page 1188.

TRIGger<m>:RFFE:TYPE....1922
TRIGger<m>:RFFE:ADDRess:CONDition....1925
TRIGger<m>:RFFE:ADDRess:MIN....1925
TRIGger<m>:RFFE:ADDRess:MAX....1925
TRIGger<m>:RFFE:BCOunt:CONDition....1925
TRIGger<m>:RFFE:BCOunt:MIN....1926
TRIGger<m>:RFFE:BCOunt:MAX....1926
TRIGger<m>:RFFE:DATA:DCON....1926
TRIGger<m>:RFFE:DATA:DMIN....1927
TRIGger<m>:RFFE:DATA:DMAX....1927
TRIGger<m>:RFFE:DATA:ICONdition....1927
TRIGger<m>:RFFE:DATA:IMIN....1928
TRIGger<m>:RFFE:DATA:IMAX....1928
TRIGger<m>:RFFE:SADD:CONDition....1928
TRIGger<m>:RFFE:SADD:MIN....1928
TRIGger<m>:RFFE:SADD:MAX....1929
TRIGger<m>:RFFE:MASK:CONDition....1929
TRIGger<m>:RFFE:MASK:MAX....1929
TRIGger<m>:RFFE:MASK:MIN....1929
TRIGger<m>:RFFE:INTerrupt....1930
TRIGger<m>:RFFE:ERRor:BP....1930
TRIGger<m>:RFFE:ERRor:LENGth....1930
TRIGger<m>:RFFE:ERRor:NOResponse....1930
TRIGger<m>:RFFE:ERRor:PARity....1930
TRIGger<m>:RFFE:ERRor:SSC....1931
TRIGger<m>:RFFE:ERRor:USEQuence....1931
TRIGger<m>:RFFE:ERRor:GAP....1931
TRIGger<m>:RFFE:ERRor:VERSion....1931
TRIGger<m>:RFFE:INTerrupt....1931

---

**TRIGger<m>:RFFE:TYPE** <Type>

Selects the trigger type for RFFE analysis.

**Parameters:**

<Type> STARt | STOP | RZWR | RWR | RRD | ERWR | ERRD | ERWL | ERRL | IRSUM | MSKW | MASOHO | MASW | MASR | MCTW | MCTR | ERRor

**STARt**
Triggers on the beginning of a command sequence, exactly after the slave address. Optionally, you can specify a slave address condition.

**STOP**
Triggers on the end of a command sequence, exactly at the start of the bus park cycle. Optionally, you can specify a slave address condition.

**RZWR**
Triggers on Register 0 Write command sequences. Optionally, you can specify a slave address condition and a data pattern condition.

**RWR**
Triggers on Register Write command sequences. Optionally, you can specify a slave address condition, a register address condition, and a data pattern condition.

**RRD**
Triggers on Register Read command sequences. Optionally, you can specify a slave address condition, a register address condition, and a data pattern condition.

**ERWR**
Triggers on Extended Register Write command sequences. Optionally, you can specify a slave address condition, a register address condition, a byte count condition, a data index condition, and a data pattern condition.

**ERRD**
Triggers on Extended Register Read command sequences. Optional trigger conditions are the same as for ERWR.

**ERWL**
Triggers on Extended Register Write Long command sequences. Optional trigger conditions are the same as for ERWR.

**ERRL**
Triggers on Extended Register Read Long command sequences. Optional trigger conditions are the same as for ERWR.

**IRSUM**
Triggers on Interrupt Summary and Notification command sequences. Optionally, you can specify a slave address condition, bit count condition and a data pattern condition.

**MSKW**
Masked Write command sequence. Optionally, you can specify a slave address condition, register address condition, mask condition and data pattern condition

**MASOHO**
Triggers onMaster Ownership Handover command sequences. Optionally, you can specify a MID condition.

**MASW**
Triggers on Master Write command sequences. Optionally, you can specify a MID condition, a register address condition, and a data pattern condition.

**MASR**
Triggers on Master Read command sequences. Optionally, you can specify a MID condition, a register address condition, and a data pattern condition.

**MCTW**
Master Context Transfer Write command sequence. Optional: MID condition, register address condition, byte count condition, data index condition, and data pattern condition.

**MCTR**
Master Context Transfer Read command sequence. Optional search conditions are the same as for `MCTW`.

**ERRor**
Triggers if one of the enabled errors occurs in a frame.
To enable the error types, use `TRIGger<m>:RFFE:ERRor:BP`, `TRIGger<m>:RFFE:ERRor:LENGth`, `TRIGger<m>:RFFE:ERRor:NOResponse`, `TRIGger<m>:RFFE:ERRor:PARity`, `TRIGger<m>:RFFE:ERRor:SSC`, and `TRIGger<m>:RFFE:ERRor:USEQuence`.

*RST: STARt

**Manual operation:** See "Type" on page 688

To specify additional trigger cosnditions for command sequences, use the following commands:

| | |
|---|---|
| Slave address /MID | `TRIGger<m>:RFFE:SADD:CONDition`<br>`TRIGger<m>:RFFE:SADD:MIN`<br>`TRIGger<m>:RFFE:SADD:MAX` |
| Register address | `TRIGger<m>:RFFE:ADDRess:CONDition`<br>`TRIGger<m>:RFFE:ADDRess:MIN`<br>`TRIGger<m>:RFFE:ADDRess:MAX` |
| Data pattern | `TRIGger<m>:RFFE:DATA:DCON`<br>`TRIGger<m>:RFFE:DATA:DMIN`<br>`TRIGger<m>:RFFE:DATA:DMAX` |
| Mask address | `TRIGger<m>:RFFE:MASK:CONDition`<br>`TRIGger<m>:RFFE:MASK:MAX`<br>`TRIGger<m>:RFFE:MASK:MIN` |
| Byte count | `TRIGger<m>:RFFE:BCOunt:CONDition`<br>`TRIGger<m>:RFFE:BCOunt:MIN`<br>`TRIGger<m>:RFFE:BCOunt:MAX` |
| Data index | `TRIGger<m>:RFFE:DATA:ICONdition`<br>`TRIGger<m>:RFFE:DATA:IMIN`<br>`TRIGger<m>:RFFE:DATA:IMAX` |

**TRIGger<m>:RFFE:ADDRess:CONDition** <AddressOperator>

Sets the operator to trigger on a specific address pattern or a range.

**Parameters:**

<AddressOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater than, Greater or equal than. These conditions require one pattern to be set with `TRIGger<m>:RFFE:ADDRess:MIN`.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `TRIGger<m>:RFFE:ADDRess:MIN` and `TRIGger<m>:RFFE:ADDRess:MAX`.

*RST: EQUal

**Manual operation:** See "Address" on page 692

---

**TRIGger<m>:RFFE:ADDRess:MIN** <AddressMin>

Specifies a pattern, or sets the start value of a pattern range.

**Parameters:**

<AddressMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Address" on page 692

---

**TRIGger<m>:RFFE:ADDRess:MAX** <AddressMax>

Sets the end value of a range, if the data condition is set to `INRange` or `OORange`. See `TRIGger<m>:RFFE:ADDRess:CONDition`.

**Parameters:**

<AddressMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Address" on page 692

---

**TRIGger<m>:RFFE:BCOunt:CONDition** <ByteCntOperator>

Sets the operator to trigger on a specific byte count or a byte count range.

**Parameters:**

<ByteCntOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater than, Greater or equal than. These conditions require one pattern to be set with `TRIGger<m>:RFFE:BCOunt:MIN` command.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value using the corresponding `TRIGger<m>:RFFE:BCOunt:MIN` and `TRIGger<m>:RFFE:BCOunt:MAX`

*RST: EQUal

**Manual operation:** See "Byte count" on page 693

---

**TRIGger<m>:RFFE:BCOunt:MIN** <ByteCountMin>

Specifies a pattern, or sets the start value of a pattern range.

**Parameters:**
<ByteCountMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Byte count" on page 693

---

**TRIGger<m>:RFFE:BCOunt:MAX** <ByteCountMax>

Sets the end value of a range, if the byte/bit count condition is set to `INRange` or `OORange`. See `TRIGger<m>:RFFE:BCOunt:CONDition`.

**Parameters:**
<ByteCountMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Byte count" on page 693

---

**TRIGger<m>:RFFE:DATA:DCON** <DatPattOptor>

Sets the operator to trigger on a specific pattern or a range.

**Parameters:**
<DatPattOptor> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater than, Greater or equal than. These conditions require one pattern to be set with `TRIGger<m>:RFFE:DATA:DMIN`.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `TRIGger<m>:RFFE:DATA:DMIN` and `TRIGger<m>:RFFE:DATA:DMAX`

*RST: EQUal

**Example:** See "Triggering on RFFE bus" on page 1188.

**Manual operation:** See "Data" on page 693

---

**TRIGger<m>:RFFE:DATA:DMIN** <DataPatternMin>

Specifies a pattern, or sets the start value of a pattern range.

**Parameters:**

<DataPatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Example:** See "Triggering on RFFE bus" on page 1188.

**Manual operation:** See "Data" on page 693

---

**TRIGger<m>:RFFE:DATA:DMAX** <DataPatternMax>

Sets the end value of a range, if the data condition is set to `INRange` or `OORange`. See `TRIGger<m>:RFFE:DATA:DCON`.

**Parameters:**

<DataPatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Example:** See "Triggering on RFFE bus" on page 1188.

**Manual operation:** See "Data" on page 693

---

**TRIGger<m>:RFFE:DATA:ICONdition** <DataIdxOperator>

Sets the operator to define the data frames in which the data pattern is expected.

**Parameters:**

<DataIdxOperator> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal | LTHan | LETHan | GTHan | GETHan**
Equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set using `TRIGger<m>:RFFE:DATA:IMIN`.

**INRange = RANGe**
In range: Set the minimum and maximum value using `TRIGger<m>:RFFE:DATA:IMIN` and `TRIGger<m>:RFFE:DATA:IMAX`.

*RST: INRange

**Manual operation:** See "Index" on page 693

**TRIGger<m>:RFFE:DATA:IMIN** <DataIndexMin>

Specifies a pattern, or sets the start value of the index range.

**Parameters:**

<DataIndexMin> Range: 1 to 32
Increment: 1
*RST: 1

**Manual operation:** See "Index" on page 693

**TRIGger<m>:RFFE:DATA:IMAX** <DataIndexMax>

Sets the end value of a range, if the index condition is set to `INRange` or `OORange`. See `TRIGger<m>:RFFE:DATA:ICONdition`.

**Parameters:**

<DataIndexMax> Range: 1 to 32
Increment: 1
*RST: 32

**Manual operation:** See "Index" on page 693

**TRIGger<m>:RFFE:SADD:CONDition** <SIDOperator>

Sets the operator to trigger on a specific slave address/MID pattern or a range.

**Parameters:**

<SIDOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater than, Greater or equal than. These conditions require one pattern to be set with `TRIGger<m>:RFFE:SADD:MIN`.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `TRIGger<m>:RFFE:SADD:MIN` and `TRIGger<m>:RFFE:SADD:MAX` on page 1929.

*RST: EQUal

**Manual operation:** See "Slave Addr. /MID " on page 692

**TRIGger<m>:RFFE:SADD:MIN** <SIDMin>

Specifies a pattern, or sets the start value of a pattern range.

**Parameters:**

<SIDMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Slave Addr. /MID " on page 692

---

**TRIGger<m>:RFFE:SADD:MAX** <SIDMax>

Sets the end value of a range, if the slave address/MID condition is set to `INRange` or `OORange`. See `SEARch:TRIGger:RFFE:SADD:CONDition`.

**Parameters:**

<SIDMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Slave Addr. /MID " on page 692

---

**TRIGger<m>:RFFE:MASK:CONDition** <MaskOperator>

Sets the operator to trigger on a mask pattern or a range.

**Parameters:**

<MaskOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater than, Greater or equal than. These conditions require one pattern to be set with `TRIGger<m>:RFFE:MASK:MIN`.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value of the range with `TRIGger<m>:RFFE:MASK:MIN` and `TRIGger<m>:RFFE:MASK:MAX`.

*RST: EQUal

**Manual operation:** See "Mask" on page 694

---

**TRIGger<m>:RFFE:MASK:MAX** <MaskMax>

Sets the end value of a range, if themask condition is set to `INRange` or `OORange`. See `TRIGger<m>:RFFE:MASK:CONDition`.

**Parameters:**

<MaskMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (do not care).

**Manual operation:** See "Mask" on page 694

---

**TRIGger<m>:RFFE:MASK:MIN** <MaskMin>

Specifies a mask pattern, or sets the start value of a mask pattern range.

**Parameters:**
<MaskMin>

**Manual operation:** See "Mask" on page 694

---

**TRIGger<m>:RFFE:INTerrupt** <Interrupt>

Sets the interrupt pattern for the interupt summary and notification trigger type.

**Parameters:**
<Interrupt>

---

**TRIGger<m>:RFFE:ERRor:BP** <BusParkError>

Enables the trigger on a bus park error - when an incorrect bus parlk cycle has been found.

**Parameters:**
<BusParkError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 691

---

**TRIGger<m>:RFFE:ERRor:LENGth** <LengthError>

Enables the trigger on a length error - when an incorrect length of the command sequence has been found.

**Parameters:**
<LengthError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 691

---

**TRIGger<m>:RFFE:ERRor:NOResponse** <NoResponse>

Enables the trigger on any No Response Frame.

**Parameters:**
<NoResponse> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 691

---

**TRIGger<m>:RFFE:ERRor:PARity** <ParityError>

Enables the trigger on incorrect parity bits.

**Parameters:**
<ParityError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 691

---

**TRIGger<m>:RFFE:ERRor:SSC** <SSCError>

Enables the trigger on an SSC error - when no valid SSC sequence has been found after the idle time.

**Parameters:**
<SSCError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 691

---

**TRIGger<m>:RFFE:ERRor:USEQuence** <UnknownSequence>

Enables the trigger on an unknown sequence - when the instrument cannot detect any supported command sequence.

**Parameters:**
<UnknownSequence> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 691

---

**TRIGger<m>:RFFE:ERRor:GAP** <GapError>

Enables the trigger on a gap error.

**Parameters:**
<GapError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 691

---

**TRIGger<m>:RFFE:ERRor:VERSion** <VersionError>

Enables the trigger on RFFE version error.

**Parameters:**
<VersionError> ON | OFF
*RST: ON

**Manual operation:** See "Error conditions" on page 691

---

**TRIGger<m>:RFFE:INTerrupt** <Interrupt>

Sets the interrupt pattern for the interupt summary and notification trigger type.

**Parameters:**
<Interrupt>

### 24.17.13.3 Decode results

In all BUS<m>:RFFE commands, the suffix <m> selects the serial bus. Suffix <n> selects index of the command sequence.

To load and acivate a label list, use:

- `BUS<m>:NEWList` on page 1639
- `BUS<m>:SYMBols` on page 1639

BUS<m>:RFFE:FCOunt?....1932
BUS<m>:RFFE:SEQuence<n>:STATe?....1932
BUS<m>:RFFE:SEQuence<n>:STARt?....1933
BUS<m>:RFFE:SEQuence<n>:STOP?....1933
BUS<m>:RFFE:SEQuence<n>:TYPE?....1933
BUS<m>:RFFE:SEQuence<n>:SADD?....1934
BUS<m>:RFFE:SEQuence<n>:ADDRess?....1934
BUS<m>:RFFE:SEQuence<n>:BCOunt?....1935
BUS<m>:RFFE:SEQuence<n>:DATA?....1935
BUS<m>:RFFE:SEQuence<n>:SYMBol?....1935
BUS<m>:RFFE:SEQuence<n>:PADone?....1936
BUS<m>:RFFE:SEQuence<n>:PADZero?....1936
BUS<m>:RFFE:SEQuence<n>:PCTRl?....1936
BUS<m>:RFFE:SEQuence<n>:RBTRate?....1936
BUS<m>:RFFE:SEQuence<n>:WBTRate?....1937
BUS<m>:RFFE:SEQuence<n>:BYTE<o>:NAME?....1937
BUS<m>:RFFE:SEQuence<n>:BYTE<o>:STATe?....1937
BUS<m>:RFFE:SEQuence<n>:BYTE<o>:VALue?....1938

---

**BUS<m>:RFFE:FCOunt?**

Returns the number of command sequences in the current acquisition.

**Suffix:**
<m> 1..4

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:RFFE:SEQuence<n>:STATe?**

Returns the overall state of the selected command sequence.

**Suffix:**
<m> 1..4

<n> *

**Return values:**

<FrameState> OK | SSC | VERSion | PARity | LENGth | BPERR | GAP | NORESPONSE | INComplete

`OK`: the sequence is valid.
`SSC`: SSC error.
`VERSion`: RFFE version.
`PARity`: parity error.
`LENGth`: length error.
`BPERR`: bus park error.
`NORESPONSE`: No response.
`INComplete`: The sequence is not completely contained in the acquisition. The acquired part of the sequence is valid.

*RST: OK

**Usage:** Query only

---

**BUS<m>:RFFE:SEQuence<n>:STARt?**
**BUS<m>:RFFE:SEQuence<n>:STOP?**

Return the start time (SSC) and stop time (BP) of the selected command sequence.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<Start>, <Stop> Time

Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:RFFE:SEQuence<n>:TYPE?**

Returns the type of the selected command sequence.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<FrameType> UNDESC | RZWR | RWR | RRD | ERWR | ERRD | ERWL | ERRL | MASR | MASW | MASOHO | IRSUM | MSKW | MCTR | MCTW | UNDEF | ERRor

`RZWR`: Register 0 Write
`RWR`: Register Write

RRD: Register Read
ERWR: Extended Register Write
ERRD: Extended Register Read
ERWL: Extended Register Write Long
ERRL: Extended Register Read Long
MASR: Master Read
MASW: Master Write
MASOHO: Master Ownership Handover
IRSUM: Interrupt Summary and Notification
MSKW: Masked Write
MCTR: Master Context Transfer Read
MCTW: Master Context Transfer Write
ERRor: The bits defining the command sequence are not valid, no supported command sequence

*RST: RZWR

**Usage:** Query only

---

**BUS<m>:RFFE:SEQuence<n>:SADD?**

Returns the slave address of the selected command sequence.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<FrameSID> Range: 0 to 15
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:RFFE:SEQuence<n>:ADDRess?**

Returns the register address of the selected command sequence.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<FrameAddress> Range: Depends on the sequence type, address can have 0, 5, 8, or 16 bits
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:RFFE:SEQuence<n>:BCOunt?**

Returns the bit/ byte count of the selected command sequence.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<ByteCount> Decimal value

Range: 15
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:RFFE:SEQuence<n>:DATA?**

Returns the data bytes of the specified command sequence.

Use `BUS<m>:RFFE:SEQuence<n>:BYTE<o>:VALue?` to access the sequence bytes.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Data> Bit pattern (B1, B2, B3, B4…) in variable length up to eight bytes.

**Example:**
```
BUS:RFFE:SEQuence3:DATA?
<-- #H08,#H49,#H54,#H33,#HFF
```

**Usage:** Query only

**BUS<m>:RFFE:SEQuence<n>:SYMBol?**

Returns the symbolic label that belongs to the address of the selected command sequence.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Translation> String containing the label name

**Usage:** Query only

**BUS<m>:RFFE:SEQuence<n>:PADone?**

Returns the address-one parity for the selected command sequence.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<ParityAddr1> OK | PARity
`PARity`: PArrity error
*RST: OK

**Usage:** Query only

**BUS<m>:RFFE:SEQuence<n>:PADZero?**

Returns the address-zero parity for the selected command sequence.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<ParityAddr0> OK | PARity
`PARity`: PArrity error
*RST: OK

**Usage:** Query only

**BUS<m>:RFFE:SEQuence<n>:PCTRl?**

Returns the control parity for the selected command sequence.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<ParityCtrl> OK | PARity
`PARity`: PArrity error
*RST: OK

**Usage:** Query only

**BUS<m>:RFFE:SEQuence<n>:RBTRate?**

Returns the read bit rate of the selected command sequence.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<ReadBitRate> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**BUS<m>:RFFE:SEQuence<n>:WBTRate?**

Returns the write bit rate of the selected command sequence.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<WriteBitRate> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

---

**BUS<m>:RFFE:SEQuence<n>:BYTE<o>:NAME?**

Returns the name of the specified data byte.

**Suffix:**
<m> 1..4

<n> *

<o> *
Selects the number of the data byte (1...n).

**Return values:**
<FrameByteName>

**Usage:** Query only

---

**BUS<m>:RFFE:SEQuence<n>:BYTE<o>:STATe?**

Returns the state of the specified data byte.

**Suffix:**
<m> 1..4

<n> *

<o> *
Selects the number of the data byte (1...n).

**Return values:**
<ByteState> OK | PARity
OK: the byte is valid.
PARity: parity error
*RST: OK

**Usage:** Query only

---

**BUS<m>:RFFE:SEQuence<n>:BYTE<o>:VALue?**

Returns the value of the specified byte in the specified command sequence.

**Suffix:**
<m> 1..4

<n> *

<o> *
Selects the number of the data byte (1...n).

**Return values:**
<ByteValue> To set the value format, use `FORMat:BPATtern`.
The values below – range, increment and reset – are decimal values.
Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

### 24.17.13.4 Search settings

For programming example, see "Searching RFFE data" on page 1188.

SEARch:TRIGger:RFFE:TYPE ... 1939
SEARch:TRIGger:RFFE:SADD:CONDition ... 1941
SEARch:TRIGger:RFFE:ADDRess:CONDition ... 1941
SEARch:TRIGger:RFFE:BCOunt:CONDition ... 1941
SEARch:TRIGger:RFFE:DATA:DCON ... 1941
SEARch:TRIGger:RFFE:DATA:ICONdition ... 1942
SEARch:TRIGger:RFFE:SADD:MIN ... 1942
SEARch:TRIGger:RFFE:ADDRess:MIN ... 1942
SEARch:TRIGger:RFFE:BCOunt:MIN ... 1942
SEARch:TRIGger:RFFE:DATA:IMIN ... 1942
SEARch:TRIGger:RFFE:DATA:DMIN ... 1942
SEARch:TRIGger:RFFE:SADD:MAX ... 1943
SEARch:TRIGger:RFFE:ADDRess:MAX ... 1943
SEARch:TRIGger:RFFE:BCOunt:MAX ... 1943
SEARch:TRIGger:RFFE:DATA:IMAX ... 1943

SEARch:TRIGger:RFFE:DATA:DMAX ..... 1943
SEARch:TRIGger:RFFE:INTerrupt ..... 1943
SEARch:TRIGger:RFFE:ERRor:LENGth ..... 1943
SEARch:TRIGger:RFFE:ERRor:PARity ..... 1944
SEARch:TRIGger:RFFE:ERRor:SSC ..... 1944
SEARch:TRIGger:RFFE:ERRor:GAP ..... 1944
SEARch:TRIGger:RFFE:ERRor:VERSion ..... 1944

**SEARch:TRIGger:RFFE:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:RFFE:TYPE?** <SearchName>

Sets the event to be searched for.

See also: "Type" on page 697

**Parameters:**

<Type> STARt | STOP | RZWR | RWR | RRD | ERWR | ERRD | ERWL | ERRL | IRSUM | MSKW | MASOHO | MASW | MASR | MCTW | MCTR | ERRor

**STARt**
Start of a command sequence (SSC). Optional: slave address.

**STOP**
End of a command sequence (start of the bus park cycle). Optional: slave address condition.

**RZWR**
Register 0 Write command sequence. Optional: slave address condition and data pattern condition.

**RWR**
Register Write command sequence. Optional: slave address condition, register address condition, and data pattern condition.

**RRD**
Register Read command sequence. Optional: slave address condition, register address condition, and data pattern condition.

**ERWR**
Extended Register Write command sequences. Optional: slave address condition, register address condition, byte count condition, data index condition, and data pattern condition.

**ERRD**
Extended Register Read command sequence. Optional search conditions are the same as for `ERWR`.

**ERWL**
Extended Register Write Long command sequence. Optional search conditions are the same as for `ERWR`.

**ERRL**
Extended Register Read Long command sequence. Optional search conditions are the same as for `ERWR`.

**MASR**
Master Read command sequence. Optional: slave address condition, register address condition, and data pattern condition.

**MASW**
Master Write command sequence. Optional: slave address condition, register address condition, and data pattern condition.

**MASOHO**
Master Ownership Handover command sequence. Optional: slave address condition, and data pattern condition.

**MSKW**
Masked Write command sequence. Optional: slave address condition, register address condition, mask condition and data pattern condition.

**MCTW**
Master Context Transfer Write command sequence. Optional: MID condition, register address condition, byte count condition, data index condition, and data pattern condition.

**MCTR**
Master Context Transfer Read command sequence. Optional search conditions are the same as for `MCTW`

**IRSUM**
Interrupt Summary and Notification. Optional: slave address condition, byte count condition, and data pattern condition.

**ERRor**
Searches for enabled errors. To enable the error types, use:
`SEARch:TRIGger:RFFE:ERRor:LENGth`
`SEARch:TRIGger:RFFE:ERRor:PARity`
`SEARch:TRIGger:RFFE:ERRor:SSC`

*RST: STARt

**Parameters for setting and query:**
<SearchName> String parameter

**Example:** See "Searching RFFE data" on page 1188

**Manual operation:** See "Type" on page 697

To specify additional search conditions for command sequences, use the following commands:

| | |
|---|---|
| Slave address | `SEARch:TRIGger:RFFE:SADD:CONDition`<br>`SEARch:TRIGger:RFFE:SADD:MIN`<br>`SEARch:TRIGger:RFFE:SADD:MAX` |
| Register address | `SEARch:TRIGger:RFFE:ADDRess:CONDition`<br>`SEARch:TRIGger:RFFE:ADDRess:MIN`<br>`SEARch:TRIGger:RFFE:ADDRess:MAX` |

| | |
|---|---|
| Data pattern | SEARch:TRIGger:RFFE:DATA:DCON<br>SEARch:TRIGger:RFFE:DATA:DMIN<br>SEARch:TRIGger:RFFE:DATA:DMAX |
| Byte count | SEARch:TRIGger:RFFE:BCOunt:CONDition<br>SEARch:TRIGger:RFFE:BCOunt:MIN<br>SEARch:TRIGger:RFFE:BCOunt:MAX |
| Data index | SEARch:TRIGger:RFFE:DATA:ICONdition<br>SEARch:TRIGger:RFFE:DATA:IMIN<br>SEARch:TRIGger:RFFE:DATA:IMAX |

**SEARch:TRIGger:RFFE:SADD:CONDition** <SearchName>,<SIDOperator>
**SEARch:TRIGger:RFFE:SADD:CONDition?** <SearchName>
**SEARch:TRIGger:RFFE:ADDRess:CONDition** <SearchName>,<AddressOperator>
**SEARch:TRIGger:RFFE:ADDRess:CONDition?** <SearchName>
**SEARch:TRIGger:RFFE:BCOunt:CONDition** <SearchName>,<ByteCntOperator>
**SEARch:TRIGger:RFFE:BCOunt:CONDition?** <SearchName>
**SEARch:TRIGger:RFFE:DATA:DCON** <SearchName>,<DatPattOptor>
**SEARch:TRIGger:RFFE:DATA:DCON?** <SearchName>

Sets the operator to search for a specific pattern or a range

**Parameters:**

<DatPattOptor> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set using the corresponding SEARch:TRIGger:RFFE:...:MIN command.

**INRange | OORange**
In range / Out of range: Set the minimum and maximum value using the corresponding SEARch:TRIGger:RFFE:...:MIN and SEARch:TRIGger:RFFE:...:MAX

*RST: EQUal

**Parameters for setting and query:**

<SearchName> String parameter

**Example:** See "Searching RFFE data" on page 1188

**Manual operation:** See "Data" on page 700

**SEARch:TRIGger:RFFE:DATA:ICONdition** <SearchName>,<DataIdxOperator>
**SEARch:TRIGger:RFFE:DATA:ICONdition?** <SearchName>

Sets the operator to define the data frames in which the data pattern is searched

**Parameters:**

<DataIdxOperator> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal | LTHan | LETHan | GTHan | GETHan**
Equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set using `SEARch:TRIGger:RFFE:DATA:IMIN`.

**INRange = RANGe**
In range: Set the minimum and maximum value using `SEARch:TRIGger:RFFE:DATA:IMIN` and `SEARch:TRIGger:RFFE:DATA:IMAX`.

*RST: INRange

**Parameters for setting and query:**

<SearchName> String parameter

**Manual operation:** See "Index" on page 700

---

**SEARch:TRIGger:RFFE:SADD:MIN** <SearchName>,<SIDMin>
**SEARch:TRIGger:RFFE:SADD:MIN?** <SearchName>
**SEARch:TRIGger:RFFE:ADDRess:MIN** <SearchName>,<AddressMin>
**SEARch:TRIGger:RFFE:ADDRess:MIN?** <SearchName>
**SEARch:TRIGger:RFFE:BCOunt:MIN** <SearchName>,<ByteCountMin>
**SEARch:TRIGger:RFFE:BCOunt:MIN?** <SearchName>
**SEARch:TRIGger:RFFE:DATA:IMIN** <SearchName>,<DataIndexMin>
**SEARch:TRIGger:RFFE:DATA:IMIN?** <SearchName>
**SEARch:TRIGger:RFFE:DATA:DMIN** <SearchName>,<DataPatternMin>
**SEARch:TRIGger:RFFE:DATA:DMIN?** <SearchName>

Specifies a pattern, or sets the the start value of a pattern range.

**Parameters:**

<DataPatternMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**

<SearchName> String parameter

**Example:** See "Searching RFFE data" on page 1188

**Manual operation:** See "Data" on page 700

**SEARch:TRIGger:RFFE:SADD:MAX** <SearchName>,<SIDMax>
**SEARch:TRIGger:RFFE:SADD:MAX?** <SearchName>
**SEARch:TRIGger:RFFE:ADDRess:MAX** <SearchName>,<AddressMax>
**SEARch:TRIGger:RFFE:ADDRess:MAX?** <SearchName>
**SEARch:TRIGger:RFFE:BCOunt:MAX** <SearchName>,<ByteCountMax>
**SEARch:TRIGger:RFFE:BCOunt:MAX?** <SearchName>
**SEARch:TRIGger:RFFE:DATA:IMAX** <SearchName>,<DataIndexMax>
**SEARch:TRIGger:RFFE:DATA:IMAX?** <SearchName>
**SEARch:TRIGger:RFFE:DATA:DMAX** <SearchName>,<DataPatternMax>
**SEARch:TRIGger:RFFE:DATA:DMAX?** <SearchName>

Sets the the end value of an range if the corresponding condition is set to `INRange` or `OORange`. See `SEARch:TRIGger:RFFE:...:CON`

**Parameters:**
<DataPatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Data" on page 700

**SEARch:TRIGger:RFFE:INTerrupt** <SearchName>,<Interrupt>
**SEARch:TRIGger:RFFE:INTerrupt?** <SearchName>

Defines the pattern of the interrrupt identification sequence, which consists of interrupt slots 15 to 0.

**Parameters:**
<Interrupt> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName> String parameter

**Firmware/software:** FW 3.30

**Manual operation:** See "Data" on page 700

**SEARch:TRIGger:RFFE:ERRor:LENGth** <SearchName>,<LengthError>
**SEARch:TRIGger:RFFE:ERRor:LENGth?** <SearchName>

Enables the search for length errors - when an incorrect length of the command sequence has been found.

**Parameters:**
<LengthError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Error conditions" on page 700

---

**SEARch:TRIGger:RFFE:ERRor:PARity** <SearchName>,<ParityError>
**SEARch:TRIGger:RFFE:ERRor:PARity?** <SearchName>

Enables the search for incorrect parity bits.

**Parameters:**
<ParityError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Error conditions" on page 700

---

**SEARch:TRIGger:RFFE:ERRor:SSC** <SearchName>,<SSCError>
**SEARch:TRIGger:RFFE:ERRor:SSC?** <SearchName>

Enables the search for SSC errors - when no valid SSC sequence has been found after the idle time.

**Parameters:**
<SSCError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName> String parameter

**Manual operation:** See "Error conditions" on page 700

---

**SEARch:TRIGger:RFFE:ERRor:GAP** <SearchName>,<GapError>
**SEARch:TRIGger:RFFE:ERRor:GAP?** <SearchName>

Enables the search for RFFE gap time error.

**Parameters:**
<GapError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error conditions" on page 700

---

**SEARch:TRIGger:RFFE:ERRor:VERSion** <SearchName>,<VersionError>
**SEARch:TRIGger:RFFE:ERRor:VERSion?** <SearchName>

Enables the search for RFFE vresion error.

**Parameters:**
<VersionError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error conditions" on page 700

### 24.17.13.5 Search results

The search on decoded RFFE data returns the same results as the queries for decode results.

In all SEARch:RESult:RFFE commands, the suffix <m> selects the command sequence. Suffix <n> selects index of byte inside a command sequence.

In search result commands, you have to specify the <SearchName> parameter. It is a string parameter that contains the search definition name.

For a description of the returned values, see the corresponding commands in Chapter 24.17.13.3, "Decode results", on page 1932.

For programming example, see "Searching RFFE data" on page 1188.

SEARch:RESult:RFFE:FCOunt? 1945
SEARch:RESult:RFFE:SEQuence<m>:TYPE? 1945
SEARch:RESult:RFFE:SEQuence<m>:STATe? 1946
SEARch:RESult:RFFE:SEQuence<m>:STARt? 1946
SEARch:RESult:RFFE:SEQuence<m>:SADD? 1946
SEARch:RESult:RFFE:SEQuence<m>:STOP? 1947
SEARch:RESult:RFFE:SEQuence<m>:ADDRess? 1947
SEARch:RESult:RFFE:SEQuence<m>:BCOunt? 1947
SEARch:RESult:RFFE:SEQuence<m>:DATA? 1948
SEARch:RESult:RFFE:SEQuence<m>:SYMBol? 1948
SEARch:RESult:RFFE:SEQuence<m>:BYTE<n>:STATe? 1948
SEARch:RESult:RFFE:SEQuence<m>:BYTE<n>:VALue? 1948

---

**SEARch:RESult:RFFE:FCOunt?** <SearchName>

**Query parameters:**
<SearchName>

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:RFFE:SEQuence<m>:TYPE?** <SearchName>

Returns the type of the specified command sequence.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameType> UNDESC | RZWR | RWR | RRD | ERWR | ERRD | ERWL | ERRL | MASR | MASW | MASOHO | IRSUM | MSKW | MCTR | MCTW | UNDEF | ERRor

See `SEARch:TRIGger:RFFE:TYPE`.

*RST: RZWR

**Usage:** Query only

---

**SEARch:RESult:RFFE:SEQuence<m>:STATe?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameState> OK | SSC | VERSion | PARity | LENGth | BPERR | GAP | NORESPONSE | INComplete

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:RFFE:SEQuence<m>:STARt?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameStart> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:RFFE:SEQuence<m>:SADD?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameSID> Range: 0 to 15
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:RFFE:SEQuence<m>:STOP?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameStop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:RFFE:SEQuence<m>:ADDRess?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameAddress> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:RFFE:SEQuence<m>:BCOunt?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<ByteCount> Range: 0 to 256
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:RFFE:SEQuence<m>:DATA?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

**SEARch:RESult:RFFE:SEQuence<m>:SYMBol?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Translation>

**Usage:** Query only

**SEARch:RESult:RFFE:SEQuence<m>:BYTE<n>:STATe?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<ByteState> OK | PARity
OK: the byte is valid.
PARity: parity error
*RST: OK

**Usage:** Query only

**SEARch:RESult:RFFE:SEQuence<m>:BYTE<n>:VALue?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**

<ByteValue> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

## 24.17.14 D-PHY (option R&S RTP-K42)


- D-PHY configuration ..... 1949
- D-PHY trigger ..... 1955
- D-PHY decode results ..... 1962
- D-PHY search settings ..... 1968
- D-PHY search results ..... 1977


### 24.17.14.1 D-PHY configuration


BUS<m>:DPHY:CONFig ..... 1950
BUS<m>:DPHY:CP:PROBe ..... 1950
BUS<m>:DPHY:CP:SOURce ..... 1950
BUS<m>:DPHY:DRATe ..... 1951
BUS<m>:DPHY:DSPData ..... 1951
BUS<m>:DPHY:THCoupling ..... 1951
BUS<m>:DPHY:THPReset ..... 1952
BUS<m>:DPHY:VARiant ..... 1952
BUS<m>:DPHY:DNZero:LPOWer:THUPper ..... 1952
BUS<m>:DPHY:DPZero:LPOWer:THUPper ..... 1952
BUS<m>:DPHY:DNZero:LPOWer:THLower ..... 1953
BUS<m>:DPHY:DPZero:LPOWer:THLower ..... 1953
BUS<m>:DPHY:DPZero:SOURce ..... 1953
BUS<m>:DPHY:DNZero:SOURce ..... 1953
BUS<m>:DPHY:DPONe:SOURce ..... 1953
BUS<m>:DPHY:DPTWo:SOURce ..... 1953
BUS<m>:DPHY:DPTHree:SOURce ..... 1953
BUS<m>:DPHY:DNZero:PROBe ..... 1953
BUS<m>:DPHY:DPZero:PROBe ..... 1953
BUS<m>:DPHY:DPONe:PROBe ..... 1954
BUS<m>:DPHY:DPTWo:PROBe ..... 1954
BUS<m>:DPHY:DPTHree:PROBe ..... 1954
BUS<m>:DPHY:CP:HSPeed:HYSTeresis ..... 1954
BUS<m>:DPHY:DPZero:HSPeed:HYSTeresis ..... 1954
BUS<m>:DPHY:DPONe:HSPeed:HYSTeresis ..... 1954
BUS<m>:DPHY:DPTWo:HSPeed:HYSTeresis ..... 1954
BUS<m>:DPHY:DPTHree:HSPeed:HYSTeresis ..... 1954
BUS<m>:DPHY:CP:HSPeed:THReshold ..... 1954
BUS<m>:DPHY:DPZero:HSPeed:THReshold ..... 1954
BUS<m>:DPHY:DPONe:HSPeed:THReshold ..... 1954

BUS<m>:DPHY:DPTWo:HSPeed:THReshold ..... 1954
BUS<m>:DPHY:DPTHree:HSPeed:THReshold ..... 1954
BUS<m>:DPHY:SMLanes ..... 1955

**BUS<m>:DPHY:CONFig** <Configuration>

Sets the number of data lanes and if the clock lane and low power are available.

**Suffix:**
<m> 1..4

**Parameters:**
<Configuration> D1CN | D1CL | D2CN | D2CL | D3CN | D3NL | D4NN

`D1CN`: 1 data lane, clock lane
`D1CL`: 1 data lane, clock lane, low power
`D2CN`: 2 data lanes, clock lane
`D2CL`: 2 data lanes, clock lane, low power
`D3CN`: 3 data lanes, clock lane
`D3CL`: 3 data lanes, clock lane, low power
`D4NN`: 4 data lanes and no clock

*RST: D4NN

**Manual operation:** See "Physical configuration" on page 703

**BUS<m>:DPHY:CP:PROBe** <ProbeCP>

Selects the type of probe used for the clock lane.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ProbeCP> DIFFerential | SINGle

*RST: SINGle

**Manual operation:** See "Probe" on page 704

**BUS<m>:DPHY:CP:SOURce** <ClockSource>

Selects the source of the clock lane.

Usually, the source is one of the analog channels. Reference and math waveforms are available as source if the trigger source is one of the analog channels but not the serial bus.

Alternatively, digital channels can be used if MSO option R&S RTP-B1 is installed. Digital and analog channels cannot be used at the same time.

For triggering on a serial bus, analog or digital channel sources are required.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ClockSource> NONE | C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: NONE

**Manual operation:** See "CP " on page 704

---

**BUS<m>:DPHY:DRATe** <DataRate>

Sets a data rate.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<DataRate> Range: 1000000 to 2500000000
Increment: 1000000
*RST: 800000000
Default unit: bps

**Manual operation:** See "Data Rate" on page 704

---

**BUS<m>:DPHY:DSPData** <DispDatVect>

Enables the display of the data vector, which means that the whole frame will be decoded.

When the display vector is disabled only the packet header will be decoded and not the actual data payload of the frame. This will speed up the decoding.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<DispDatVect> ON | OFF
*RST: OFF

**Manual operation:** See "Data Rate" on page 704

---

**BUS<m>:DPHY:THCoupling** <ThresCpl>

Enables coupling, i.e. the same threshold and hysteresis value is used for all lanes.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ThresCpl> ON | OFF
*RST: ON

**Manual operation:** See "Couple thresholds" on page 705

---

**BUS<m>:DPHY:THPReset** <ThresholdPreset>

Prests the threshold and hysteresis values of the high speed data lanes. A preset sets the low power threshold to 1.20V and high speed threshold to 200 mV.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ThresholdPreset> LPHS | MANual
*RST: LPHS

---

**BUS<m>:DPHY:VARiant** <ProtSel>

Selects the protocol running on the interface.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ProtSel> DSI | CSI2
*RST: CSI2

**Manual operation:** See "Variant" on page 703

---

**BUS<m>:DPHY:DNZero:LPOWer:THUPper** <ThresLPDN0High>
**BUS<m>:DPHY:DPZero:LPOWer:THUPper** <ThresLPDP0High>

Sets the upper threshold value for the respective lane used for the low power mode.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ThresLPDP0High> Range: -1.5 to 1.5
Increment: 0.01
*RST: 0.88
Default unit: V

**Manual operation:** See "DP0 High / DN0 High" on page 705

**BUS<m>:DPHY:DNZero:LPOWer:THLower** <ThresLPDN0Low>
**BUS<m>:DPHY:DPZero:LPOWer:THLower** <ThresLPDP0Low>

Sets the lower threshold value for the respective lane used for the low power mode.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ThresLPDP0Low> Range: -1.5 to 1.5
Increment: 0.01
*RST: 0.55
Default unit: V

**Manual operation:** See "DP0 Low / DN0 Low" on page 705

**BUS<m>:DPHY:DPZero:SOURce** <DataSource>

Selects the source of the DP0 data line.

**Suffix:**
<m> 1..4

**Parameters:**
<DataSource> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: C1W1

**Manual operation:** See "DP0/DN0" on page 704

**BUS<m>:DPHY:DNZero:SOURce** <DataSource>
**BUS<m>:DPHY:DPONe:SOURce** <DataSource>
**BUS<m>:DPHY:DPTWo:SOURce** <DataSource>
**BUS<m>:DPHY:DPTHree:SOURce** <DataSource>

Selects the source of the corresponding data line.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<DataSource> NONE | C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: DNZero: NONE, DPONe: C2W1, DPTWo: C3W1, DPTHree: C4W1

**Manual operation:** See "DP1/DP2/DP3" on page 704

**BUS<m>:DPHY:DNZero:PROBe** <ProbeDN0>
**BUS<m>:DPHY:DPZero:PROBe** <ProbeDP0>

**BUS<m>:DPHY:DPONe:PROBe** <ProbeDP1>
**BUS<m>:DPHY:DPTWo:PROBe** <ProbeDP2>
**BUS<m>:DPHY:DPTHree:PROBe** <ProbeDP3>

Selects the type of probe used for the corresponding data lane.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ProbeDP3> DIFFerential | SINGle
*RST: SINGle

**Manual operation:** See "Probe" on page 704

---

**BUS<m>:DPHY:CP:HSPeed:HYSTeresis** <HysteresisHSCP>
**BUS<m>:DPHY:DPZero:HSPeed:HYSTeresis** <HysteresisHSDP0>
**BUS<m>:DPHY:DPONe:HSPeed:HYSTeresis** <HysteresisHSDP1>
**BUS<m>:DPHY:DPTWo:HSPeed:HYSTeresis** <HysteresisHSDP2>
**BUS<m>:DPHY:DPTHree:HSPeed:HYSTeresis** <HysteresisHSDP3>

Sets a value for the hysteresis of the respective lane.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<HysteresisHSDP3> Range: 0 to 0.5
Increment: 0.01
*RST: 0.2
Default unit: V

**Manual operation:** See "HS Hysteresis" on page 705

---

**BUS<m>:DPHY:CP:HSPeed:THReshold** <ThresholdHSCP>
**BUS<m>:DPHY:DPZero:HSPeed:THReshold** <ThresholdHSDP0>
**BUS<m>:DPHY:DPONe:HSPeed:THReshold** <ThresholdHSDP1>
**BUS<m>:DPHY:DPTWo:HSPeed:THReshold** <ThresholdHSDP2>
**BUS<m>:DPHY:DPTHree:HSPeed:THReshold** <ThresholdHSDP3>

Sets the threshold value for the digitization of the respective high speed data line.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ThresholdHSDP3> Range: -1 to 1
Increment: 0.01
*RST: 0.2
Default unit: V

**Manual operation:** See "HS Threshold" on page 704

---

**BUS<m>:DPHY:SMLanes** <Active>

Enables the display of the results in multiple lanes.

**Suffix:**
<m> 1..4

**Parameters:**
<Active> ON | OFF
*RST: ON

**Manual operation:** See "Show multiple lanes" on page 706

### 24.17.14.2 D-PHY trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:DPHY:TYPE ... 1956
TRIGger<m>:DPHY:DATA:CONDition ... 1956
TRIGger<m>:DPHY:DATA:MAX ... 1957
TRIGger<m>:DPHY:DATA:MIN ... 1957
TRIGger<m>:DPHY:DIDX:CONDition ... 1957
TRIGger<m>:DPHY:DIDX:MAX ... 1957
TRIGger<m>:DPHY:DIDX:MIN ... 1958
TRIGger<m>:DPHY:DTYPe:CONDition ... 1958
TRIGger<m>:DPHY:DTYPe:MAX ... 1958
TRIGger<m>:DPHY:DTYPe:MIN ... 1959
TRIGger<m>:DPHY:ESCMode:CONDition ... 1959
TRIGger<m>:DPHY:ESCMode:MAX ... 1959
TRIGger<m>:DPHY:ESCMode:MIN ... 1959
TRIGger<m>:DPHY:ESDTa:CONDition ... 1960
TRIGger<m>:DPHY:ESDTa:MAX ... 1960
TRIGger<m>:DPHY:ESDTa:MIN ... 1960
TRIGger<m>:DPHY:ESINdex:CONDition ... 1961
TRIGger<m>:DPHY:ESINdex:MAX ... 1961
TRIGger<m>:DPHY:ESINdex:MIN ... 1961
TRIGger<m>:DPHY:HSVC ... 1961
TRIGger<m>:DPHY:WCOunt:CONDition ... 1962
TRIGger<m>:DPHY:WCOunt:MAX ... 1962
TRIGger<m>:DPHY:WCOunt:MIN ... 1962

**TRIGger<m>:DPHY:TYPE** <Type>

Sets the type of frame to be triggered on.

**Parameters:**

<Type> HS_SOP | HS_EOP | HS_PH | HS_DATA | LP_ESC | LP_TURN | LP_HSRQ

**HS_SOP**
High speed start of packet

**HS_EOP**
High speed end of packet

**HS_PH**
High speed packet header

**HS_DATA**
High speed data

**LP_ESC**
Low power escape mode frame

**LP_TURN**
Low power lane turnaround

**LP_HSRQ**
Low power high speed request

*RST: HS_SOP

**Manual operation:** See "Type" on page 708

**TRIGger<m>:DPHY:DATA:CONDition** <Format>

Set the condition for the data value. You can define an exact value or a value range.

**Parameters:**

<Format> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:DPHY:DATA:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:DPHY:DATA:MIN` and `TRIGger<m>:DPHY:DATA:MAX`.

*RST: EQUal

**Manual operation:** See "Data Value" on page 710

**TRIGger<m>:DPHY:DATA:MAX** <DataMax>

Sets the the end value of a data type range if `TRIGger<m>:DPHY:DATA:CONDition` is set to `INRange` or `OORange`.

**Parameters:**

<DataMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data Value" on page 710

**TRIGger<m>:DPHY:DATA:MIN** <DataMin>

Specifies a data bit pattern, or sets the the start value of a pattern range.

**Parameters:**

<DataMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data Value" on page 710

**TRIGger<m>:DPHY:DIDX:CONDition** <Format>

Sets the condition for the data index. You can define an exact value or a value range.

**Parameters:**

<Format> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `TRIGger<m>:DPHY:DIDX:MIN` command.

**INRange = RANGe**
In range: Set the minimum and maximum value using `TRIGger<m>:DPHY:DIDX:MIN` and `TRIGger<m>:DPHY:DIDX:MAX`.

*RST: INRange

**Manual operation:** See "Data Index" on page 711

**TRIGger<m>:DPHY:DIDX:MAX** <DXSymbol>

Sets the the end value of a data index range if `TRIGger<m>:DPHY:DIDX:CONDition` is set to `INRange` or `RANge`.

**Parameters:**

<DXSymbol> Range: 0 to 65535
Increment: 1
*RST: 0

**Manual operation:** See "Data Index" on page 711

---

**TRIGger<m>:DPHY:DIDX:MIN** <SymbolType>

Specifies a data index minimum, or sets the the start value of a range.

**Parameters:**

<SymbolType> Range: 0 to 65535
Increment: 1
*RST: 0

**Manual operation:** See "Data Index" on page 711

---

**TRIGger<m>:DPHY:DTYPe:CONDition** <Format>

Set the condition for the data type. You can define an exact value or a value range

**Parameters:**

<Format> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `TRIGger<m>:DPHY:DTYPe:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:DPHY:DTYPe:MIN` and `TRIGger<m>:DPHY:DTYPe:MAX`.

*RST: EQUal

**Manual operation:** See "Data Type" on page 710

---

**TRIGger<m>:DPHY:DTYPe:MAX** <HSDataTypeMax>

Sets the the end value of a data type range if `TRIGger<m>:DPHY:DTYPe:CONDition` is set to `INRange` or `OORange`.

**Parameters:**

<HSDataTypeMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data Type" on page 710

**TRIGger<m>:DPHY:DTYPe:MIN** <HSDataTypeMin>

Specifies a data type pattern, or sets the the start value of a pattern range.

**Parameters:**

<HSDataTypeMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data Type" on page 710

**TRIGger<m>:DPHY:ESCMode:CONDition** <Format>

Set the condition for the escape mode. You can define an exact value or a value range

**Parameters:**

<Format> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `TRIGger<m>:DPHY:ESCMode:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:DPHY:ESCMode:MIN` and `TRIGger<m>:DPHY:ESCMode:MAX`.

*RST: EQUal

**Manual operation:** See "Escape mode" on page 711

**TRIGger<m>:DPHY:ESCMode:MAX** <LPEscapeModeMax>

Sets the the end value of an escape mode range if `TRIGger<m>:DPHY:ESCMode:CONDition` is set to `INRange` or `OORange`.

**Parameters:**

<LPEscapeModeMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Escape mode" on page 711

**TRIGger<m>:DPHY:ESCMode:MIN** <LPEscapeModeMin>

Specifies a escape mode bit pattern, or sets the the start value of a pattern range.

**Parameters:**

<LPEscapeModeMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Escape mode" on page 711

---

**TRIGger<m>:DPHY:ESDTa:CONDition** <Format>

Set the condition for the escape mode data value. You can define an exact value or a value range.

**Parameters:**

<Format> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:DPHY:ESDTa:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:DPHY:ESDTa:MIN` and `TRIGger<m>:DPHY:ESDTa:MAX`.

**OFF | ANY**
Any pattern is detected.

*RST: EQUal

**Manual operation:** See "Data Value" on page 710

---

**TRIGger<m>:DPHY:ESDTa:MAX** <LPEscapeDataMax>

Sets the the end value of a escape mode data type range if `TRIGger<m>:DPHY:ESDTa:CONDition` is set to `INRange` or `OORange`.

**Parameters:**

<LPEscapeDataMax>

**Manual operation:** See "Data Value" on page 710

---

**TRIGger<m>:DPHY:ESDTa:MIN** <LPEscapeDataMin>

Specifies an escape mode data bit pattern, or sets the the start value of a pattern range.

**Parameters:**

<LPEscpMdDatMin>

**Manual operation:** See "Data Value" on page 710

**TRIGger<m>:DPHY:ESINdex:CONDition** <Format>

Sets the condition for the escape mode data index. You can define an exact value or a value range.

**Parameters:**

<Format> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `TRIGger<m>:DPHY:ESINdex:MIN` command.

**INRange = RANGe**
In range: Set the minimum and maximum value using `TRIGger<m>:DPHY:ESINdex:MIN` and `TRIGger<m>:DPHY:ESINdex:MAX`.

*RST: INRange

**Manual operation:** See "Data Index" on page 711

**TRIGger<m>:DPHY:ESINdex:MAX** <DXSymbol>

Sets the the end value of a data index range if `TRIGger<m>:DPHY:ESINdex:CONDition` is set to `INRange` or `RANge`.

**Parameters:**

<DXSymbol> Range: 0 to 255
Increment: 1
*RST: 0

**Manual operation:** See "Data Index" on page 711

**TRIGger<m>:DPHY:ESINdex:MIN** <SymbolType>

Specifies an escape mode data index minimum, or sets the the start value of a range.

**Parameters:**

<SymbolType> Range: 0 to 255
Increment: 1
*RST: 0

**Manual operation:** See "Data Index" on page 711

**TRIGger<m>:DPHY:HSVC** <HSVC>

Sets the virtual channel to be triggered on.

**Parameters:**

<HSVC>

**Manual operation:** See "Virtual Channel" on page 710

---

**TRIGger<m>:DPHY:WCOunt:CONDition** <Format>

Set the condition for the word count. You can define an exact value or a value range.

**Parameters:**

<Format> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `TRIGger<m>:DPHY:WCOunt:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:DPHY:WCOunt:MIN` and `TRIGger<m>:DPHY:WCOunt:MAX`.

*RST: EQUal

**Manual operation:** See "Word Count" on page 710

---

**TRIGger<m>:DPHY:WCOunt:MAX** <HSWordCountMax>

Sets the the end value of a data type range if `TRIGger<m>:DPHY:WCOunt:CONDition` is set to `INRange` or `OORange`.

**Parameters:**

<HSWordCountMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Word Count" on page 710

---

**TRIGger<m>:DPHY:WCOunt:MIN** <HSWordCountMin>

Specifies a word bit pattern, or sets the the start value of a pattern range.

**Parameters:**

<HSWordCountMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Word Count" on page 710

### 24.17.14.3 D-PHY decode results

To show the results on the screen, use the following commands:

- `BUS<m>:RESult` on page 1638

BUS<m>:DPHY:RESult:FCOunt? ..... 1963
BUS<m>:DPHY:RESult:FRAMe<n>:CS? ..... 1963
BUS<m>:DPHY:RESult:FRAMe<n>:DATA? ..... 1963
BUS<m>:DPHY:RESult:FRAMe<n>:DTName? ..... 1964
BUS<m>:DPHY:RESult:FRAMe<n>:DTYPe? ..... 1964
BUS<m>:DPHY:RESult:FRAMe<n>:ECC? ..... 1965
BUS<m>:DPHY:RESult:FRAMe<n>:PACKet<o>:IDX? ..... 1965
BUS<m>:DPHY:RESult:FRAMe<n>:PACKet<o>:VALue? ..... 1965
BUS<m>:DPHY:RESult:FRAMe<n>:STARt? ..... 1966
BUS<m>:DPHY:RESult:FRAMe<n>:STATe? ..... 1966
BUS<m>:DPHY:RESult:FRAMe<n>:STOP? ..... 1967
BUS<m>:DPHY:RESult:FRAMe<n>:TYPE? ..... 1967
BUS<m>:DPHY:RESult:FRAMe<n>:NUMPackets? ..... 1968
BUS<m>:DPHY:RESult:FRAMe<n>:VCHannel? ..... 1968

### BUS<m>:DPHY:RESult:FCOunt?

Returns the number of decoded frames for the selected serial bus.

**Suffix:**
<m> 1..4

**Return values:**
<Count> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

### BUS<m>:DPHY:RESult:FRAMe<n>:CS?

Returns the checksum of the specified frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<CS> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

### BUS<m>:DPHY:RESult:FRAMe<n>:DATA?

Returns the data or word count value.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<WordCountOrData> 16 bit integer value
Range: 0 to 65535
Increment: 1
*RST: 0

**Example:**
```
BUS:DPHY:RESult:FRAMe2:DATA?
<-- 13245
```

**Usage:** Query only

---

**BUS<m>:DPHY:RESult:FRAMe<n>:DTName?**

Returns the data type name for the specified frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<DataTypeName>

**Usage:** Query only

---

**BUS<m>:DPHY:RESult:FRAMe<n>:DTYPe?**

Returns the data type for the specified frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<DataType> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:DPHY:RESult:FRAMe<n>:ECC?**

Returns the error correction code (ECC) of the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<ECC> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:DPHY:RESult:FRAMe<n>:PACKet<o>:IDX?**

Returns the index of the data payload.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the packet number.

**Return values:**

<PacketIndex> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:DPHY:RESult:FRAMe<n>:PACKet<o>:VALue?**

Returns the byte value of the data payload.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the packet number.

**Return values:**

<PacketValue> Range: 0 to 9223372036854775808
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:DPHY:RESult:FRAMe<n>:STARt?**

Returns the start time of the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:DPHY:RESult:FRAMe<n>:STATe?**

Returns the overall state of the frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<State> OK | CS | ECC | ECCWarning | LENGth | FORMat | UNKNOWN_FRAME | INComplete

**OK**
The frame is valid.

**LENGth**
Length error.

**UNKNOWN_FRAME**
Unknown frame type

**INComplete**
The sequence is not completely contained in the acquisition

*RST: OK

**Usage:** Query only

**BUS<m>:DPHY:RESult:FRAMe<n>:STOP?**

Returns the end time of the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:DPHY:RESult:FRAMe<n>:TYPE?**

Returns the type of frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<FrameType> HS_SP | HS_LP | HS_INCOMPLETE | LP_HSREQ | LP_TA | LP_ESC

**HS_SP**
High speed short packet frame

**HS_LP**
High speed long packet frame

**HS_INCOMPLETE**
Incomplete high speed frame

**LP_HSREQ**
High speed request frame

**LP_TA**
Low power lane turnaround frame

**LP_ESC**
Low power escape mode frame

*RST: HS_SP

**Usage:** Query only

**BUS<m>:DPHY:RESult:FRAMe<n>:NUMPackets?**

Returns the number of packets in the indicated frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<NumPackets> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:DPHY:RESult:FRAMe<n>:VCHannel?**

Returns the number of virtual channels.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<VirtualChannel> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only

### 24.17.14.4 D-PHY search settings

SEARch:TRIGger:DPHY:TYPE....1969
SEARch:TRIGger:DPHY:DATA:CONDition....1969
SEARch:TRIGger:DPHY:DATA:MAX....1970
SEARch:TRIGger:DPHY:DATA:MIN....1970
SEARch:TRIGger:DPHY:DIDX:CONDition....1971
SEARch:TRIGger:DPHY:DIDX:MAX....1971
SEARch:TRIGger:DPHY:DIDX:MIN....1971
SEARch:TRIGger:DPHY:DTYPe:CONDition....1972
SEARch:TRIGger:DPHY:DTYPe:MAX....1972
SEARch:TRIGger:DPHY:DTYPe:MIN....1972
SEARch:TRIGger:DPHY:ESCMode:CONDition....1973
SEARch:TRIGger:DPHY:ESCMode:MAX....1973
SEARch:TRIGger:DPHY:ESCMode:MIN....1973
SEARch:TRIGger:DPHY:ESDTa:CONDition....1974
SEARch:TRIGger:DPHY:ESDTa:MAX....1974

SEARch:TRIGger:DPHY:ESDTa:MIN ..... 1975
SEARch:TRIGger:DPHY:ESINdex:CONDition ..... 1975
SEARch:TRIGger:DPHY:ESINdex:MAX ..... 1975
SEARch:TRIGger:DPHY:ESINdex:MIN ..... 1976
SEARch:TRIGger:DPHY:HSVC ..... 1976
SEARch:TRIGger:DPHY:WCOunt:CONDition ..... 1976
SEARch:TRIGger:DPHY:WCOunt:MAX ..... 1977
SEARch:TRIGger:DPHY:WCOunt:MIN ..... 1977

**SEARch:TRIGger:DPHY:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:DPHY:TYPE?** <SearchName>

Sets the type of frame to be searched for.

**Parameters:**

<Type> HS_SOP | HS_EOP | HS_PH | HS_DATA | LP_ESC | LP_TURN | LP_HSRQ

**HS_SOP**
High speed start of packet

**HS_EOP**
High speed end of packet

**HS_PH**
High speed packet header

**HS_DATA**
High speed data

**LP_ESC**
Low power escape mode frame

**LP_TURN**
Low power lane turnaround

**LP_HSRQ**
Low power high speed request

*RST: HS_SOP

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Type" on page 714

**SEARch:TRIGger:DPHY:DATA:CONDition** <SearchName>,<Format>
**SEARch:TRIGger:DPHY:DATA:CONDition?** <SearchName>

Set the condition for the data value. You can define an exact value or a value range.

**Parameters:**

<Format> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:DPHY:DATA:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:DPHY:DATA:MIN` and `SEARch:TRIGger:DPHY:DATA:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Value" on page 715

---

**SEARch:TRIGger:DPHY:DATA:MAX** <SearchName>,<DataMax>
**SEARch:TRIGger:DPHY:DATA:MAX?** <SearchName>

Sets the the end value of a data type range if `SEARch:TRIGger:DPHY:DATA:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<DataMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Value" on page 715

---

**SEARch:TRIGger:DPHY:DATA:MIN** <SearchName>,<DataMin>
**SEARch:TRIGger:DPHY:DATA:MIN?** <SearchName>

Specifies a data bit pattern, or sets the the start value of a pattern range.

**Parameters:**
<DataMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Value" on page 715

**SEARch:TRIGger:DPHY:DIDX:CONDition** <SearchName>,<Format>
**SEARch:TRIGger:DPHY:DIDX:CONDition?** <SearchName>

Sets the condition for the data index. You can define an exact value or a value range.

**Parameters:**

<Format> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `SEARch:TRIGger:DPHY:DIDX:MIN` command.

**INRange = RANGe**
In range: Set the minimum and maximum value using `SEARch:TRIGger:DPHY:DIDX:MIN` and `SEARch:TRIGger:DPHY:DIDX:MAX`.

*RST: INRange

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Index" on page 716

---

**SEARch:TRIGger:DPHY:DIDX:MAX** <SearchName>,<DXSymbol>
**SEARch:TRIGger:DPHY:DIDX:MAX?** <SearchName>

Sets the the end value of a data index range if `SEARch:TRIGger:DPHY:DIDX:CONDition` is set to `INRange` or `RANge`.

**Parameters:**

<DXSymbol> Range: 0 to 65535
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Index" on page 716

---

**SEARch:TRIGger:DPHY:DIDX:MIN** <SearchName>,<SymbolType>
**SEARch:TRIGger:DPHY:DIDX:MIN?** <SearchName>

Specifies a data index minimum, or sets the the start value of a range.

**Parameters:**

<SymbolType> Range: 0 to 65535
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Index" on page 716

---

**SEARch:TRIGger:DPHY:DTYPe:CONDition** <SearchName>,<Format>
**SEARch:TRIGger:DPHY:DTYPe:CONDition?** <SearchName>

Set the condition for the data type. You can define an exact value or a value range

**Parameters:**
<Format> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `SEARch:TRIGger:DPHY:DTYPe:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:DPHY:DATA:MIN` and `SEARch:TRIGger:DPHY:DATA:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Type" on page 715

---

**SEARch:TRIGger:DPHY:DTYPe:MAX** <SearchName>,<HSDataTypeMax>
**SEARch:TRIGger:DPHY:DTYPe:MAX?** <SearchName>

Sets the the end value of a data type range if `SEARch:TRIGger:DPHY:DTYPe:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<HSDataTypeMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Type" on page 715

---

**SEARch:TRIGger:DPHY:DTYPe:MIN** <SearchName>,<HSDataTypeMin>
**SEARch:TRIGger:DPHY:DTYPe:MIN?** <SearchName>

Specifies a data type pattern, or sets the the start value of a pattern range.

**Parameters:**
<HSDataTypeMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Type" on page 715

---

**SEARch:TRIGger:DPHY:ESCMode:CONDition** <SearchName>,<Format>
**SEARch:TRIGger:DPHY:ESCMode:CONDition?** <SearchName>

Set the condition for the escape mode. You can define an exact value or a value range

**Parameters:**
<Format> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `SEARch:TRIGger:DPHY:ESCMode:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:DPHY:ESCMode:MIN` and `SEARch:TRIGger:DPHY:ESCMode:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Escape mode" on page 716

---

**SEARch:TRIGger:DPHY:ESCMode:MAX** <SearchName>,<LPEscapeModeMax>
**SEARch:TRIGger:DPHY:ESCMode:MAX?** <SearchName>

Sets the the end value of an escape mode range if `SEARch:TRIGger:DPHY:ESCMode:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<LPEscapeModeMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Escape mode" on page 716

---

**SEARch:TRIGger:DPHY:ESCMode:MIN** <SearchName>,<LPEscapeModeMin>
**SEARch:TRIGger:DPHY:ESCMode:MIN?** <SearchName>

Specifies a escape mode bit pattern, or sets the the start value of a pattern range.

**Parameters:**
<LPEscapeModeMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Escape mode" on page 716

---

**SEARch:TRIGger:DPHY:ESDTa:CONDition** <SearchName>,<Format>
**SEARch:TRIGger:DPHY:ESDTa:CONDition?** <SearchName>

Set the condition for the escape mode data value. You can define an exact value or a value range.

**Parameters:**
<Format> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:DPHY:ESDTa:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:DPHY:ESDTa:MIN` and `SEARch:TRIGger:DPHY:ESDTa:MAX`.

**OFF | ANY**
Any pattern is detected.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Value" on page 715

---

**SEARch:TRIGger:DPHY:ESDTa:MAX** <SearchName>,<LPEscapeDataMax>
**SEARch:TRIGger:DPHY:ESDTa:MAX?** <SearchName>

Sets the the end value of a escape mode data type range if `SEARch:TRIGger:DPHY:ESDTa:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<LPEscapeDataMax>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Value" on page 715

**SEARch:TRIGger:DPHY:ESDTa:MIN** <SearchName>,<LPEscapeDataMin>
**SEARch:TRIGger:DPHY:ESDTa:MIN?** <SearchName>

Specifies an escape mode data bit pattern, or sets the the start value of a pattern range.

**Parameters:**
<LPEscapeDataMin>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Value" on page 715

**SEARch:TRIGger:DPHY:ESINdex:CONDition** <SearchName>,<Format>
**SEARch:TRIGger:DPHY:ESINdex:CONDition?** <SearchName>

Sets the condition for the escape mode data index. You can define an exact value or a value range.

**Parameters:**
<Format> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `SEARch:TRIGger:DPHY:ESINdex:MIN` command.

**INRange = RANGe**
In range: Set the minimum and maximum value using `SEARch:TRIGger:DPHY:ESINdex:MIN` and `SEARch:TRIGger:DPHY:ESINdex:MAX`.

*RST: INRange

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Index" on page 716

**SEARch:TRIGger:DPHY:ESINdex:MAX** <SearchName>,<DXSymbol>
**SEARch:TRIGger:DPHY:ESINdex:MAX?** <SearchName>

Sets the the end value of a data index range if `SEARch:TRIGger:DPHY:ESINdex:CONDition` is set to `INRange` or `RANge`.

**Parameters:**
<DXSymbol> Range: 0 to 255
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Index" on page 716

---

**SEARch:TRIGger:DPHY:ESINdex:MIN** <SearchName>,<SymbolType>
**SEARch:TRIGger:DPHY:ESINdex:MIN?** <SearchName>

Specifies an escape mode data index minimum, or sets the the start value of a range.

**Parameters:**
<SymbolType> Range: 0 to 255
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data Index" on page 716

---

**SEARch:TRIGger:DPHY:HSVC** <SearchName>,<HSVC>
**SEARch:TRIGger:DPHY:HSVC?** <SearchName>

Sets the virtual channel to be searched for.

**Parameters:**
<HSVC>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Virtual Channel" on page 714

---

**SEARch:TRIGger:DPHY:WCOunt:CONDition** <SearchName>,<Format>
**SEARch:TRIGger:DPHY:WCOunt:CONDition?** <SearchName>

Set the condition for the word count. You can define an exact value or a value range.

**Parameters:**
<Format> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `SEARch:TRIGger:DPHY:WCOunt:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:DPHY:WCOunt:MIN` and `SEARch:TRIGger:DPHY:WCOunt:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Word Count/ Data" on page 715

---

**SEARch:TRIGger:DPHY:WCOunt:MAX** <SearchName>,<HSWordCountMax>
**SEARch:TRIGger:DPHY:WCOunt:MAX?** <SearchName>

Sets the the end value of a data type range if `SEARch:TRIGger:DPHY:WCOunt:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<HSWordCountMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Word Count/ Data" on page 715

---

**SEARch:TRIGger:DPHY:WCOunt:MIN** <SearchName>,<HSWordCountMin>
**SEARch:TRIGger:DPHY:WCOunt:MIN?** <SearchName>

Specifies a word bit pattern, or sets the the start value of a pattern range.

**Parameters:**
<HSWordCountMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Word Count/ Data" on page 715

### 24.17.14.5 D-PHY search results

In all `SEARch:RESult:DPHY:FRAMe<m>` commands, the suffix <m> selects the frame number in the list of search results.

SEARch:RESult:DPHY:FCOunt? 1978
SEARch:RESult:DPHY:FRAMe<m>:CS? 1978
SEARch:RESult:DPHY:FRAMe<m>:DATA? 1978
SEARch:RESult:DPHY:FRAMe<m>:DTName? 1978
SEARch:RESult:DPHY:FRAMe<m>:DTYPe? 1979
SEARch:RESult:DPHY:FRAMe<m>:ECC? 1979
SEARch:RESult:DPHY:FRAMe<m>:PACKet<n>:IDX? 1979
SEARch:RESult:DPHY:FRAMe<m>:PACKet<n>:VALue? 1980
SEARch:RESult:DPHY:FRAMe<m>:STARt? 1980
SEARch:RESult:DPHY:FRAMe<m>:STATe? 1980
SEARch:RESult:DPHY:FRAMe<m>:STOP? 1981
SEARch:RESult:DPHY:FRAMe<m>:TYPE? 1981
SEARch:RESult:DPHY:FRAMe<m>:VCHannel? 1982

**SEARch:RESult:DPHY:FCOunt?** <SearchName>

Returns the number of frames within the search result for the selected serial bus.

**Query parameters:**
<SearchName>

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:DPHY:FRAMe<m>:CS?** <SearchName>

Returns the checksum for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<CS> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:DPHY:FRAMe<m>:DATA?** <SearchName>

Returns the data or word count value for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<WordCountOrData> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:DPHY:FRAMe<m>:DTName?** <SearchName>

Returns the data type name for the specified frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<DataTypeName>

**Usage:** Query only

---

**SEARch:RESult:DPHY:FRAMe<m>:DTYPe?** <SearchName>

Returns the data type for the specified frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<DataType> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:DPHY:FRAMe<m>:ECC?** <SearchName>

Returns the error correction code (ECC) of the specified frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<ECC> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:DPHY:FRAMe<m>:PACKet<n>:IDX?** <SearchName>

Returns the index of the data payload for the selected frame within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<PacketIndex> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:DPHY:FRAMe<m>:PACKet<n>:VALue?** <SearchName>

Returns the byte value of the data payload for the selected frame within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<PacketValue> Range: 0 to 9223372036854775808
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:DPHY:FRAMe<m>:STARt?** <SearchName>

Returns the start time of the specified frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:DPHY:FRAMe<m>:STATe?** <SearchName>

Returns the overall state of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | CS | ECC | ECCWarning | LENGth | FORMat | UNKNOWN_FRAME | INComplete

**OK**
The frame is valid.

**LENGth**
Length error.

**UNKNOWN_FRAME**
Unknown frame type

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:DPHY:FRAMe<m>:STOP?** <SearchName>

Returns the end time for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:DPHY:FRAMe<m>:TYPE?** <SearchName>

Returns the type of frame for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameType> HS_SP | HS_LP | HS_INCOMPLETE | LP_HSREQ | LP_TA | LP_ESC

**HS_SP**
High speed short packet frame

**HS_LP**
High speed long packet frame

**HS_INCOMPLETE**
Incomplete high speed frame

**LP_HSREQ**
High speed request frame

**LP_TA**
Low power lane turnaround frame

**LP_ESC**
Low power escape mode frame

*RST: HS_SP

**Usage:** Query only

---

**SEARch:RESult:DPHY:FRAMe<m>:VCHannel?** <SearchName>

Returns the number of virtual channels for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<VirtualChannel> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only

## 24.17.15 M-PHY (option R&S RTP-K44)


- Configuration....1982
- Trigger M-PHY....1985
- Decode results M-PHY....2002
- Search settings M-PHY....2009
- Search results M-PHY....2026
- Trigger USB SSIC (Option R&S RTP-K64)....2031
- Search settings USB SSIC (Option R&S RTP-K64)....2036


### 24.17.15.1 Configuration


BUS<m>:MPHY:VARiant....1983
BUS<m>:MPHY:DLANes....1983
BUS<m>:MPHY:DZERo:SOURce....1983
BUS<m>:MPHY:DONE:SOURce....1984
BUS<m>:MPHY:DTWO:SOURce....1984
BUS<m>:MPHY:DTHRee:SOURce....1984
BUS<m>:MPHY:DZERo:THReshold....1984
BUS<m>:MPHY:DONE:THReshold....1984

BUS<m>:MPHY:DTWO:THReshold ..... 1984
BUS<m>:MPHY:DTHRee:THReshold ..... 1984
BUS<m>:MPHY:THCoupling ..... 1984
BUS<m>:MPHY:THPReset ..... 1985
BUS<m>:MPHY:SCRMode ..... 1985
BUS<m>:MPHY:SMLanes ..... 1985

**BUS<m>:MPHY:VARiant** <Protocol>

Selects the protocol running on the interface.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Protocol> MPHY | UNIPRO | SSIC
*RST: MPHY

**Manual operation:** See "Variant" on page 720

**BUS<m>:MPHY:DLANes** <DataLaneCount>

Sets the number of data lanes.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<DataLaneCount> Range: 1 to 4
Increment: 1
*RST: 4

**Manual operation:** See "Lanes" on page 721

**BUS<m>:MPHY:DZERo:SOURce** <DataSource>

Sets the source for the lane.

**Suffix:**
<m> 1..4

**Parameters:**
<DataSource> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: C1W1

**Manual operation:** See "Lane 0 / Lane 1 / Lane 2 / Lane 3: Channel" on page 721

**BUS<m>:MPHY:DONE:SOURce** <DataSource>
**BUS<m>:MPHY:DTWO:SOURce** <DataSource>
**BUS<m>:MPHY:DTHRee:SOURce** <DataSource>

Selects the source of the corresponding data lane.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<DataSource> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | NONE
*RST: DONe: C2W1, DTWo: C3W1, DTHree: C4W1

**Manual operation:** See "Lane 0 / Lane 1 / Lane 2 / Lane 3: Channel" on page 721

**BUS<m>:MPHY:DZERo:THReshold** <ThresholdD0>
**BUS<m>:MPHY:DONE:THReshold** <ThresholdD1>
**BUS<m>:MPHY:DTWO:THReshold** <ThresholdD2>
**BUS<m>:MPHY:DTHRee:THReshold** <ThresholdD3>

Sets the threshold value for the respective lane.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ThresholdD1>
<ThresholdD2>
<ThresholdD3>
Range: 0 to 1
Increment: 0.01
*RST: 0.1
Default unit: V

**Manual operation:** See "Threshold setup " on page 721

**BUS<m>:MPHY:THCoupling** <Coupling>

Enables the same threshold value to be used for all lanes.

When the threshold coupling is disabled, changes to the threshold value only affect the respective lane.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<Coupling> ON | OFF
*RST: ON

**Manual operation:** See "Threshold setup " on page 721

**BUS<m>:MPHY:THPReset** <ThresholdPreset>

Selects the predefined value to preset the threshold value of the data lanes.

**Suffix:**
<m> 1..4
Selects the serial bus.

**Parameters:**
<ThresholdPreset> THRES_60MV | THRES_120MV | THRES_240MV | MANual

**THRES_60MV | THRES_120MV | THRES_240MV**
Sets the threshold value of the data lane to the selected predefined value.

**MANual**
Automatically switches to this option if you edit the threshold text box manually.

*RST: MANual

**BUS<m>:MPHY:SCRMode** <SSICScrambleMd>

Sets if the SSIC data is scrambled or unscrambled.

**Suffix:**
<m> 1..4

**Parameters:**
<SSICScrambleMd> SCRM | NSCR

`SCRM`: scrambled
`NSCR`: unscrambled

*RST: SCRM

**Manual operation:** See "Scrambled Mode" on page 722

**BUS<m>:MPHY:SMLanes** <Active>

Enables the display of the results in multiple lanes.

**Suffix:**
<m> 1..4

**Parameters:**
<Active> ON | OFF

*RST: ON

**Manual operation:** See "Show multiple lanes" on page 722

### 24.17.15.2 Trigger M-PHY

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:MPHY:TYPE ..... 1987
TRIGger<m>:MPHY:LCCType ..... 1987
TRIGger<m>:MPHY:LWONe:CONDition ..... 1988
TRIGger<m>:MPHY:LWONe:MIN ..... 1989
TRIGger<m>:MPHY:LWONe:MAX ..... 1989
TRIGger<m>:MPHY:LWTWo:CONDition ..... 1989
TRIGger<m>:MPHY:LWTWo:MIN ..... 1990
TRIGger<m>:MPHY:LWTWo:MAX ..... 1990
TRIGger<m>:MPHY:LWTHree:CONDition ..... 1990
TRIGger<m>:MPHY:LWTHree:MIN ..... 1991
TRIGger<m>:MPHY:LWTHree:MAX ..... 1991
TRIGger<m>:MPHY:LWFour:CONDition ..... 1991
TRIGger<m>:MPHY:LWFour:MIN ..... 1992
TRIGger<m>:MPHY:LWFour:MAX ..... 1992
TRIGger<m>:MPHY:TC ..... 1992
TRIGger<m>:MPHY:DATA:DCON ..... 1993
TRIGger<m>:MPHY:DATA:DMIN ..... 1993
TRIGger<m>:MPHY:DATA:DMAX ..... 1993
TRIGger<m>:MPHY:DATA:ICONdition ..... 1994
TRIGger<m>:MPHY:DATA:IMIN ..... 1994
TRIGger<m>:MPHY:DATA:IMAX ..... 1994
TRIGger<m>:MPHY:FSNumber:CONDition ..... 1995
TRIGger<m>:MPHY:FSNumber:MIN ..... 1995
TRIGger<m>:MPHY:FSNumber:MAX ..... 1995
TRIGger<m>:MPHY:CRC:CONDition ..... 1996
TRIGger<m>:MPHY:CRC:MIN ..... 1996
TRIGger<m>:MPHY:CRC:MAX ..... 1996
TRIGger<m>:MPHY:CREQ ..... 1997
TRIGger<m>:MPHY:RREQ ..... 1997
TRIGger<m>:MPHY:CREDit:CONDition ..... 1997
TRIGger<m>:MPHY:CREDit:MIN ..... 1998
TRIGger<m>:MPHY:CREDit:MAX ..... 1998
TRIGger<m>:MPHY:PBEGin:CONDition ..... 1998
TRIGger<m>:MPHY:PBEGin:MIN ..... 1999
TRIGger<m>:MPHY:PBEGin:MAX ..... 1999
TRIGger<m>:MPHY:PFID:CONDition ..... 1999
TRIGger<m>:MPHY:PFID:MIN ..... 2000
TRIGger<m>:MPHY:PFID:MAX ..... 2000
TRIGger<m>:MPHY:PRPLength ..... 2000
TRIGger<m>:MPHY:ERRor:LCMD ..... 2001
TRIGger<m>:MPHY:ERRor:LENGth ..... 2001
TRIGger<m>:MPHY:ERRor:REServed ..... 2001
TRIGger<m>:MPHY:ERRor:SYMBol ..... 2001
TRIGger<m>:MPHY:ERRor:UNKNown ..... 2002

**TRIGger<m>:MPHY:TYPE** <Type>

Selects the type of frame to be triggered on.

**Suffix:**
<m> 1..3

**Parameters:**
<Type> START | BURST | ADAPT | LCC | DLPDUSOF | DLPDUCOF | DLPDUEOF | DLPDUNAC | DLPDUAFC | PACP | TRIGUPR0 | TRIGUPR1 | TRIGUPR2 | ERRor

**START**
M-PHY or UniPro start of frame.

**BURST**
M-PHY burst frame.

**ADAPT**
M-PHY Adapt frame. This control frame is used for clock/bit synchronization.

**LCC**
M-PHY LCC frame. This control frame is used for line configuration depending on the `LCC Type`.

**DLPDUSOF | DLPDUCOF | DLPDUEOF | DLPDUNAC | DLPDUAFC**
UniPro DL PDU frame.

**PACP**
UniPro PACP frame. This control frame is used mainly for power mode change and L1.5 link management.

**TRIGUPR0 | TRIGUPR1 | TRIGUPR2**
UniPro Trigger Upper frame. This control frame is used for link startup sequence.

**ERRor**
M-PHY or UniPro error frame.

*RST: START

**Manual operation:** See "Type" on page 723

**TRIGger<m>:MPHY:LCCType** <LCCType>

Selects the type of LCC frame to be triggered on.

**Suffix:**
<m> 1..3

**Parameters:**
<LCCType> ANY | SLEEP | STALL | READCAP | READMFG | READVEND | WRITEATTR | PWMG0 | PWMG1 | PWMG2 | PWMG3 | PWMG4 | PWMG5 | PWMG6 | PWMG7 | HSG1A | HSG2A | HSG3A | HSG4A | HSG1B | HSG2B | HSG3B | HSG4B | RESERVED

**ANY**
All the available LCC commands.

**SLEEP**
Switches the power-saving state to ultra-low power. In this state, up to 90% power saving is achieved.

**STALL**
Switches the power-saving state to ultra-low power. In this state, up to 75% power saving is achieved.

**READCAP**
Recovers data about the OMC's capabilities.

**READMFG**
Retrieves manufacturing ID and vendor-specific information.

**READVEND**
Retrieves the additional four delimited bytes containing vendor-specific information.

**WRITEATTR**
Sets the configuration parameters required for lane operation.

**PWMG0 | PWMG1 | PWMG2 | PWMG3 | PWMG4 | PWMG5 | PWMG6 | PWMG7**
Switches the transmission mode to the selected low-power gear.

**HSG1A | HSG2A | HSG3A | HSG4A | HSG1B | HSG2B | HSG3B | HSG4B**
Switches the transmission mode to the selected high-speed gear.

**RESERVED**
Reserved bit command. These bits are reserved for future use.

*RST: ANY

**Manual operation:** See "LCC Type" on page 729

---

**TRIGger<m>:MPHY:LWONe:CONDition** <CondOperator>

Sets the condition for WORD 1. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<CondOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `TRIGger<m>:MPHY:LWONe:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MPHY:LWONe:MIN` and `TRIGger<m>:MPHY:LWONe:MAX`.

*RST: EQUal

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 730

---

**TRIGger<m>:MPHY:LWONe:MIN** <LCCWORD1Min>

Specifies the WORD 1 pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<LCCWORD1Min> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 730

---

**TRIGger<m>:MPHY:LWONe:MAX** <LCCWORD1Max>

Sets the end value of the WORD 1 range if `TRIGger<m>:MPHY:LWONe:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<LCCWORD1Max> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 730

---

**TRIGger<m>:MPHY:LWTWo:CONDition** <CondOperator>

Sets the condition for WORD 2. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<CondOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `TRIGger<m>:MPHY:LWTWo:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MPHY:LWTWo:MIN` and `TRIGger<m>:MPHY:LWTWo:MAX`.

*RST: EQUal

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 730

---

**TRIGger<m>:MPHY:LWTWo:MIN** <LCCWORD2Min>

Specifies the WORD 2 pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<LCCWORD2Min> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 730

---

**TRIGger<m>:MPHY:LWTWo:MAX** <LCCWORD2Max>

Sets the end value of the WORD 2 range if `TRIGger<m>:MPHY:LWTWo:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<LCCWORD2Max> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 730

---

**TRIGger<m>:MPHY:LWTHree:CONDition** <CondOperator>

Sets the condition for WORD 3. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<CondOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `TRIGger<m>:MPHY:LWTHree:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MPHY:LWTHree:MIN` and `TRIGger<m>:MPHY:LWTHree:MAX`.

*RST: EQUal

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 730

---

**TRIGger<m>:MPHY:LWTHree:MIN** <LCCWORD3Min>

Specifies the WORD 3 pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<LCCWORD3Min> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 730

---

**TRIGger<m>:MPHY:LWTHree:MAX** <LCCWORD3Max>

Sets the end value of the WORD 3 range if `TRIGger<m>:MPHY:LWTHree:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<LCCWORD3Max> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 730

---

**TRIGger<m>:MPHY:LWFour:CONDition** <CondOperator>

Sets the condition for WORD 4. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<CondOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `TRIGger<m>:MPHY:LWFour:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MPHY:LWFour:MIN` and `TRIGger<m>:MPHY:LWFour:MAX`.

*RST: EQUal

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 730

---

**TRIGger<m>:MPHY:LWFour:MIN** <LCCWORD4Min>

Specifies the WORD 4 pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<LCCWORD4Min> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 730

---

**TRIGger<m>:MPHY:LWFour:MAX** <LCCWORD4Max>

Sets the end value of the WORD 4 range if `TRIGger<m>:MPHY:LWFour:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<LCCWORD4Max> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 730

---

**TRIGger<m>:MPHY:TC** <TC>

Specifies the TC bit pattern.

**Suffix:**
<m> 1..3

**Parameters:**
<TC> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "TC" on page 730

**TRIGger<m>:MPHY:DATA:DCON** <DataOperator>

Sets the condition for the data. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:MPHY:DATA:DMIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MPHY:DATA:DMIN` and `TRIGger<m>:MPHY:DATA:DMAX`.

*RST: EQUal

**Manual operation:** See "Data" on page 731

**TRIGger<m>:MPHY:DATA:DMIN** <DataMin>

Specifies the data bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data" on page 731

**TRIGger<m>:MPHY:DATA:DMAX** <DataMax>

Sets the end value of the data range if `TRIGger<m>:MPHY:DATA:DCON` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<DataMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Data" on page 731

**TRIGger<m>:MPHY:DATA:ICONdition** <DataIdxOperator>

Sets the condition for the index. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataIdxOperator> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal | LTHan | LETHan | GTHan | GETHan**
Equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `TRIGger<m>:MPHY:DATA:IMIN` command.

**INRange = RANGe**
In range: Set the minimum and maximum value using `TRIGger<m>:MPHY:DATA:IMIN` and `TRIGger<m>:MPHY:DATA:IMAX`.

*RST: INRange

**Manual operation:** See "Index" on page 731

**TRIGger<m>:MPHY:DATA:IMIN** <DataIndexMin>

Specifies the index minimum, or sets the start value of a range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataIndexMin> Range: 1 to 0
Increment: 1
*RST: 1

**Manual operation:** See "Index" on page 731

**TRIGger<m>:MPHY:DATA:IMAX** <DataIndexMax>

Sets the end value of the index range if `TRIGger<m>:MPHY:DATA:ICONdition` is set to `INRange` or `RANge`.

**Suffix:**
<m> 1..3

**Parameters:**
<DataIndexMax> Range: 1 to 0
Increment: 1
*RST: 0

**Manual operation:** See "Index" on page 731

**TRIGger<m>:MPHY:FSNumber:CONDition** <FSNOperator>

Sets the condition for FSN. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<FSNOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:MPHY:FSNumber:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MPHY:FSNumber:MIN` and `TRIGger<m>:MPHY:FSNumber:MAX`.

*RST: EQUal

**Manual operation:** See "FSN" on page 731

**TRIGger<m>:MPHY:FSNumber:MIN** <FSNMin>

Specifies the FSN bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<FSNMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "FSN" on page 731

**TRIGger<m>:MPHY:FSNumber:MAX** <FSNMax>

Sets the end value of the FSN range if `TRIGger<m>:MPHY:FSNumber:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<FSNMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "FSN" on page 731

**TRIGger<m>:MPHY:CRC:CONDition** <CRCOperator>

Sets the condition for CRC. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<CRCOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:MPHY:CRC:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MPHY:CRC:MIN` and `TRIGger<m>:MPHY:CRC:MAX`.

*RST: EQUal

**Manual operation:** See "CRC" on page 732

---

**TRIGger<m>:MPHY:CRC:MIN** <CRCMin>

Specifies the CRC bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<CRCMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "CRC" on page 732

---

**TRIGger<m>:MPHY:CRC:MAX** <CRCMax>

Sets the end value of the CRC range if `TRIGger<m>:MPHY:CRC:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<CRCMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "CRC" on page 732

**TRIGger<m>:MPHY:CREQ** <REQ>

Specifies the CREQ bit pattern.

**Suffix:**
<m> 1..3

**Parameters:**
<REQ> ONE | ZERO | DC

**ONE**
Sets the trigger condition to binary 1.

**ZERO**
Sets the trigger condition to binary 0.

**DC**
Sets the trigger condition to any CREQ value (don't care).

*RST: DC

**Manual operation:** See "Req" on page 732

**TRIGger<m>:MPHY:RREQ** <REQ>

Specifies the RREQ bit pattern.

**Suffix:**
<m> 1..3

**Parameters:**
<REQ> ONE | ZERO | DC

**ONE**
Sets the trigger condition to binary 1.

**ZERO**
Sets the trigger condition to binary 0.

**DC**
Sets the trigger condition to any RREQ value (don't care).

*RST: DC

**Manual operation:** See "Req" on page 732

**TRIGger<m>:MPHY:CREDit:CONDition** <CreditOperator>

Sets the condition for Credit. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<CreditOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:MPHY:CREDit:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MPHY:CREDit:MIN` and `TRIGger<m>:MPHY:CREDit:MAX`.

*RST: EQUal

**Manual operation:** See "Credit" on page 732

---

**TRIGger<m>:MPHY:CREDit:MIN** <CreditMin>

Specifies the Credit bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<CreditMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Credit" on page 732

---

**TRIGger<m>:MPHY:CREDit:MAX** <CreditMax>

Sets the end value of the Credit range if `TRIGger<m>:MPHY:CREDit:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<CreditMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Credit" on page 732

---

**TRIGger<m>:MPHY:PBEGin:CONDition** <CondPACPBegin>

Sets the condition for PACP Begin. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<CondPACPBegin> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:MPHY:PBEGin:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MPHY:PBEGin:MIN` and `TRIGger<m>:MPHY:PBEGin:MAX`.

*RST: EQUal

**Manual operation:** See "PACP Begin" on page 733

---

**TRIGger<m>:MPHY:PBEGin:MIN** <PACPBEGINMin>

Specifies the PACP Begin bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<PACPBEGINMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "PACP Begin" on page 733

---

**TRIGger<m>:MPHY:PBEGin:MAX** <PACPBEGINMax>

Sets the end value of the PACP Begin range if `TRIGger<m>:MPHY:PBEGin:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<PACPBEGINMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "PACP Begin" on page 733

---

**TRIGger<m>:MPHY:PFID:CONDition** <CondPACPFunc>

Sets the condition for PACP Function ID. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<CondPACPFunc> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:MPHY:PFID:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:MPHY:PFID:MIN` and `TRIGger<m>:MPHY:PFID:MAX`.

*RST: EQUal

**Manual operation:** See "PACP Function ID" on page 733

---

**TRIGger<m>:MPHY:PFID:MIN** <PACPFunIDMin>

Specifies the PACP Function ID bit pattern, or sets the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<PACPFunIDMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "PACP Function ID" on page 733

---

**TRIGger<m>:MPHY:PFID:MAX** <PACPFunIDMax>

Sets the end value of the PACP Function ID range if `TRIGger<m>:MPHY:PFID:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<PACPFunIDMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "PACP Function ID" on page 733

---

**TRIGger<m>:MPHY:PRPLength** <Width>

Sets the width for the width trigger based upon the data rate.

**Suffix:**
<m> 1..3

**Parameters:**
<Width> Range: 100E-12 to 10000
Increment: 100E-9
*RST: 5E-9
Default unit: s

---

**TRIGger<m>:MPHY:ERRor:LCMD** <LCCError>

Enables/disables trigger on LCC error.

**Suffix:**
<m> 1..3

**Parameters:**
<LCCError> ON | OFF
*RST: ON

**Manual operation:** See "LCC Command Error" on page 734

---

**TRIGger<m>:MPHY:ERRor:LENGth** <ErrorLength>

Enables/disables trigger on length error.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorLength> ON | OFF
*RST: ON

**Manual operation:** See "Length Error" on page 734

---

**TRIGger<m>:MPHY:ERRor:REServed** <ErrorReserved>

Enables/disables trigger on reserved bit error.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorReserved> ON | OFF
*RST: ON

**Manual operation:** See "Reserved Bit Error" on page 734

---

**TRIGger<m>:MPHY:ERRor:SYMBol** <ErrorSymbol>

Enables/disables trigger on 8b10b symbol error.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorSymbol> ON | OFF
*RST: ON

**Manual operation:** See "8b10b Symbol Error" on page 734

---

**TRIGger<m>:MPHY:ERRor:UNKNown** <ErrorUnknown>

Enables/disables trigger on unidentified UniPro frame error.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorUnknown> ON | OFF
*RST: ON

**Manual operation:** See "Unknown Unipro Frame" on page 734

### 24.17.15.3 Decode results M-PHY

To show the results on the screen, use the following commands:

- `BUS<m>:RESult` on page 1638

BUS<m>:MPHY:RESult:FRAMe<n>:FTYPe? ..... 2002
BUS<m>:MPHY:RESult:FRAMe<n>:STATe? ..... 2003
BUS<m>:MPHY:RESult:FCOunt? ..... 2004
BUS<m>:MPHY:RESult:FRAMe<n>:CCOunt? ..... 2004
BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:DATA? ..... 2005
BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:STARt? ..... 2005
BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:STOP? ..... 2006
BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:STATe? ..... 2006
BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:TYPE? ..... 2006
BUS<m>:MPHY:RESult:FRAMe<n>:DATA? ..... 2007
BUS<m>:MPHY:RESult:FRAMe<n>:INFO? ..... 2007
BUS<m>:MPHY:RESult:FRAMe<n>:SSFT? ..... 2007
BUS<m>:MPHY:RESult:FRAMe<n>:STARt? ..... 2008
BUS<m>:MPHY:RESult:FRAMe<n>:STOP? ..... 2008

---

**BUS<m>:MPHY:RESult:FRAMe<n>:FTYPe?**

Returns the type of frame for the selected frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<FrameType> BURST | ADAPT | LCC | DLPDUSOF | DLPDUCOF | DLPDUEOF | DLPDUNAC | DLPDUAFC | PACP | TRIGUPR0 | TRIGUPR1 | TRIGUPR2 | SKIP | FILLER | SPACER | UNKNOWN

**BURST**
M-PHY burst frame.

**ADAPT**
M-PHY Adapt frame. This control frame is used for clock/bit synchronization.

**LCC**
M-PHY LCC frame. This control frame is used for line configuration depending on the `LCC Type`.

**DLPDUSOF | DLPDUCOF | DLPDUEOF | DLPDUNAC | DLPDUAFC**
UniPro DL PDU frame.

**PACP**
UniPro frame. This control frame is used mainly for power mode change and L1.5 link management.

**TRIGUPR0 | TRIGUPR1 | TRIGUPR2**
UniPro frame. This control frame is used for link startup sequence.

**SKIP**
UniPro frame with Skip symbols for reducing protocol payload bandwidth.

**FILLER**
M-PHY or UniPro frame with filler words to maintain transmission activity.

**SPACER**
Scrambled UniPro frame with filler words.

**UNKNOWN**
No meaningful frame can be determined.

*RST: UNKNOWN

**Usage:** Query only

---

**BUS<m>:MPHY:RESult:FRAMe<n>:STATe?**

Returns the state of the frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<State> OK | RSVDERROR | CMDERROR | SYMBOLERROR | UNKNOWNERROR | INVALID | INCOMPLETE | UNCorrelated | CRCERROR | VOOR | LENGTHERROR

**OK**
Valid frame.

**RSVDERROR**
Erroneous frame due to reserved bit error.

**CMDERROR**
Erroneous frame due to LCC error.

**SYMBOLERROR**
Erroneous frame due to 8b10b symbol error.

**UNKNOWNERROR**
Erroneous frame due to unidentified UniPro frame error.

**INVALID**
Invalid frame.

**INCOMPLETE**
The frame is not complete.

**LENGTHERROR**
Erroneous frame due to length error.

*RST: OK

**Usage:** Query only

---

**BUS<m>:MPHY:RESult:FCOunt?**

Returns the number of decoded frames.

**Suffix:**

<m> 1..4
Selects the serial bus.

**Return values:**

<Count> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:MPHY:RESult:FRAMe<n>:CCOunt?**

Returns the number of decoded cells for the selected frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<NumWords> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:DATA?**

Returns the data value for the selected cell.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the cell.

**Return values:**
<WordValue> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:STARt?**

Returns the start time for the selected cell.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the cell.

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:STOP?**

Returns the stop time for the selected cell.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the cell.

**Return values:**

<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:STATe?**

Returns the state for the selected cell.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the cell.

**Return values:**

<State> OK | RSVDERROR | CMDERROR | SYMBOLERROR | UNKNOWNERROR | INVALID | INCOMPLETE | UNCorrelated | CRCERROR | VOOR | LENGTHERROR

*RST: OK

**Usage:** Query only

**BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:TYPE?**

Returns the data type for the selected cell.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

<o> *
Selects the cell.

**Return values:**
<WordType>

**Usage:** Query only

---

**BUS<m>:MPHY:RESult:FRAMe<n>:DATA?**

Returns the data for the selected frame, corresponds to the Data column in the decode results table.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<Data>

**Usage:** Query only

---

**BUS<m>:MPHY:RESult:FRAMe<n>:INFO?**

Returns the selected frame label. This label is on top of the frame as seen in the honeycomb display, also corresponds to the Info column in the decode results table.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<Info>

**Usage:** Query only

---

**BUS<m>:MPHY:RESult:FRAMe<n>:SSFT?**

Returns the type of frame for the selected SSIC frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<SSICFrameType> TSEQ | TS1 | TS1A | TS1B | TS2 | SLF | U2IT | VDT | PCAP | PCNF | PCNR | LDM | ACK | NRDY | ERDY | STATus | STALl | FWAKe | LTM | BIAM | HRR | SSPD | PING | PNGR | DPH | DPHN | DPP | DPPA | ITP | LC | SKIP | BRST | BDAT | BERC | BCNT | IDLE | CP0 | CP1 | CP2 | CP3 | CP5or6 | UHP | ULMP | UTP | UDVN | UDEF | ERRor | CTLB | DATB | SYNC | SDS

*RST: UDEF

**Usage:** Query only

---

**BUS<m>:MPHY:RESult:FRAMe<n>:STARt?**

Returns the start time of the selected frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:MPHY:RESult:FRAMe<n>:STOP?**

Returns the end time of the selected frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

### 24.17.15.4 Search settings M-PHY

SEARch:TRIGger:MPHY:TYPE ..... 2010
SEARch:TRIGger:MPHY:LCCType ..... 2010
SEARch:TRIGger:MPHY:LWONe:CONDition ..... 2011
SEARch:TRIGger:MPHY:LWONe:MIN ..... 2012
SEARch:TRIGger:MPHY:LWONe:MAX ..... 2012
SEARch:TRIGger:MPHY:LWTWo:CONDition ..... 2012
SEARch:TRIGger:MPHY:LWTWo:MIN ..... 2013
SEARch:TRIGger:MPHY:LWTWo:MAX ..... 2013
SEARch:TRIGger:MPHY:LWTHree:CONDition ..... 2014
SEARch:TRIGger:MPHY:LWTHree:MIN ..... 2014
SEARch:TRIGger:MPHY:LWTHree:MAX ..... 2014
SEARch:TRIGger:MPHY:LWFour:CONDition ..... 2015
SEARch:TRIGger:MPHY:LWFour:MIN ..... 2015
SEARch:TRIGger:MPHY:LWFour:MAX ..... 2015
SEARch:TRIGger:MPHY:TC ..... 2016
SEARch:TRIGger:MPHY:DATA:DCONdition ..... 2016
SEARch:TRIGger:MPHY:DATA:DMIN ..... 2016
SEARch:TRIGger:MPHY:DATA:DMAX ..... 2017
SEARch:TRIGger:MPHY:DATA:ICONdition ..... 2017
SEARch:TRIGger:MPHY:DATA:IMIN ..... 2018
SEARch:TRIGger:MPHY:DATA:IMAX ..... 2018
SEARch:TRIGger:MPHY:FSNumber:CONDition ..... 2018
SEARch:TRIGger:MPHY:FSNumber:MIN ..... 2019
SEARch:TRIGger:MPHY:FSNumber:MAX ..... 2019
SEARch:TRIGger:MPHY:CRC:CONDition ..... 2019
SEARch:TRIGger:MPHY:CRC:MIN ..... 2020
SEARch:TRIGger:MPHY:CRC:MAX ..... 2020
SEARch:TRIGger:MPHY:CREQ ..... 2020
SEARch:TRIGger:MPHY:RREQ ..... 2021
SEARch:TRIGger:MPHY:CREDit:CONDition ..... 2021
SEARch:TRIGger:MPHY:CREDit:MIN ..... 2022
SEARch:TRIGger:MPHY:CREDit:MAX ..... 2022
SEARch:TRIGger:MPHY:PBEGin:CONDition ..... 2022
SEARch:TRIGger:MPHY:PBEGin:MIN ..... 2023
SEARch:TRIGger:MPHY:PBEGin:MAX ..... 2023
SEARch:TRIGger:MPHY:PFID:CONDition ..... 2023
SEARch:TRIGger:MPHY:PFID:MIN ..... 2024
SEARch:TRIGger:MPHY:PFID:MAX ..... 2024
SEARch:TRIGger:MPHY:ERRor:REServed ..... 2024
SEARch:TRIGger:MPHY:ERRor:LCMD ..... 2025
SEARch:TRIGger:MPHY:ERRor:SYMBol ..... 2025
SEARch:TRIGger:MPHY:ERRor:UNKNown ..... 2025
SEARch:TRIGger:MPHY:ERRor:LENGth ..... 2025

**SEARch:TRIGger:MPHY:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:MPHY:TYPE?** <SearchName>

Selects the type of frame to be searched for.

**Parameters:**

<Type> START | BURST | ADAPT | LCC | DLPDUSOF | DLPDUCOF | DLPDUEOF | DLPDUNAC | DLPDUAFC | PACP | TRIGUPR0 | TRIGUPR1 | TRIGUPR2 | ERRor

**START**
M-PHY or UniPro start of frame.

**BURST**
M-PHY burst frame.

**ADAPT**
M-PHY Adapt frame. This control frame is used for clock/bit synchronization.

**LCC**
M-PHY LCC frame. This control frame is used for line configuration depending on the `LCC Type`.

**DLPDUSOF | DLPDUCOF | DLPDUEOF | DLPDUNAC | DLPDUAFC**
UniPro DL PDU frame.

**PACP**
UniPro PACP frame. This control frame is used mainly for power mode change and L1.5 link management.

**TRIGUPR0 | TRIGUPR1 | TRIGUPR2**
UniPro Trigger Upper frame. This control frame is used for link startup sequence.

**ERRor**
M-PHY or UniPro error frame.

*RST: START

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Type" on page 739

**SEARch:TRIGger:MPHY:LCCType** <SearchName>,<LCCType>
**SEARch:TRIGger:MPHY:LCCType?** <SearchName>

Selects the type of LCC frame to be searched for.

**Parameters:**

<LCCType> ANY | SLEEP | STALL | READCAP | READMFG | READVEND | WRITEATTR | PWMG0 | PWMG1 | PWMG2 | PWMG3 | PWMG4 | PWMG5 | PWMG6 | PWMG7 | HSG1A | HSG2A | HSG3A | HSG4A | HSG1B | HSG2B | HSG3B | HSG4B | RESERVED

**ANY**
All the available LCC commands.

**SLEEP**
Switches the power-saving state to ultra-low power. In this state, up to 90% power saving is achieved.

**STALL**
Switches the power-saving state to ultra-low power. In this state, up to 75% power saving is achieved.

**READCAP**
Recovers data about the OMC's capabilities.

**READMFG**
Retrieves manufacturing ID and vendor-specific information.

**READVEND**
Retrieves the additional four delimited bytes containing vendor-specific information.

**WRITEATTR**
Sets the configuration parameters required for lane operation.

**PWMG0 | PWMG1 | PWMG2 | PWMG3 | PWMG4 | PWMG5 | PWMG6 | PWMG7**
Switches the transmission mode to the selected low-power gear.

**HSG1A | HSG2A | HSG3A | HSG4A | HSG1B | HSG2B | HSG3B | HSG4B**
Switches the transmission mode to the selected high-speed gear.

**RESERVED**
Reserved bit command. These bits are reserved for future use.

*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "LCC Type" on page 741

---

**SEARch:TRIGger:MPHY:LWONe:CONDition** <SearchName>,<CondOperator>
**SEARch:TRIGger:MPHY:LWONe:CONDition?** <SearchName>

Sets the condition for WORD 1. You can define an exact value or a value range.

**Parameters:**
<CondOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `SEARch:TRIGger:MPHY:LWONe:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:MPHY:LWONe:MIN` and `SEARch:TRIGger:MPHY:LWONe:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 741

---

**SEARch:TRIGger:MPHY:LWONe:MIN** <SearchName>,<LCCWORD1Min>
**SEARch:TRIGger:MPHY:LWONe:MIN?** <SearchName>

Specifies the WORD 1 pattern, or sets the the start value of a pattern range.

**Parameters:**
<LCCWORD1Min> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 741

---

**SEARch:TRIGger:MPHY:LWONe:MAX** <SearchName>,<LCCWORD1Max>
**SEARch:TRIGger:MPHY:LWONe:MAX?** <SearchName>

Sets the end value of the WORD 1 range if `SEARch:TRIGger:MPHY:LWONe:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<LCCWORD1Max> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 741

---

**SEARch:TRIGger:MPHY:LWTWo:CONDition** <SearchName>,<CondOperator>
**SEARch:TRIGger:MPHY:LWTWo:CONDition?** <SearchName>

Sets the condition for WORD 2. You can define an exact value or a value range.

**Parameters:**
<CondOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `SEARch:TRIGger:MPHY:LWTWo:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:MPHY:LWTWo:MIN` and `SEARch:TRIGger:MPHY:LWTWo:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 741

---

**SEARch:TRIGger:MPHY:LWTWo:MIN** <SearchName>,<LCCWORD2Min>
**SEARch:TRIGger:MPHY:LWTWo:MIN?** <SearchName>

Specifies the WORD 2 pattern, or sets the the start value of a pattern range.

**Parameters:**
<LCCWORD2Min> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 741

---

**SEARch:TRIGger:MPHY:LWTWo:MAX** <SearchName>,<LCCWORD2Max>
**SEARch:TRIGger:MPHY:LWTWo:MAX?** <SearchName>

Sets the end value of the WORD 2 range if `SEARch:TRIGger:MPHY:LWTWo:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<LCCWORD2Max> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 741

**SEARch:TRIGger:MPHY:LWTHree:CONDition** <SearchName>,<CondOperator>
**SEARch:TRIGger:MPHY:LWTHree:CONDition?** <SearchName>

Sets the condition for WORD 3. You can define an exact value or a value range.

**Parameters:**

<CondOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `SEARch:TRIGger:MPHY:LWTHree:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:MPHY:LWTHree:MIN` and `SEARch:TRIGger:MPHY:LWTHree:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 741

---

**SEARch:TRIGger:MPHY:LWTHree:MIN** <SearchName>,<LCCWORD3Min>
**SEARch:TRIGger:MPHY:LWTHree:MIN?** <SearchName>

Specifies the WORD 3 pattern, or sets the the start value of a pattern range.

**Parameters:**

<LCCWORD3Min> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 741

---

**SEARch:TRIGger:MPHY:LWTHree:MAX** <SearchName>,<LCCWORD3Max>
**SEARch:TRIGger:MPHY:LWTHree:MAX?** <SearchName>

Sets the end value of the WORD 3 range if `SEARch:TRIGger:MPHY:LWTHree:CONDition` is set to `INRange` or `OORange`.

**Parameters:**

<LCCWORD3Max> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 741

---

**SEARch:TRIGger:MPHY:LWFour:CONDition** <SearchName>,<CondOperator>
**SEARch:TRIGger:MPHY:LWFour:CONDition?** <SearchName>

Sets the condition for WORD 4. You can define an exact value or a value range.

**Parameters:**

<CondOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `SEARch:TRIGger:MPHY:LWFour:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:MPHY:LWFour:MIN` and `SEARch:TRIGger:MPHY:LWFour:MAX`.

*RST: EQUal

**Parameters for setting and query:**

<SearchName>

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 741

---

**SEARch:TRIGger:MPHY:LWFour:MIN** <SearchName>,<LCCWORD4Min>
**SEARch:TRIGger:MPHY:LWFour:MIN?** <SearchName>

Specifies the WORD 4 pattern, or sets the the start value of a pattern range.

**Parameters:**

<LCCWORD4Min> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**

<SearchName>

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 741

---

**SEARch:TRIGger:MPHY:LWFour:MAX** <SearchName>,<LCCWORD4Max>
**SEARch:TRIGger:MPHY:LWFour:MAX?** <SearchName>

Sets the end value of the WORD 4 range if `SEARch:TRIGger:MPHY:LWFour:CONDition` is set to `INRange` or `OORange`.

**Parameters:**

<LCCWORD4Max> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "WORD 1/WORD 2/WORD 3/WORD 4" on page 741

---

**SEARch:TRIGger:MPHY:TC** <SearchName>,<TC>
**SEARch:TRIGger:MPHY:TC?** <SearchName>

Specifies the TC bit pattern.

**Parameters:**
<TC> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "TC" on page 742

---

**SEARch:TRIGger:MPHY:DATA:DCONdition** <SearchName>,<DataOperator>
**SEARch:TRIGger:MPHY:DATA:DCONdition?** <SearchName>

Sets the condition for the data. You can define an exact value or a value range.

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:MPHY:DATA:DMIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:MPHY:DATA:DMIN` and `SEARch:TRIGger:MPHY:DATA:DMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data" on page 742

---

**SEARch:TRIGger:MPHY:DATA:DMIN** <SearchName>,<DataMin>
**SEARch:TRIGger:MPHY:DATA:DMIN?** <SearchName>

Specifies the data bit pattern, or sets the the start value of a pattern range.

**Parameters:**
<DataMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data" on page 742

---

**SEARch:TRIGger:MPHY:DATA:DMAX** <SearchName>,<DataMax>
**SEARch:TRIGger:MPHY:DATA:DMAX?** <SearchName>

Sets the end value of the data range if `SEARch:TRIGger:MPHY:DATA:DCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<DataMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data" on page 742

---

**SEARch:TRIGger:MPHY:DATA:ICONdition** <SearchName>,<DataIdxOperator>
**SEARch:TRIGger:MPHY:DATA:ICONdition?** <SearchName>

Sets the condition for the index. You can define an exact value or a value range.

**Parameters:**
<DataIdxOperator> EQUal | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**EQUal | LTHan | LETHan | GTHan | GETHan**
Equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one pattern to be set with the corresponding `SEARch:TRIGger:MPHY:DATA:IMIN` command.

**INRange = RANGe**
In range: Set the minimum and maximum value using `SEARch:TRIGger:MPHY:DATA:IMIN` and `SEARch:TRIGger:MPHY:DATA:IMAX`.

*RST: INRange

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Index" on page 743

**SEARch:TRIGger:MPHY:DATA:IMIN** <SearchName>,<DataIndexMin>
**SEARch:TRIGger:MPHY:DATA:IMIN?** <SearchName>

Specifies the index minimum, or sets the start value of a range.

**Parameters:**
<DataIndexMin> Range: 1 to 0
Increment: 1
*RST: 1

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Index" on page 743

**SEARch:TRIGger:MPHY:DATA:IMAX** <SearchName>,<DataIndexMax>
**SEARch:TRIGger:MPHY:DATA:IMAX?** <SearchName>

Sets the end value of the index range if `SEARch:TRIGger:MPHY:DATA:ICONdition` is set to `INRange` or `RANge`.

**Parameters:**
<DataIndexMax> Range: 1 to 0
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Index" on page 743

**SEARch:TRIGger:MPHY:FSNumber:CONDition** <SearchName>,<FSNOperator>
**SEARch:TRIGger:MPHY:FSNumber:CONDition?** <SearchName>

Sets the condition for FSN. You can define an exact value or a value range.

**Parameters:**
<FSNOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:MPHY:FSNumber:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:MPHY:FSNumber:MIN` and `SEARch:TRIGger:MPHY:FSNumber:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "FSN" on page 743

---

**SEARch:TRIGger:MPHY:FSNumber:MIN** <SearchName>,<FSNMin>
**SEARch:TRIGger:MPHY:FSNumber:MIN?** <SearchName>

Specifies the FSN bit pattern, or sets the start value of a pattern range.

**Parameters:**
<FSNMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "FSN" on page 743

---

**SEARch:TRIGger:MPHY:FSNumber:MAX** <SearchName>,<FSNMax>
**SEARch:TRIGger:MPHY:FSNumber:MAX?** <SearchName>

Sets the end value of the FSN range if `SEARch:TRIGger:MPHY:FSNumber:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<FSNMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "FSN" on page 743

---

**SEARch:TRIGger:MPHY:CRC:CONDition** <SearchName>,<CRCOperator>
**SEARch:TRIGger:MPHY:CRC:CONDition?** <SearchName>

Sets the condition for CRC. You can define an exact value or a value range.

**Parameters:**
<CRCOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:MPHY:CRC:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:MPHY:CRC:MIN` and `SEARch:TRIGger:MPHY:CRC:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "CRC" on page 743

---

**SEARch:TRIGger:MPHY:CRC:MIN** <SearchName>,<CRCMin>
**SEARch:TRIGger:MPHY:CRC:MIN?** <SearchName>

Specifies the CRC bit pattern, or sets the start value of a pattern range.

**Parameters:**
<CRCMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "CRC" on page 743

---

**SEARch:TRIGger:MPHY:CRC:MAX** <SearchName>,<CRCMax>
**SEARch:TRIGger:MPHY:CRC:MAX?** <SearchName>

Sets the end value of the CRC range if `SEARch:TRIGger:MPHY:CRC:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<CRCMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "CRC" on page 743

---

**SEARch:TRIGger:MPHY:CREQ** <SearchName>,<REQ>
**SEARch:TRIGger:MPHY:CREQ?** <SearchName>

Specifies the CREQ bit pattern.

**Parameters:**
<REQ> ONE | ZERO | DC

**ONE**
Sets the trigger condition to binary 1.

**ZERO**
Sets the trigger condition to binary 0.

**DC**
Sets the trigger condition to any CREQ value (don't care).

*RST: DC

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Req" on page 744

---

**SEARch:TRIGger:MPHY:RREQ** <SearchName>,<REQ>
**SEARch:TRIGger:MPHY:RREQ?** <SearchName>

Specifies the RREQ bit pattern.

**Parameters:**
<REQ> ONE | ZERO | DC

**ONE**
Sets the trigger condition to binary 1.

**ZERO**
Sets the trigger condition to binary 0.

**DC**
Sets the trigger condition to any RREQ value (don't care).

*RST: DC

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Req" on page 744

---

**SEARch:TRIGger:MPHY:CREDit:CONDition** <SearchName>,<CreditOperator>
**SEARch:TRIGger:MPHY:CREDit:CONDition?** <SearchName>

Sets the condition for Credit. You can define an exact value or a value range.

**Parameters:**
<CreditOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:MPHY:CREDit:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:MPHY:CREDit:MIN` and `SEARch:TRIGger:MPHY:CREDit:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Credit" on page 744

**SEARch:TRIGger:MPHY:CREDit:MIN** <SearchName>,<CreditMin>
**SEARch:TRIGger:MPHY:CREDit:MIN?** <SearchName>

Specifies the Credit bit pattern, or sets the start value of a pattern range.

**Parameters:**
<CreditMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Credit" on page 744

**SEARch:TRIGger:MPHY:CREDit:MAX** <SearchName>,<CreditMax>
**SEARch:TRIGger:MPHY:CREDit:MAX?** <SearchName>

Sets the end value of the Credit range if `SEARch:TRIGger:MPHY:CREDit:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<CreditMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Credit" on page 744

**SEARch:TRIGger:MPHY:PBEGin:CONDition** <SearchName>,<CondPACPBegin>
**SEARch:TRIGger:MPHY:PBEGin:CONDition?** <SearchName>

Sets the condition for PACP Begin. You can define an exact value or a value range.

**Parameters:**
<CondPACPBegin> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:MPHY:PBEGin:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:MPHY:PBEGin:MIN` and `SEARch:TRIGger:MPHY:PBEGin:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "PACP Begin" on page 745

---

**SEARch:TRIGger:MPHY:PBEGin:MIN** <SearchName>,<PACPBEGINMin>
**SEARch:TRIGger:MPHY:PBEGin:MIN?** <SearchName>

Specifies the PACP Begin bit pattern, or sets the start value of a pattern range.

**Parameters:**
<PACPBEGINMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "PACP Begin" on page 745

---

**SEARch:TRIGger:MPHY:PBEGin:MAX** <SearchName>,<PACPBEGINMax>
**SEARch:TRIGger:MPHY:PBEGin:MAX?** <SearchName>

Sets the end value of the PACP Begin range if `SEARch:TRIGger:MPHY:PBEGin:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<PACPBEGINMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "PACP Begin" on page 745

---

**SEARch:TRIGger:MPHY:PFID:CONDition** <SearchName>,<CondPACPFunc>
**SEARch:TRIGger:MPHY:PFID:CONDition?** <SearchName>

Sets the condition for PACP Function ID. You can define an exact value or a value range.

**Parameters:**
<CondPACPFunc> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:MPHY:PFID:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:MPHY:PFID:MIN` and `SEARch:TRIGger:MPHY:PFID:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "PACP Function ID" on page 745

---

**SEARch:TRIGger:MPHY:PFID:MIN** <SearchName>,<PACPFunIDMin>
**SEARch:TRIGger:MPHY:PFID:MIN?** <SearchName>

Specifies the PACP Function ID bit pattern, or sets the start value of a pattern range.

**Parameters:**
<PACPFunIDMin> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "PACP Function ID" on page 745

---

**SEARch:TRIGger:MPHY:PFID:MAX** <SearchName>,<PACPFunIDMax>
**SEARch:TRIGger:MPHY:PFID:MAX?** <SearchName>

Sets the end value of the PACP Function ID range if `SEARch:TRIGger:MPHY:PFID:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<PACPFunIDMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "PACP Function ID" on page 745

---

**SEARch:TRIGger:MPHY:ERRor:REServed** <SearchName>,<ErrorReserved>
**SEARch:TRIGger:MPHY:ERRor:REServed?** <SearchName>

Enables/disables search for reserved bit error.

**Parameters:**
<ErrorReserved> ON | OFF

*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Reserved Bit Error" on page 745

**SEARch:TRIGger:MPHY:ERRor:LCMD** <SearchName>,<LCCError>
**SEARch:TRIGger:MPHY:ERRor:LCMD?** <SearchName>

Enables/disables search for LCC error.

**Parameters:**
<LCCError> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "LCC Command Error" on page 746

**SEARch:TRIGger:MPHY:ERRor:SYMBol** <SearchName>,<ErrorSymbol>
**SEARch:TRIGger:MPHY:ERRor:SYMBol?** <SearchName>

Enables/disables search for 8b10b symbol error.

**Parameters:**
<ErrorSymbol> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "8b10b Symbol Error" on page 746

**SEARch:TRIGger:MPHY:ERRor:UNKNown** <SearchName>,<ErrorUnknown>
**SEARch:TRIGger:MPHY:ERRor:UNKNown?** <SearchName>

Enables/disables search for unidentified UniPro frame error.

**Parameters:**
<ErrorUnknown> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Unknown Unipro Frame" on page 746

**SEARch:TRIGger:MPHY:ERRor:LENGth** <SearchName>,<ErrorLength>
**SEARch:TRIGger:MPHY:ERRor:LENGth?** <SearchName>

Enables/disables search for length error.

**Parameters:**
<ErrorLength> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Length Error" on page 746

### 24.17.15.5 Search results M-PHY

In all `SEARch:RESult:MPHY:FRAMe<m>` commands, the suffix <m> selects the frame number and suffix <n> selects the cell number in the list of search results.

SEARch:RESult:MPHY:FCOunt?....2026
SEARch:RESult:MPHY:FRAMe<m>:CCOunt?....2026
SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:DATA?....2027
SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:TYPE?....2027
SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:STARt?....2027
SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:STOP?....2028
SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:STATe?....2028
SEARch:RESult:MPHY:FRAMe<m>:DATA?....2028
SEARch:RESult:MPHY:FRAMe<m>:FTYPe?....2029
SEARch:RESult:MPHY:FRAMe<m>:STATe?....2029
SEARch:RESult:MPHY:FRAMe<m>:STARt?....2030
SEARch:RESult:MPHY:FRAMe<m>:STOP?....2031

---

**SEARch:RESult:MPHY:FCOunt?** <SearchName>

Returns the number of frames within the search result.

**Query parameters:**
<SearchName>

**Return values:**

| | | |
|---|---|---|
| <Count> | Range: | 0 to 100000 |
| | Increment: | 1 |
| | *RST: | 0 |

**Usage:** Query only

---

**SEARch:RESult:MPHY:FRAMe<m>:CCOunt?** <SearchName>

Returns the number of cells within the search result for the selected frame.

**Suffix:**

| | |
|---|---|
| <m> | * |

**Query parameters:**
<SearchName>

**Return values:**

| | | |
|---|---|---|
| <NumWords> | Range: | 0 to 4294967295 |
| | Increment: | 1 |
| | *RST: | 0 |

**Usage:** Query only

**SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:DATA?** <SearchName>

Returns the data value within the search result for the selected cell.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<WordValue> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:TYPE?** <SearchName>

Returns the data type within the search result for the selected cell.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<WordType>

**Usage:** Query only

**SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:STARt?** <SearchName>

Returns the start time of the cell with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:STOP?** <SearchName>

Returns the stop time of the cell with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:STATe?** <SearchName>

Returns the state type within the search result for the selected cell.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | RSVDERROR | CMDERROR | SYMBOLERROR | UNKNOWNERROR | INVALID | INCOMPLETE | UNCorrelated | CRCERROR | VOOR | LENGTHERROR
*RST: OK

**Usage:** Query only

**SEARch:RESult:MPHY:FRAMe<m>:DATA?** <SearchName>

Returns the data for the selected frame within the search result, corresponds to the Data column in the search results table.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

**SEARch:RESult:MPHY:FRAMe<m>:FTYPe?** <SearchName>

Returns the type of frame for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameType> BURST | ADAPT | LCC | DLPDUSOF | DLPDUCOF | DLPDUEOF | DLPDUNAC | DLPDUAFC | PACP | TRIGUPR0 | TRIGUPR1 | TRIGUPR2 | SKIP | FILLER | SPACER | UNKNOWN

**BURST**
M-PHY burst frame.

**ADAPT**
M-PHY Adapt frame. This control frame is used for clock/bit synchronization.

**LCC**
M-PHY LCC frame. This control frame is used for line configuration depending on the `LCC Type`.

**DLPDUSOF | DLPDUCOF | DLPDUEOF | DLPDUNAC | DLPDUAFC**
UniPro DL PDU frame.

**PACP**
UniPro frame. This control frame is used mainly for power mode change and L1.5 link management.

**TRIGUPR0 | TRIGUPR1 | TRIGUPR2**
UniPro frame. This control frame is used for link startup sequence.

**SKIP**
UniPro frame with Skip symbols for reducing protocol payload bandwidth.

**FILLER**
M-PHY or UniPro frame with filler words to maintain transmission activity.

**SPACER**
Scrambled UniPro frame with filler words.

**UNKNOWN**
No meaningful frame can be determined.

*RST: UNKNOWN

**Usage:** Query only

**SEARch:RESult:MPHY:FRAMe<m>:STATe?** <SearchName>

Returns the state of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | RSVDERROR | CMDERROR | SYMBOLERROR | UNKNOWNERROR | INVALID | INCOMPLETE | UNCorrelated | CRCERROR | VOOR | LENGTHERROR

**OK**
Valid frame.

**RSVDERROR**
Erroneous frame due to reserved bit error.

**CMDERROR**
Erroneous frame due to LCC error.

**LENGTHERROR**
Erroneous frame due to length error.

**SYMBOLERROR**
Erroneous frame due to 8b10b symbol error.

**UNKNOWNERROR**
Erroneous frame due to unidentified UniPro frame error.

**INVALID**
Invalid frame.

**INCOMPLETE**
The frame is not complete.

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:MPHY:FRAMe<m>:STARt?** <SearchName>

Returns the start time of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:MPHY:FRAMe<m>:STOP?** <SearchName>

Returns the end time for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

### 24.17.15.6 Trigger USB SSIC (Option R&S RTP-K64)

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

There are two commands for each parameter, that you can use for defining the SSIC trigger settings.

For example, to set the *Frame type =ACK > Field =Route String >Data* value you can use one of the following commands:

- `TRIGger:MPHY:SSIC:FRAMe13:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:MPHY:SSIC:DMIN "ACK", "Route String", 01100`
  Defines the parameter by using the frame and field name.

TRIGger<m>:MPHY:SSIC:ERENable....2032
TRIGger<m>:MPHY:SSIC:ERRor<n>:ENABle....2032
TRIGger<m>:MPHY:SSIC:FRENable....2032
TRIGger<m>:MPHY:SSIC:FRAMe<n>:ENABle....2032
TRIGger<m>:MPHY:SSIC:FIENable....2033
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:ENABle....2033
TRIGger<m>:MPHY:SSIC:BIT....2033
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:BIT....2033
TRIGger<m>:MPHY:SSIC:DMAX....2033
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DMAX....2033
TRIGger<m>:MPHY:SSIC:DMIN....2034
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DMIN....2034

TRIGger<m>:MPHY:SSIC:DOPerator ..... 2034
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DOPerator ..... 2034
TRIGger<m>:MPHY:SSIC:IMAX ..... 2035
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IMAX ..... 2035
TRIGger<m>:MPHY:SSIC:IMIN ..... 2035
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IMIN ..... 2035
TRIGger<m>:MPHY:SSIC:IOPerator ..... 2036
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IOPerator ..... 2036

**TRIGger<m>:MPHY:SSIC:ERENable** <ErrorName>, <Enabler>
**TRIGger<m>:MPHY:SSIC:ERRor<n>:ENABle** <Enable>

Enables or disables the trigger on an error condition.

**Suffix:**
<m> 1..3

<n> *

**Parameters:**
<Enable> ON | OFF

*RST: ON

**Example:** Trigger on CRC and length errors:
```
TRIGger:MPHY:SSIC:ERRor1:ENABle ON
```
Enables CRC Error.
```
TRIGger:MPHY:SSIC:ERENable "Length Error",ON
```
Enables Length Error.

**Manual operation:** See "Error types, Enable" on page 735

**TRIGger<m>:MPHY:SSIC:FRENable** <Frame>, <Enabler>
**TRIGger<m>:MPHY:SSIC:FRAMe<n>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the field number within the frame.

**Parameters:**
<Enable> ON | OFF

*RST: OFF

**Example:**
```
TRIGger:MPHY:SSIC:FRAMe1:ENABle ON
```
Enables the checking condition for the TSEQ frame.
```
TRIGger:MPHY:SSIC:FRENable "TS1" ON
```
Enables the checking condition for the TS1 frame.

**TRIGger<m>:MPHY:SSIC:FIENable** <Frame>, <Field>, <Enabler>
**TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the field number within the frame.

<o> *
Specifies the field number within the frame.

**Parameters:**
<CondEnabler> ON | OFF
*RST: OFF

**Example:** `TRIGger:MPHY:SSIC:FRAMe1:FLD1:ENABle ON`
Enables the checking condition for the Symbol field of the TSEQ frame.
`TRIGger:MPHY:SSIC:FRENable "TS1", "Loop" ON`
Enables the checking condition for the Loop field of the TS1 frame.

**Manual operation:** See "AND / Field name / Condition" on page 736

**TRIGger<m>:MPHY:SSIC:BIT** <Frame>, <Field>, <Bit>
**TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the frame number.

<o> *
Specifies the field number within the frame.

**Parameters:**
<BitState> ONE | ZERO | DC
*RST: DC

**Manual operation:** See "BitState" on page 737

**TRIGger<m>:MPHY:SSIC:DMAX** <Frame>, <Field>, <Data>
**TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DMAX** <DataMax>

Sets the end value of a data pattern range if `TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DOPerator` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.
<n> *
Specifies the frame number.
<o> *
Specifies the field number within the frame.

**Parameters:**
<DataMax>

**Manual operation:** See "Data" on page 736

---

**TRIGger<m>:MPHY:SSIC:DMIN** <Frame>, <Field>, <Data>
**TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DMIN** <DataMin>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.
<n> *
Specifies the frame number.
<o> *
Specifies the field number within the frame.

**Parameters:**
<DataMin>

**Manual operation:** See "Data" on page 736

---

**TRIGger<m>:MPHY:SSIC:DOPerator** <Frame>, <Field>, <Operator>
**TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DOPerator** <DataOperator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.
<n> *
Specifies the field number within the frame.
<o> *

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one endpoint value to be set using `TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of endpoint values to be set using `TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DMIN` and `TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DMAX`.

*RST: EQUal

**Manual operation:** See "Data" on page 736

---

**TRIGger<m>:MPHY:SSIC:IMAX** <Frame>, <Field>, <Data>
**TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IMAX** <IndexMax>

Sets the end value of an index range if `TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IOPerator` is set to `INRange` or `RANGe`.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the field number within the frame.

<o> *

**Parameters:**
<IndexMax> Specifies the end value for the index range within the field. The index range, increment and *RST values depend on the field type.

**Manual operation:** See "Index" on page 737

---

**TRIGger<m>:MPHY:SSIC:IMIN** <Frame>, <Field>, <Data>
**TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IMIN** <IndexMin>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<m> 1..3

<n> *
Specifies the field number within the frame.

<o> *

**Parameters:**

<IndexMin> Specifies the index value or index start value within the field. The index range, increment and *RST values depend on the field type.

Range: 1 to 65534
Increment: 1
*RST: 1

**Manual operation:** See "Index" on page 737

---

**TRIGger<m>:MPHY:SSIC:IOPerator** <Frame>, <Field>, <Operator>
**TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IOPerator** <IndexOperator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**

<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the field number within the frame.

<o> *

**Parameters:**

<IndexOperator> EQUal | INRange | RANGe

**EQUal**
This condition requires one endpoint value to be set using `TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IMIN`.

**INRange | RANGe**
This condition requires a range of endpoint values to be set using `TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IMIN` and `TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IMAX`.

*RST: INRange

**Manual operation:** See "Index" on page 737

### 24.17.15.7 Search settings USB SSIC (Option R&S RTP-K64)

There are two commands for each parameter, that you can use for defining the SSIC search settings.

For example, to set the *Frame type =ACK > Field =Route String >Data* value you can use one of the following commands:

- `SEARch:TRIGger:MPHY:SSIC:FRAMe13:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SEARch:TRIGger:MPHY:SSIC:DMIN "ACK", "Route String", 01100`
  Defines the parameter by using the frame and field name.

**SEARch:TRIGger:MPHY:SSIC:ERENable** <SearchName>,<ErrorName>, <Enabler>
**SEARch:TRIGger:MPHY:SSIC:ERRor<m>:ENABle** <SearchName>,<Enable>
**SEARch:TRIGger:MPHY:SSIC:ERRor<m>:ENABle?** <SearchName>

Enables or disables the searching for an error condition.

**Suffix:**
<m> *

**Parameters:**
<Enable> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Example:** Searches for CRC and length errors:
```
SEARch:TRIGger:MPHY:SSIC:ERRor1:ENABle ON
```
Enables searching for CRC Error.
```
SEARch:TRIGger:MPHY:SSIC:ERENable "Length Error",ON
```
Enables searching for Length Error.

**Manual operation:** See "Search criteria" on page 747

**SEARch:TRIGger:MPHY:SSIC:FRENable** <SearchName>,<Frame>, <Enabler>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:ENABle** <SearchName>,<Enable>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:ENABle?** <SearchName>

Defines the frame type to be searched for.

**Suffix:**
<m> *
Specifies the frame number.

**Parameters:**
<Enable> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Example:**
```
SEARch:TRIGger:MPHY:SSIC:FRAMe1:ENABle ON
```
Enables searching for the TSEQ frame.
```
SEARch:TRIGger:MPHY:SSIC:FRENable "TS1" ON
```
Enables searching for the TS1 frame.

**Manual operation:** See "Search criteria" on page 747

**SEARch:TRIGger:MPHY:SSIC:FIENable** <SearchName>,<Frame>, <Field>, <Enabler>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:ENABle** <SearchName>,<CondEnabler>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:ENABle?** <SearchName>

Enables or disables the checking condition for searching a specific data pattern in the selected field of the selected frame.

**Suffix:**
<m> *
Specifies the frame number.

<n> *
Specifies the field number within the frame.

**Parameters:**
<CondEnabler> ON | OFF

**ON**
Checking condition enabled

**OFF**
Checking condition disabled

*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Example:** `SEARch:TRIGger:MPHY:SSIC:FRAMe1:FLD1:ENABle ON`
Enables searching for the Symbol field of the TSEQ frame.
`SEARch:TRIGger:MPHY:SSIC:FRENable "TS1", "Loop" ON`
Enables searching for the Loop field of the TS1 frame.

**Manual operation:** See "Search criteria" on page 747

**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:ENABle** <SearchName>,<CondEnabler>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:ENABle?** <SearchName>

Enables or disables the checking condition for searching a specific data pattern in the selected field of the selected frame.

**Suffix:**
<m> *
Specifies the frame number.

<n> *
Specifies the field number within the frame.

**Parameters:**
<CondEnabler> ON | OFF

**ON**
Checking condition enabled

**OFF**
Checking condition disabled

*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Example:** `SEARch:TRIGger:MPHY:SSIC:FRAMe1:FLD1:ENABle ON`
Enables searching for the Symbol field of the TSEQ frame.
`SEARch:TRIGger:MPHY:SSIC:FRENable "TS1", "Loop" ON`
Enables searching for the Loop field of the TS1 frame.

**Manual operation:** See "Search criteria" on page 747

---

**SEARch:TRIGger:MPHY:SSIC:BIT** <SearchName>,<Frame>, <Field>, <Bit>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:BIT** <SearchName>,<BitState>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:BIT?** <SearchName>

Sets the bit state of a field to be searched that only consists of one bit.

**Suffix:**
<m> *
Specifies the frame number.

<n> *
Specifies the field number within the frame.

**Parameters:**
<BitState> ONE | ZERO | DC

**ONE**
1

**ZERO**
0

**DC**
"Don't care" (DC) = X

*RST: DC

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search criteria" on page 747

---

**SEARch:TRIGger:MPHY:SSIC:DMAX** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DMAX**
<SearchName>,<DataMax>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DMAX?** <SearchName>

Sets the end value of a data pattern range if `SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DOPerator` is set to `INRange` or `OORange`.

**Suffix:**
<m> *
Specifies the frame number.

<n> *
Specifies the field number within the frame.

**Parameters:**
<DataMax> Specifies the name of the generic trigger setting frame.

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search criteria" on page 747

---

**SEARch:TRIGger:MPHY:SSIC:DMIN** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DMIN**
<SearchName>,<DataMin>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DMIN?** <SearchName>

Specifies the data pattern to be searched, or sets the start value of a data pattern range to be searched.

**Suffix:**
<m> *
Specifies the frame number.

<n> *
Specifies the field number within the frame.

**Parameters:**
<DataMin> Specifies the name of the generic trigger setting frame.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search criteria" on page 747

---

**SEARch:TRIGger:MPHY:SSIC:DOPerator** <SearchName>,<Frame>, <Field>,
<Operator>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DOPerator**
<SearchName>,<DataOperator>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DOPerator?** <SearchName>

Sets the operator to set a specific data pattern to be searched in the selected field of the selected frame.

**Suffix:**
<m> *
Specifies the frame number.

<n> *
Specifies the field number within the frame.

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one endpoint value to be set using `SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of endpoint values to be set using `SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DMIN` and `SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search criteria" on page 747

---

**SEARch:TRIGger:MPHY:SSIC:FRENable** <SearchName>,<Frame>, <Enabler>
**Setting parameters:**
<SearchName>

<Frame>

<Enabler> ON | OFF

**Usage:** Setting only

**Manual operation:** See "Search criteria" on page 747

---

**SEARch:TRIGger:MPHY:SSIC:IMAX** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IMAX**
<SearchName>,<IndexMax>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IMAX?** <SearchName>

Sets the end value of an index range if `SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IOPerator` is set to `INRange` or `RANGe`.

**Suffix:**
<m> *
Specifies the frame number.

<n> *
Specifies the field number within the frame.

**Parameters:**
<IndexMax> Specifies the end value for the index range within the field.
The index range, increment and *RST values depend on the field type.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search criteria" on page 747

---

**SEARch:TRIGger:MPHY:SSIC:IMIN** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IMIN**
<SearchName>,<IndexMin>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IMIN?** <SearchName>

Specifies the index at which the data is to be searched, or sets the start value of an index range in which the data is to be searched.

**Suffix:**
<m> *
Specifies the field number within the frame.

<n> *

**Parameters:**
<IndexMin> Specifies the index value or the start value of an index range within the field.
The index range, increment and *RST values depend on the field type.

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search criteria" on page 747

---

**SEARch:TRIGger:MPHY:SSIC:IOPerator** <SearchName>,<Frame>, <Field>,
<Operator>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IOPerator**
<SearchName>,<IndexOperator>
**SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IOPerator?** <SearchName>

Sets the operator for the index for searching in the selected field of the selected frame.

**Suffix:**
<m> *
Specifies the field number within the frame.

<n> *

**Parameters:**
<IndexOperator> EQUal | INRange | RANGe

**EQUal**
This condition requires one endpoint value to be set using `SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IMIN`.

**INRange | RANGe**
This condition requires a range of endpoint values to be set using `SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IMIN` and `SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IMAX`.

*RST: INRange

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search criteria" on page 747

## 24.17.16 Custom: manchester / NRZ (option R&S RTP-K50)


- Configuration....2043
- Trigger....2060
- Filter....2066
- Decode results....2071
- Search settings....2077
- Search results....2082


### 24.17.16.1 Configuration

In all `BUS<m>:CMSB` commands, the suffix <m> selects the serial bus.

In all `BUS<m>:CMSB:FRAMe<n>` commands, the suffix <n> selects a frame number, and the suffix <o> selects a cell number.


BUS<m>:CMSB:CODing....2044
BUS<m>:CMSB:MANChester:DATA....2045
BUS<m>:CMSB:MANChester:POLarity....2045
BUS<m>:CMSB:MANChester:THReshold:HIGH....2045
BUS<m>:CMSB:MANChester:THReshold:LOW....2046
BUS<m>:CMSB:MANChester:THReshold:PRESet....2046
BUS<m>:CMSB:MANChester:THReshold:COUPling....2046
BUS<m>:CMSB:MANChester:CPHase....2047
BUS<m>:CMSB:NRZ:CLCK....2047
BUS<m>:CMSB:NRZ:DATA....2048
BUS<m>:CMSB:NRZ:IDLPolarity....2048
BUS<m>:CMSB:NRZ:CPOLarity....2049
BUS<m>:CMSB:NRZ:CPHase....2049
BUS<m>:CMSB:NRZ:ENBLe....2050
BUS<m>:CMSB:NRZ:ENAPolarity....2050
BUS<m>:CMSB:NRZ:POLarity....2050
BUS<m>:CMSB:NRZ:THReshold:CLCK....2051
BUS<m>:CMSB:NRZ:HYSTeresis:CLCK....2051
BUS<m>:CMSB:NRZ:THReshold:DATA....2051
BUS<m>:CMSB:NRZ:HYSTeresis:DATA....2051
BUS<m>:CMSB:NRZ:THReshold:ENBLe....2052
BUS<m>:CMSB:NRZ:HYSTeresis:ENBLe....2052

BUS<m>:CMSB:NRZ:THReshold:PRESet....2052
BUS<m>:CMSB:NRZ:THReshold:COUPling....2053
BUS<m>:CMSB:BITRate:ENABle....2053
BUS<m>:CMSB:BITRate:VALue....2054
BUS<m>:CMSB:GAPTime:ENABle....2054
BUS<m>:CMSB:GAPTime:VALue....2054
BUS<m>:CMSB:ADDFrame....2054
BUS<m>:CMSB:FCOunt?....2055
BUS<m>:CMSB:CLR....2055
BUS<m>:CMSB:FRAMe<n>:TYPE....2055
BUS<m>:CMSB:FRAMe<n>:APPend....2055
BUS<m>:CMSB:FRAMe<n>:CCOunt?....2056
BUS<m>:CMSB:FRAMe<n>:CELL<o>:NAME....2056
BUS<m>:CMSB:FRAMe<n>:CELL<o>:BITCount....2056
BUS<m>:CMSB:FRAMe<n>:CELL<o>:CONDition....2057
BUS<m>:CMSB:FRAMe<n>:CELL<o>:FORMat....2057
BUS<m>:CMSB:FRAMe<n>:CELL<o>:BITorder....2057
BUS<m>:CMSB:FRAMe<n>:CELL<o>:CRGB....2058
BUS<m>:CMSB:EXRBits....2058
BUS<m>:CMSB:FRAMe<n>:CELL<o>:CLMN....2058
BUS<m>:CMSB:LOAD....2059
BUS<m>:CMSB:SAVE....2059

**BUS<m>:CMSB:CODing** <CodingStandard>

Selects the custom serial bus coding standard.

**Suffix:**
<m> 1..4

**Parameters:**
<CodingStandard> MANC | MANT | NRZ | NRZU

**MANC**
Manchester (normal polarity)

**MANT**
Manchester II (inverted polarity).
Note that some additional subtle differences between MANC and MANT require separate protocols.

**NRZ**
NRZ (non-return-to-zero), clocked

**NRZU**
NRZ (non-return-to-zero), unclocked

*RST: MANC

**Manual operation:** See "Coding Standard" on page 751

**BUS<m>:CMSB:MANChester:DATA** <SourceMANData>

Selects the source channel for the data signal. For triggering on a serial bus, analog channels "C1"–"C4" are required. Otherwise, if no serial bus trigger has been selected, permitted source selections include the mathematical channels and the reference channels .

**Suffix:**
<m> 1..4

**Parameters:**
<SourceMANData> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: C1W1

**Manual operation:** See "Data Source" on page 751

**BUS<m>:CMSB:MANChester:POLarity** <PolarityData>

Selects the polarity of the custom serial bus data signal in Manchester coding standards.

**Suffix:**
<m> 1..4

**Parameters:**
<PolarityData> NORMal | INVert

**NORMal**
Manchester or Manchester II polarity remains unchanged

**INVert**
Manchester polarity is inverted and becomes Manchester II polarity, Manchester II polarity is inverted and becomes Manchester polarity

*RST: NORMal

**Manual operation:** See "Data Polarity (Manchester)" on page 752

**BUS<m>:CMSB:MANChester:THReshold:HIGH** <ThresholdUpper>

Sets the upper threshold for data signal digitization in Manchester coding standards.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdUpper> Range: -25 to 25
Increment: 0.1
*RST: 5
Default unit: V

**Manual operation:** See "Set Thresholds" on page 754

**BUS<m>:CMSB:MANChester:THReshold:LOW** <ThresholdLower>

Sets the lower threshold for data signal digitization in Manchester coding standards.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdLower> Range: -25 to 25
Increment: 0.1
*RST: -5
Default unit: V

**Manual operation:** See "Set Thresholds" on page 754

**BUS<m>:CMSB:MANChester:THReshold:PRESet** <PresetValue>

Sets the Manchester thresholds to predefined or individually definable voltage levels.

**Suffix:**
<m> 1..4

**Parameters:**
<PresetValue> V05 | V2 | V5 | V7 | MANual

**V05**
Sets the upper threshold to +0.5 V and the lower threshold to -0.5 V

**V2**
Sets the upper threshold to +2.0 V and the lower threshold to -2.0 V

**V5**
Sets the upper threshold to +5.0 V and the lower threshold to -5.0 V

**V7**
Sets the upper threshold to +7.0 V and the lower threshold to -7.0 V

**MANual**
Allows to set individual threshold voltage levels

*RST: V5

**BUS<m>:CMSB:MANChester:THReshold:COUPling** <ThresholdCoupling>

Couples the upper and lower threshold values for the Manchester and Manchester II coding standards. The values are coupled to voltages with the same magnitude but opposite sign (positive for the upper threshold and negative for the lower threshold). However, if the upper threshold is set to a negative voltage or the lower threshold is set to a positive voltage, coupling is disabled, and the other voltage (the one that was not actively set) is automatically adjusted, to avoid an upper threshold below the lower one, or a lower threshold above the upper one.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdCoupling> ON | OFF

**ON**
Activates coupling of the upper and lower threshold values.

**OFF**
Disables coupling of the upper and lower threshold values.

*RST: ON

**Manual operation:** See "Set Thresholds" on page 754

---

**BUS<m>:CMSB:MANChester:CPHase** <ClockPhase>

Selects the phase of the custom serial bus clock signal for the "Manchester" coding standards. For details, see "Clock Phase (Manchester)" on page 753.

**Suffix:**
<m> 1..4

**Parameters:**
<ClockPhase> FEDGe | SEDGe

**FEDGe**
Sets the sampling edge to be on the first edge.

**SEDGe**
Sets the sampling edge to be on the second edge.

**AUTO**
Lets the decoder automatically select the method ("First Edge" or "Second Edge") for detecting the sampling edge.

*RST: SEDGe

**Manual operation:** See "Clock Phase (Manchester)" on page 753

---

**BUS<m>:CMSB:NRZ:CLCK** <SourceNRZClock>

Selects the source channel for the clock signal in the NRZ Clocked coding standard.

For triggering on the serial bus when the NRZ clocked coding standard is selected, analog or digital channel sources are required.

Otherwise, if no serial bus trigger has been selected, permitted source selections include the mathematical channels and the reference channels.

**Suffix:**
<m> 1..4

**Parameters:**

<SourceNRZClock> NONE | C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

*RST: C2W1

**Manual operation:** See "Clock Source" on page 752

---

**BUS<m>:CMSB:NRZ:DATA** <SourceNRZData>

Selects the source channel for the data signal in NRZ coding standards.

For triggering on the serial bus when the NRZ clocked coding standard is selected, analog or digital channel sources are required.

For triggering on the serial bus when the NRZ unclocked coding standard is selected, analog channel sources are required.

Otherwise, if no serial bus trigger has been selected, permitted source selections include the mathematical channels and the reference channels.

**Suffix:**

<m> 1..4

**Parameters:**

<SourceNRZData> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

*RST: C1W1

**Manual operation:** See "Data Source" on page 751

---

**BUS<m>:CMSB:NRZ:IDLPolarity** <PolarityIdle>

Selects the idle polarity of the custom serial bus data signal (only available for the coding standard "NRZ Unclocked").

**Suffix:**

<m> 1..4

**Parameters:**

<PolarityIdle> IDLLow | IDLHigh

**IDLLow**
Sets the base value of the data bus to be "0". After an idle period, the data signal starts with a low-to-high transition

**IDLHigh**
Sets the base value of the data bus to be "1". After an idle period, the data signal starts with a high-to-low transition

*RST: IDLLow

**Manual operation:** See "Data Idle Polarity (NRZ Unclocked)" on page 753

**BUS<m>:CMSB:NRZ:CPOLarity** <PolarityClock>

Selects the polarity of the clock signal for the coding standard NRZ Clocked.

**Suffix:**
<m> 1..4

**Parameters:**
<PolarityClock> IDLLow | IDLHigh

**IDLLow**
Sets the base value of the clock to be "0", the clock signal starts with a low-to-high transition

**IDLHigh**
Sets the base value of the clock to be "1", the clock signal starts with a high-to-low transition.

*RST: IDLLow

**Manual operation:** See "Clock Polarity (NRZ Clocked)" on page 753

**BUS<m>:CMSB:NRZ:CPHase** <ClockPhase>

Selects the phase of the custom serial bus clock signal for the coding standard "NRZ Clocked", depending on `BUS<m>:CMSB:NRZ:IDLPolarity` on page 2048.

**Suffix:**
<m> 1..4

**Parameters:**
<ClockPhase> FEDGe | SEDGe

**FEDGe**
Sets the clocking transaction to be on the first edge:
If Clock Polarity = "IDLLow", data are captured on the clock's rising edge (low-to-high transition) and propagated on a falling edge
If Clock Polarity = "IDLHigh", data are captured on the clock's falling edge (high-to-low transition) and propagated on a rising edge

**SEDGe**
Sets the clocking transaction to be on the second edge:
If Clock Polarity = "IDLLow", data are captured on the clock's falling edge (high-to-low transition) and propagated on a rising edge
If Clock Polarity = "IDLHigh", data are captured on the clock's rising edge (low-to-high transition) and propagated on a falling edge

*RST: FEDGe

**Manual operation:** See "Clock Phase (NRZ Clocked)" on page 754

**BUS<m>:CMSB:NRZ:ENBLe** <SourceNRZEnable>

Selects the input source for the custom serial bus enable signal.

If an input is chosen, signals will be only decoded while this channel is in the enabled state. This allows you to mark a time when the signal on the selected source is active and when not.

For triggering on a serial bus, analog channels "C1"–"C4" are required. Otherwise, if no serial bus trigger has been selected, permitted source selections include the mathematical channels and the reference channels.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceNRZEnable> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

**Manual operation:** See "Enable Source (NRZ)" on page 754

**BUS<m>:CMSB:NRZ:ENAPolarity** <PolarityEnable>

Sets whether the transmitted enable signal is active when the voltage is below the threshold (`ENALow`) or higher than it (`ENAHIgh`).

**Suffix:**
<m> 1..4

**Parameters:**
<PolarityEnable> ENALow | ENAHigh

*RST: ENAHigh

**Manual operation:** See "Enable Polarity (NRZ)" on page 754

**BUS<m>:CMSB:NRZ:POLarity** <PolarityData>

Selects the polarity of the custom serial bus data signal in NRZ coding standards.

**Suffix:**
<m> 1..4

**Parameters:**
<PolarityData> ACTLow | ACTHigh

**ACTLow**
Active low: the value "1" is represented by a voltage below the threshold

**ACTHigh**
Active high: the value "1" is represented by a voltage above the threshold

*RST: ACTHigh

**Manual operation:** See "Data Polarity (NRZ)" on page 752

---

**BUS<m>:CMSB:NRZ:THReshold:CLCK** <ThresholdClock>

Sets the threshold for the clock signal digitization in the NRZ Clocked coding standard.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdClock> Range: -25 to 25
Increment: 0.1
*RST: 2
Default unit: V

**Manual operation:** See "Set Thresholds" on page 754

---

**BUS<m>:CMSB:NRZ:HYSTeresis:CLCK** <HystClock>

Sets the hysteresis size on the clock channel.

**Suffix:**
<m> 1..4

**Parameters:**
<HystClock> Range: -25 to 25
Increment: 0.1
*RST: 0.1
Default unit: V

**Manual operation:** See "Set Thresholds" on page 754

---

**BUS<m>:CMSB:NRZ:THReshold:DATA** <ThresholdData>

Sets the threshold for the data signal digitization in NRZ coding standards.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdData> Range: -25 to 25
Increment: 0.1
*RST: 2
Default unit: V

**Manual operation:** See "Set Thresholds" on page 754

---

**BUS<m>:CMSB:NRZ:HYSTeresis:DATA** <HystData>

Sets the hysteresis size on the data line channel.

**Suffix:**
<m> 1..4

**Parameters:**

<HystData> Range: -25 to 25
Increment: 0.1
*RST: 0.1
Default unit: V

**Manual operation:** See "Set Thresholds" on page 754

---

**BUS<m>:CMSB:NRZ:THReshold:ENBLe** <ThresholdEnable>

Sets the threshold for the enable signal digitization in NRZ coding standards.

**Suffix:**

<m> 1..4

**Parameters:**

<ThresholdEnable> Range: -25 to 25
Increment: 0.1
*RST: 2
Default unit: V

**Manual operation:** See "Set Thresholds" on page 754

---

**BUS<m>:CMSB:NRZ:HYSTeresis:ENBLe** <HystEnable>

Sets the hysteresis size on the enable line channel.

**Suffix:**

<m> 1..4

**Parameters:**

<HystEnable> Range: -25 to 25
Increment: 0.1
*RST: 0.1
Default unit: V

**Manual operation:** See "Set Thresholds" on page 754

---

**BUS<m>:CMSB:NRZ:THReshold:PRESet** <PresetValue>

Sets the NRZ thresholds to predefined or individually definable voltage levels.

**Suffix:**

<m> 1..4

**Parameters:**

<PresetValue> V05 | V2 | V5 | V7 | MANual

**V05**
Sets the clock and data threshold to +0.5 V (in case of NRZ Unclocked: data threshold, only)

**V2**
Sets the clock and data threshold to +2.0 V (in case of NRZ Unclocked: data threshold, only)

**V5**
Sets the clock and data threshold to +5.0 V (in case of NRZ Unclocked: data threshold, only)

**V7**
Sets the clock and data threshold to +7.0 V (in case of NRZ Unclocked: data threshold, only)

**MANual**
Allows to set individual threshold voltage levels

*RST: V5

---

**BUS<m>:CMSB:NRZ:THReshold:COUPling** <ThresholdCoupling>

Couples the clock and data threshold values for the NRZ Clocked coding standard. The values are coupled to the same number.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdCoupling> ON | OFF

**ON**
Activates coupling of the NRZ clock and data threshold values.

**OFF**
Disables coupling of the NRZ clock and data threshold values.

*RST: ON

**Manual operation:** See "Set Thresholds" on page 754

---

**BUS<m>:CMSB:BITRate:ENABle** <BitrateEnable>

Enables the bit rate settings (must always be enabled for the coding standard "NRZ Unclocked", and also for triggering on signals in any coding standard).

**Suffix:**
<m> 1..4

**Parameters:**
<BitrateEnable> ON | OFF

**ON**
Bit rate settings enabled

**OFF**
Bit rate settings disabled

*RST: OFF

**Manual operation:** See "Check bit rate" on page 757

**BUS<m>:CMSB:BITRate:VALue** <Bitrate>

Sets the transmission speed setting for the custom serial bus data signal.

**Suffix:**
<m> 1..4

**Parameters:**
<Bitrate> Range: 300 to 2E9
Increment: 100E3
*RST: 10E6
Default unit: bps

**Manual operation:** See "Bit Rate" on page 757

**BUS<m>:CMSB:GAPTime:ENABle** <GapTimeEnable>

Enables the gap time settings (must always be enabled for the coding standard "NRZ Unclocked", and also for triggering on signals in any coding standard).

**Suffix:**
<m> 1..4

**Parameters:**
<GapTimeEnable> ON | OFF

**ON**
Gap time settings enabled

**OFF**
Gap time settings disabled

*RST: OFF

**Manual operation:** See "Check gap time" on page 757

**BUS<m>:CMSB:GAPTime:VALue** <MinGapTime>

Sets a minimum gap time for synchronization.

**Suffix:**
<m> 1..4

**Parameters:**
<MinGapTime> Range: 1E-9 to 1
Increment: 1E-9
*RST: 10E-6
Default unit: s

**Manual operation:** See "Min gap time" on page 758

**BUS<m>:CMSB:ADDFrame**

Creates an empty frame format description and adds it to the end of the frame description list.

**Suffix:**
<m> 1..4

**Usage:** Event

**Manual operation:** See "Frames" on page 759

---

**BUS<m>:CMSB:FCOunt?**

Returns the number of frames.

**Suffix:**
<m> 1..4

**Return values:**
<Count>

**Usage:** Query only

**Manual operation:** See "Frames" on page 759

---

**BUS<m>:CMSB:CLR**

Erases all cells and frames that have been created for a specific custom protocol.

**Suffix:**
<m> 1..4

**Usage:** Event

**Manual operation:** See "Frames" on page 759

---

**BUS<m>:CMSB:FRAMe<n>:TYPE** <FrameType>

Enables the user to set a string to describe the frame type, typically according to the applicable protocol standard specifications. (For example, MDIO (option R&S RTP-K55) defines the frames READ, WRITE, ADDRESS, etc.)

**Suffix:**
<m> 1..4

<n> *

**Parameters:**
<FrameType>

**Manual operation:** See "Frames" on page 759

---

**BUS<m>:CMSB:FRAMe<n>:APPend**

Creates an empty cell description and adds it to the end of the active frame description.

**Suffix:**
<m> 1..4

<n> *

**Usage:** Event

**Manual operation:** See "Field name" on page 760

---

**BUS<m>:CMSB:FRAMe<n>:CCOunt?**

Returns the number of cells in the specified frame.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Count>

**Usage:** Query only

**Manual operation:** See "Field name" on page 760

---

**BUS<m>:CMSB:FRAMe<n>:CELL<o>:NAME** <CellName>

Enables the user to set a cell name within a frame. Names do not have to be unique, they are just for user support.

**Suffix:**
<m> 1..4

<n> *

<o> *

**Parameters:**
<CellName>

**Manual operation:** See "Field Name" on page 760

---

**BUS<m>:CMSB:FRAMe<n>:CELL<o>:BITCount** <BitCount>

Sets the bit count of a cell, hence its length. Based upon the lengths of the previous cells, this also defines the position of the cell start and end within a frame.

**Suffix:**
<m> 1..4

<n> *

<o> *

**Parameters:**
<BitCount>
Range: 1 to 64
Increment: 1
*RST: 1

**Manual operation:** See "Bits" on page 761

**BUS<m>:CMSB:FRAMe<n>:CELL<o>:CONDition** <Condition>

Sets various operators for a cell, to identify, e.g., mandatory values such as a CRC checksum or an ID, that help to identify a frame.

The implemented conditions and functionalities are the "equal" and "array" operators.

The numeric format of the condition needs to be set according to `BUS<m>:CMSB:FRAMe<n>:CELL<o>:FORMat` on page 2057.

**Suffix:**
<m> 1..4
<n> *
<o> *

**Parameters:**
<Condition>

**Manual operation:** See "Condition" on page 761

**BUS<m>:CMSB:FRAMe<n>:CELL<o>:FORMat** <CellFormat>

Selects the numeric data format for the command `BUS<m>:CMSB:FRAMe<n>:CELL<o>:CONDition` on page 2057, as well as for the result and honeycomb display.

**Suffix:**
<m> 1..4
<n> *
<o> *

**Parameters:**
<CellFormat> DEC | HEX | OCT | BIN

**DEC**
Decimal

**HEX**
Hexadecimal

**OCT**
Octal

**BIN**
Binary

*RST: BIN

**Manual operation:** See "Format" on page 764

**BUS<m>:CMSB:FRAMe<n>:CELL<o>:BITorder** <BitOrder>

Selects in which order the bits of a cell are evaluated, as well as presented in the results table and honeycomb display.

**Suffix:**
<m> 1..4
<n> *
<o> *

**Parameters:**
<BitOrder> LSBF | MSBF

**LSBF**
Least significant bit first, evaluation starts at the LSB

**MSBF**
Most significant bit first, evaluation starts at the MSB

*RST: MSBF

**Manual operation:** See "Bit Order" on page 765

---

**BUS<m>:CMSB:FRAMe<n>:CELL<o>:CRGB** <Color>

Selects a cell's color representation in the honeycomb display.

**Suffix:**
<m> 1..4
<n> *
<o> *

**Parameters:**
<Color> ARGB value of the color to be used for the table entry.
ARGB=<Opacity(alpha) value><red value><green value><blue value>, in hexadecimal or decimal format.

Range: 0 to 4294967295
Increment: 1
*RST: 0

**Manual operation:** See "Color" on page 765

---

**BUS<m>:CMSB:EXRBits**

Exports the decoded frames as raw bits.

**Suffix:**
<m> 1..4

**Usage:** Event

---

**BUS<m>:CMSB:FRAMe<n>:CELL<o>:CLMN** <Column>

Selects which cell shall be displayed in which result column of the decode table.

The decode table supports three result columns, which have to be unique for each frame type. For different frame types, though, different result columns can be defined to display unrelated information.

**Suffix:**
<m> 1..4

<n> *

<o> *

**Parameters:**
<Column> NONE | COL1 | COL2 | COL3

**NONE**
The result is not displayed in the decode table.

**COL1**
The result is displayed in column 1 of the decode table.

**COL2**
The result is displayed in column 2 of the decode table.

**COL3**
The result is displayed in column 3 of the decode table.

*RST: NONE

**Manual operation:** See "Result" on page 765

---

**BUS<m>:CMSB:LOAD** <FileName>

Opens an existing frame description file in xml format. The default path is `\\Public\Documents\Rohde-Schwarz\RTx\SaveXML`

**Suffix:**
<m> 1..4

**Setting parameters:**
<FileName>

**Usage:** Setting only

**Manual operation:** See "Open or Save XML File" on page 765

---

**BUS<m>:CMSB:SAVE** <FileName>

Saves a created frame description into an xml file ("Save As..."). The default path is `\\Public\Documents\Rohde-Schwarz\RTx\SaveXML`

**Suffix:**
<m> 1..4

**Setting parameters:**
<FileName>

**Usage:** Setting only

**Manual operation:** See "Open or Save XML File" on page 765

### 24.17.16.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:CMSB:TYPE....2060
TRIGger<m>:CMSB:PATTern....2061
TRIGger<m>:CMSB:ICONdition....2061
TRIGger<m>:CMSB:IMIN....2061
TRIGger<m>:CMSB:IMAX....2062
TRIGger<m>:CMSB:NRZ:WRDLength....2062
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:ENABle....2062
TRIGger<m>:CMSB:ADVanced:FRENable....2062
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:ENABle....2063
TRIGger<m>:CMSB:ADVanced:FIENable....2063
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:BIT....2063
TRIGger<m>:CMSB:ADVanced:BIT....2063
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:DMAX....2064
TRIGger<m>:CMSB:ADVanced:DMAX....2064
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:DMIN....2064
TRIGger<m>:CMSB:ADVanced:DMIN....2064
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:DOPerator....2064
TRIGger<m>:CMSB:ADVanced:DOPerator....2064
TRIGger<m>:CMSB:ADVanced:ERRor<n>:ENABle....2065
TRIGger<m>:CMSB:ADVanced:ERENable....2065
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:IMAX....2065
TRIGger<m>:CMSB:ADVanced:IMAX....2065
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:IMIN....2066
TRIGger<m>:CMSB:ADVanced:IMIN....2066
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:IOPerator....2066
TRIGger<m>:CMSB:ADVanced:IOPerator....2066

---

**TRIGger<m>:CMSB:TYPE** <Type>

Selects the trigger type for custom serial bus analysis.

**Suffix:**
<m> 1..3

**Parameters:**
<Type> STARt | PATTern | ADVanced

**STARt**
Triggers on the frame start, which is the end of the gap time as specified in `BUS<m>:CMSB:GAPTime:VALue` on page 2054

**ADVanced**
Advanced trigger with `TRIGger:CMSB:ADVanced` commands. Only available if frames are defined.

**PATTern**
Triggers on a data pattern to be specified in `TRIGger<m>:CMSB:PATTern` on page 2061

*RST: STARt

**Manual operation:** See "Type" on page 770

---

**TRIGger<m>:CMSB:PATTern** <DataPattern>

Sets the pattern match conditions for a payload data check. The trigger is set to the first occurrence of a matching data bit pattern (which can be freely specified), starting after the minimum gap time (`BUS<m>:CMSB:GAPTime:VALue` on page 2054), and after the detected start of the data frame.

**Suffix:**
<m> 1..3

**Parameters:**
<DataPattern>

**Manual operation:** See "Pattern" on page 771

---

**TRIGger<m>:CMSB:ICONdition** <DataIdxOperator>

Sets the operator to set a specific bit index (data position).

**Suffix:**
<m> 1..3

**Parameters:**
<DataIdxOperator> EQUal | GETHan | INRange

**EQUal**
Equal

**GETHan**
Greater than or equal

**INRange**
In range

*RST: INRange

**Manual operation:** See "Pattern" on page 771

---

**TRIGger<m>:CMSB:IMIN** <DataPosition>

Sets the bit index (data position). If `TRIGger<m>:CMSB:ICONdition` on page 2061 is set to "INRange", "IMIN" sets the start value of the bit index range.

**Suffix:**
<m> 1..3

**Parameters:**

<DataPosition> Range: 0 to 65535
Increment: 1
*RST: 0

**Manual operation:** See "Pattern" on page 771

---

**TRIGger<m>:CMSB:IMAX** <DataPositionTo>

Sets the end value of the bit index range (data position range), if `TRIGger<m>:CMSB:ICONdition` on page 2061 is set to "INRange".

**Suffix:**

<m> 1..3

**Parameters:**

<DataPositionTo> Range: 0 to 65535
Increment: 1
*RST: 65535

**Manual operation:** See "Pattern" on page 771

---

**TRIGger<m>:CMSB:NRZ:WRDLength** <NRZWordLength>

Sets the number of bits in an NRZ Unclocked word.

**Suffix:**

<m> 1..3

**Parameters:**

<NRZWordLength> Range: 0 to 31
Increment: 1
*RST: 8

**Manual operation:** See "Pattern" on page 771

---

**TRIGger<m>:CMSB:ADVanced:FRAMe<n>:ENABle** <Enable>
**TRIGger<m>:CMSB:ADVanced:FRENable** <Frame>, <Enabler>

Enables or disables the specific frame to be triggered on.

**Suffix:**

<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Setting parameters:**

<Frame>

<Enabler> ON | OFF

**Example:** `:TRIGger:CMSB:ADVanced:FRENable "Master" 1`
Enables the Master frame.
`:TRIGger:CMSB:ADVanced:FRAMe1:ENABle 1`
Enables the 1st frame that you have defined.

**Usage:** Setting only

---

**TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:ENABle** <CondEnabler>
**TRIGger<m>:CMSB:ADVanced:FIENable** <Frame>, <Field>, <Enabler>

Enables or disables the specific field within the defined frame to be triggered on.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Setting parameters:**
<Frame>

<Field>

<Enabler> ON | OFF

**Example:** `:TRIGger:CMSB:ADVanced:FIEN "Master", "CRC", 1`
Enables the CRC field of the Master frame.
`:TRIGger:CMSB:ADVanced:FRAMe1:FLD3:ENABle 1`
Enables the 3rd field of the 1st frame that you have defined.

**Usage:** Setting only

**Manual operation:** See "Field name / Condition / Enable" on page 773

---

**TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:BIT** <BitState>
**TRIGger<m>:CMSB:ADVanced:BIT** <Frame>, <Field>, <Bit>

Sets the bit state of a field to be triggered on that only consists of one bit.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Setting parameters:**
<Frame>

<Field>

<Bit> ONE | ZERO | DC

**ONE**
1

**ZERO**
0

**DC**
"Do not care" (DC) = X

**Usage:** Setting only

**Manual operation:** See "BitState" on page 774

**TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:DMAX** <DataMax>
**TRIGger<m>:CMSB:ADVanced:DMAX** <Frame>, <Field>, <Data>

Sets the end value of a data pattern range if `TRIGger<m>:CMSB:ADVanced:DOPerator` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Setting parameters:**
<Frame>

<Field>

<Data>

**Usage:** Setting only

**Manual operation:** See "Data" on page 774

**TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:DMIN** <DataMin>
**TRIGger<m>:CMSB:ADVanced:DMIN** <Frame>, <Field>, <Data>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Setting parameters:**
<Frame>

<Field>

<Data>

**Usage:** Setting only

**Manual operation:** See "Data" on page 774

**TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:DOPerator** <DataOperator>
**TRIGger<m>:CMSB:ADVanced:DOPerator** <Frame>, <Field>, <Operator>

Sets the operator for the data pattern.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Setting parameters:**
<Frame>

<Field>

<Operator> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**ANY = OFF**
The data of the required pattern is not relevant.

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
This condition requires one endpoint value to be set using `TRIGger<m>:CMSB:ADVanced:DMIN`.

**INRange | RANGe**
This condition requires a range of endpoint values to be set using `TRIGger<m>:CMSB:ADVanced:DMAX` and `TRIGger<m>:CMSB:ADVanced:DMIN`.

**Usage:** Setting only

**Manual operation:** See "Data" on page 774

---

**TRIGger<m>:CMSB:ADVanced:ERRor<n>:ENABle** <Enable>
**TRIGger<m>:CMSB:ADVanced:ERENable** <ErrorName>, <Enabler>

Enables triggering on errors. You can search for all error types in parallel.

**Suffix:**
<m> 1..3

**Setting parameters:**
<ErrorName>

<Enabler> ON | OFF

**Usage:** Setting only

**Manual operation:** See "Error types: Item name, Enable" on page 773

---

**TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:IMAX** <IndexMax>
**TRIGger<m>:CMSB:ADVanced:IMAX** <Frame>, <Field>, <Data>

Sets the end value of an index range if `TRIGger<m>:CMSB:ADVanced:IMAX` is set to `INRange` or `RANGe`.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Setting parameters:**
<Frame>

<Field>

<Data>

**Usage:** Setting only

**Manual operation:** See "Index" on page 774

**TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:IMIN** <IndexMin>
**TRIGger<m>:CMSB:ADVanced:IMIN** <Frame>, <Field>, <Data>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Setting parameters:**
<Frame>

<Field>

<Data>

**Usage:** Setting only

**Manual operation:** See "Index" on page 774

**TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:IOPerator** <IndexOperator>
**TRIGger<m>:CMSB:ADVanced:IOPerator** <Frame>, <Field>, <Operator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<m> Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

**Setting parameters:**
<Frame>

<Field>

<Operator> ANY | OFF | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | RANGe

**ANY = OFF**
The index of the required pattern is not relevant.

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
This condition requires one endpoint value to be set using `TRIGger<m>:CMSB:ADVanced:IMIN`.

**INRange | RANGe**
This condition requires a range of endpoint values to be set using `TRIGger<m>:CMSB:ADVanced:IMAX` and `TRIGger<m>:CMSB:ADVanced:IMIN`.

**Usage:** Setting only

**Manual operation:** See "Index" on page 774

### 24.17.16.3 Filter

**BUS<m>:CMSB:FILTer:ENABle** <Enable>

Enables filtering for the custom serial bus.

**Suffix:**
<m> 1..4

**Parameters:**
<Enable> ON | OFF
*RST: OFF

**Manual operation:** See "Apply Filter" on page 768

---

**BUS<m>:CMSB:FILTer:ERENable** <ErrorName>, <Enabler>

Enables filtering on error.

**Suffix:**
<m> 1..4

**Setting parameters:**
<ErrorName>

<Enabler> ON | OFF

**Usage:** Setting only

---

**BUS<m>:CMSB:FILTer:ERRor<n>:ENABle** <Enable>

Enables filtering on error. You can filter all error types in parallel.

**Suffix:**
<m> 1..4

<n> *

**Parameters:**
<Enable> ON | OFF
*RST: ON

**Manual operation:** See "Error types: Item name, Enable" on page 777

---

**BUS<m>:CMSB:FILTer:FRAMe<n>:ENABle** <Enable>

Enables or disables the specific frame to be filtered.

**Suffix:**
<m> 1..4

<n> *

**Parameters:**
<Enable> ON | OFF
*RST: OFF

**Manual operation:** See "Frame type" on page 776

**BUS<m>:CMSB:FILTer:FRENable** <Frame>, <Enabler>

Enables or disables the specific frame to be filtered.

**Suffix:**
<m> 1..4

**Setting parameters:**
<Frame>

<Enabler> ON | OFF

**Usage:** Setting only

**BUS<m>:CMSB:FILTer:FIENable** <Frame>, <Field>, <Enabler>

Enables or disables the specific field within the defined frame to be filtered.

**Suffix:**
<m> 1..4

**Setting parameters:**
<Frame>

<Field>

<Enabler> ON | OFF

**Usage:** Setting only

**BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:BIT** <BitState>

Sets the bit state of a field to be filtered on that only consists of one bit.

**Suffix:**
<m> 1..4

<n> *

<o> *

**Parameters:**
<BitState> ONE | ZERO | DC
*RST: DC

**Manual operation:** See "Bit State" on page 777

**BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:DMAX** <DataMax>

Sets the end value of a data pattern range if `BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:DOPerator` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..4

<n> *

<o> *

**Parameters:**
<DataMax>

**Manual operation:** See "Data" on page 777

---

**BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:DMIN** <DataMin>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<m> 1..4

<n> *

<o> *

**Parameters:**
<DataMin>

**Manual operation:** See "Data" on page 777

---

**BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:DOPerator** <DataOperator>

Sets the operator for the data pattern.

**Suffix:**
<m> 1..4

<n> *

<o> *

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

*RST: EQUal

**Manual operation:** See "Data" on page 777

---

**BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:ENABle** <CondEnabler>

Enables or disables the specific field within the defined frame to be filtered.

**Suffix:**
<m> 1..4

<n> *

<o> *

**Parameters:**
<CondEnabler> ON | OFF

*RST: OFF

**Manual operation:** See "Field name / Condition / Enable" on page 777

---

**BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:IMAX** <IndexMax>

Sets the end value of an index range if `BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:IOPerator` is set to `INRange` or `RANGe`.

**Suffix:**
<m> 1..4
<n> *
<o> *

**Parameters:**
<IndexMax> Range: 1 to 65534
Increment: 1
*RST: 65534

**Manual operation:** See "Index" on page 777

---

**BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:IMIN** <IndexMin>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<m> 1..4
<n> *
<o> *

**Parameters:**
<IndexMin> Range: 1 to 65534
Increment: 1
*RST: 1

**Manual operation:** See "Index" on page 777

---

**BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:IOPerator** <IndexOperator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<m> 1..4
<n> *
<o> *

**Parameters:**
<IndexOperator> EQUal | INRange | RANGe
*RST: INRange

**Manual operation:** See "Index" on page 777

### 24.17.16.4 Decode results

In all BUS<m>:CSMB:RESult<n> commands, the suffix <m> selects the serial bus, the suffix <n> selects the result number in the decode table, and the suffix <o> selects the cell number.

The following is an example for a set of query commands for bus 1 and result 1 is shown in the following, together with examples for outcomes of these queries:

- :BUS1:CMSB:RCOunt? !5

- :BUS1:CMSB:RESult1:STATe? !OK
- :BUS1:CMSB:RESult1:STARt? ! -0.0024964177
- :BUS1:CMSB:RESult1:STOP? ! -0.0024030384
- :BUS1:CMSB:RESult1:TYPE? !ff
- :BUS1:CMSB:RESult1:CONE? !0b11111111
- :BUS1:CMSB:RESult1:CTWO? !0xAA
- :BUS1:CMSB:RESult1:CTHR? !0xF590

- :BUS1:CMSB:RESult1:CCOunt? !5

- :BUS1:CMSB:RESult1:CELL1:STAT? !OK
- :BUS1:CMSB:RESult1:CELL1:NAME? !Start Delim
- :BUS1:CMSB:RESult1:CELL1:VALue? !101010101HL10HL0

- :BUS1:CMSB:RESult1:CELL2:STAT? !OK
- :BUS1:CMSB:RESult1:CELL2:NAME? !OP-FF
- :BUS1:CMSB:RESult1:CELL2:VALue? !0b11111111

- :BUS1:CMSB:RESult1:CELL3:STAT? !OK
- :BUS1:CMSB:RESult1:CELL3:NAME? !data
- :BUS1:CMSB:RESult1:CELL3:VALue? !0xAA

- :BUS1:CMSB:RESult1:CELL4:STAT? !OK
- :BUS1:CMSB:RESult1:CELL4:NAME? !CRC
- :BUS1:CMSB:RESult1:CELL4:VALue? !0xF590

- :BUS1:CMSB:RESult1:CELL5:STAT? !OK
- :BUS1:CMSB:RESult1:CELL5:NAME? !End Delim
- :BUS1:CMSB:RESult1:CELL5:VALue? !1HLHL101

To show the results on the screen, use the following commands:

- BUS<m>:RESult on page 1638

BUS<m>:CMSB:RCOunt? 2072
BUS<m>:CMSB:RESult<n>:STATe? 2072
BUS<m>:CMSB:RESult<n>:STARt? 2073
BUS<m>:CMSB:RESult<n>:STOP? 2073
BUS<m>:CMSB:RESult<n>:TYPE? 2073
BUS<m>:CMSB:RESult<n>:CONE? 2074
BUS<m>:CMSB:RESult<n>:CTWO? 2074
BUS<m>:CMSB:RESult<n>:CTHRee? 2074
BUS<m>:CMSB:RESult<n>:CCOunt? 2075
BUS<m>:CMSB:RESult<n>:CELL<o>:NAME? 2075
BUS<m>:CMSB:RESult<n>:CELL<o>:STARt? 2075
BUS<m>:CMSB:RESult<n>:CELL<o>:STOP? 2075
BUS<m>:CMSB:RESult<n>:CELL<o>:STATe? 2076
BUS<m>:CMSB:RESult<n>:CELL<o>:VALue? 2077

### BUS<m>:CMSB:RCOunt?

Returns the count number of decoded result frames in a custom serial bus waveform. Basically, this is the maximum result index <n> when querying results by using `BUS<m>:CMSB:RESult<n>:XXX`.

**Suffix:**
<m> 1..4

**Return values:**
<Count>

**Usage:** Query only

### BUS<m>:CMSB:RESult<n>:STATe?

Returns the overall state of the frame: either OK or the relevant error condition. R&S RTP-K50 marks each frame with a status that indicates whether the decode succeeded or not.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<State> OK | LENGth | UNKNown | INComplete | CRC | PARity

**OK**
The frame was decoded normally and conforms to the frame description.

**LENGth**
The length error indicates that the frame ended prematurely, or an array in the frame had too few elements. The amount of bits that the software expected (based upon the user's frame description) was not found before the frame was terminated. This might occur because a new frame synchronized, or a gap appeared between the bits.

**UNKNown**
Unknown error

**INComplete**
The frame ended prematurely because it extends past the end of the record.

**CRC**
Checksum error in cyclic redundancy check (error in data)

**PARity**
Parity bit error, indicating a transmission error (only available if a parity is configured)

*RST: OK

**Usage:** Query only

---

**BUS<m>:CMSB:RESult<n>:STARt?**

Returns the start time of the frame.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:CMSB:RESult<n>:STOP?**

Returns the stop time of the frame.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:CMSB:RESult<n>:TYPE?**

Returns the name of the selected frame (n) from the user defined frame format description, labeled according to `BUS<m>:CMSB:FRAMe<n>:TYPE` on page 2055.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<Type>

**Usage:** Query only

---

**BUS<m>:CMSB:RESult<n>:CONE?**

Returns the 1st cell content as specified in the "Result" column of the "Frame Format" specification table.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<Custom1>

**Usage:** Query only

---

**BUS<m>:CMSB:RESult<n>:CTWO?**

Returns the 2nd cell content as specified in the "Result" column of the "Frame Format" specification table.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<Custom2>

**Usage:** Query only

---

**BUS<m>:CMSB:RESult<n>:CTHRee?**

Returns the 3rd cell content as specified in the "Result" column of the "Frame Format" specification table.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<Custom3>

**Usage:** Query only

**BUS<m>:CMSB:RESult<n>:CCOunt?**

Returns the number of decoded cells in the specified result frame.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<NumWords> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:CMSB:RESult<n>:CELL<o>:NAME?**

Returns the name of the specified cell. Cell names are not necessarily unique.

**Suffix:**
<m> 1..4

<n> *

<o> *

**Return values:**
<Name>

**Usage:** Query only

**BUS<m>:CMSB:RESult<n>:CELL<o>:STARt?**

Returns the start time of the selected cell.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *

<o> *
Selects the cell.

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:CMSB:RESult<n>:CELL<o>:STOP?**

Returns the end time of the selected cell.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *

<o> *
Selects the cell.

**Return values:**

<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:CMSB:RESult<n>:CELL<o>:STATe?**

Returns the state of the cell.

**Suffix:**

<m> 1..4

<n> *

<o> *

**Return values:**

<State> OK | LENGth | UNKNown | INComplete | CRC | PARity

**OK**
No error detected

**LENGth**
The length error indicates that the cell ended prematurely. The amount of bits that the software expected (based upon the user's frame description) was not found before the cell was terminated. This might occur because a new frame synchronized, or a gap appeared between the bits.

**UNKNown**
Unknown error

**INComplete**
The cell ended prematurely because it extends past the end of the record.

**CRC**
Checksum error in cyclic redundancy check (error in data)

**PARity**
Parity bit error, indicating a transmission error (only available if a parity is configured)

**Usage:** Query only

**BUS<m>:CMSB:RESult<n>:CELL<o>:VALue?**

Data content of the specified cell.

**Suffix:**
<m> 1..4
<n> *
<o> *

**Return values:**
<Value>

**Usage:** Query only

### 24.17.16.5 Search settings

SEARch:TRIGger:CMSB:ERENable ... 2077
SEARch:TRIGger:CMSB:ERRor<m>:ENABle ... 2077
SEARch:TRIGger:CMSB:FIENable ... 2078
SEARch:TRIGger:CMSB:FRENable ... 2078
SEARch:TRIGger:CMSB:BIT ... 2078
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:BIT ... 2078
SEARch:TRIGger:CMSB:DMAX ... 2079
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMAX ... 2079
SEARch:TRIGger:CMSB:DMIN ... 2079
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMIN ... 2079
SEARch:TRIGger:CMSB:DOPerator ... 2080
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DOPerator ... 2080
SEARch:TRIGger:CMSB:FRAMe<m>:ENABle ... 2080
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:ENABle ... 2080
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMAX ... 2081
SEARch:TRIGger:CMSB:IMAX ... 2081
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMIN ... 2081
SEARch:TRIGger:CMSB:IMIN ... 2081
SEARch:TRIGger:CMSB:IOPerator ... 2081
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IOPerator ... 2081

**SEARch:TRIGger:CMSB:ERENable** <SearchName>,<ErrorName>, <Enabler>
**SEARch:TRIGger:CMSB:ERRor<m>:ENABle** <SearchName>,<Enable>
**SEARch:TRIGger:CMSB:ERRor<m>:ENABle?** <SearchName>

Defines the error type to be searched for. You can search for all error types in parallel.

**Suffix:**
<m> *

**Parameters:**
<Enable> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame" on page 779

---

**SEARch:TRIGger:CMSB:FIENable** <SearchName>,<Frame>, <Field>, <Enabler>

Enables or disables the specific field within the defined frame to be searched for.

**Setting parameters:**
<SearchName>

<Frame>

<Field>

<Enabler> ON | OFF

**Usage:** Setting only

**Manual operation:** See "Frame" on page 779

---

**SEARch:TRIGger:CMSB:FRENable** <SearchName>,<Frame>, <Enabler>

Enables or disables the specific frame to be searched for.

**Setting parameters:**
<SearchName>

<Frame>

<Enabler> ON | OFF

**Usage:** Setting only

**Manual operation:** See "Frame" on page 779

---

**SEARch:TRIGger:CMSB:BIT** <SearchName>,<Frame>, <Field>, <Bit>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:BIT** <SearchName>,<BitState>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:BIT?** <SearchName>

Sets the bit state of a field to be searched that only consists of one bit.

**Suffix:**
<m> *

<n> *

**Parameters:**
<BitState> ONE | ZERO | DC

**ONE**
1

**ZERO**
0

**DC**
"Do not care" (DC) = X
*RST: DC

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame" on page 779

---

**SEARch:TRIGger:CMSB:DMAX** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMAX** <SearchName>,<DataMax>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMAX?** <SearchName>

Sets the end value of a data pattern range if `SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DOPerator` is set to `INRange` or `OORange`.

**Suffix:**
<m> *

<n> *

**Parameters:**
<DataMax> Specifies the name of the generic trigger setting frame.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame" on page 779

---

**SEARch:TRIGger:CMSB:DMIN** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMIN** <SearchName>,<DataMin>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMIN?** <SearchName>

Specifies the data pattern to be searched, or sets the start value of a data pattern range to be searched.

**Suffix:**
<m> *

<n> *

**Parameters:**
<DataMin> Specifies the name of the generic trigger setting frame.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame" on page 779

**SEARch:TRIGger:CMSB:DOPerator** <SearchName>,<Frame>, <Field>, <Operator>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DOPerator**
<SearchName>,<DataOperator>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DOPerator?** <SearchName>

Sets the operator to set a specific data pattern to be searched in the selected field of the selected frame.

**Suffix:**
<m> *

<n> *

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one endpoint value to be set using `SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of endpoint values to be set using `SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMIN` and `SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame" on page 779

**SEARch:TRIGger:CMSB:FRAMe<m>:ENABle** <SearchName>,<Enable>
**SEARch:TRIGger:CMSB:FRAMe<m>:ENABle?** <SearchName>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:ENABle**
<SearchName>,<CondEnabler>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:ENABle?** <SearchName>

Enables or disables the checking condition for searching a specific data pattern in the selected field of the selected frame.

**Suffix:**
<m> *

<n> *

**Parameters:**
<CondEnabler> ON | OFF

**ON**
Checking condition enabled

**OFF**
Checking condition disabled
*RST: OFF

**Parameters for setting and query:**
<SearchName>

---

**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMAX** <SearchName>,<IndexMax>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMAX?** <SearchName>
**SEARch:TRIGger:CMSB:IMAX** <SearchName>,<Frame>, <Field>, <Data>

Sets the end value of the bit index range (data position range), if `SEARch:TRIGger:CMSB:IOPerator` is set to `INRange`.

**Setting parameters:**
<SearchName>

<Frame>

<Field>

<Data>

**Usage:** Setting only

**Manual operation:** See "Frame" on page 779

---

**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMIN** <SearchName>,<IndexMin>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMIN?** <SearchName>
**SEARch:TRIGger:CMSB:IMIN** <SearchName>,<Frame>, <Field>, <Data>

Sets the bit index (data position).

**Setting parameters:**
<SearchName>

<Frame>

<Field>

<Data>

**Usage:** Setting only

**Manual operation:** See "Frame" on page 779

---

**SEARch:TRIGger:CMSB:IOPerator** <SearchName>,<Frame>, <Field>, <Operator>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IOPerator**
<SearchName>,<IndexOperator>
**SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IOPerator?** <SearchName>

Sets the operator for the index for searching in the selected field of the selected frame.

**Suffix:**
<m> *

<n> *

**Parameters:**

<IndexOperator> EQUal | INRange | RANGe

**EQUal**
This condition requires one endpoint value to be set using `SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMIN`.

**INRange | RANGe**
This condition requires a range of endpoint values to be set using `SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMIN` and `SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMAX`.

*RST: INRange

**Parameters for setting and query:**

<SearchName>

**Manual operation:** See "Frame" on page 779

### 24.17.16.6 Search results

SEARch:RESult:CMSB<m>:FCOunt? ..... 2082
SEARch:RESult:CMSB<m>:FRAMe<n>:CCOunt? ..... 2083
SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:NAME? ..... 2083
SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:STARt? ..... 2083
SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:STATe? ..... 2084
SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:STOP? ..... 2084
SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:VALue? ..... 2085
SEARch:RESult:CMSB<m>:FRAMe<n>:CONE? ..... 2085
SEARch:RESult:CMSB<m>:FRAMe<n>:CTHRee? ..... 2086
SEARch:RESult:CMSB<m>:FRAMe<n>:CTWO? ..... 2086
SEARch:RESult:CMSB<m>:FRAMe<n>:STARt? ..... 2086
SEARch:RESult:CMSB<m>:FRAMe<n>:STATe? ..... 2087
SEARch:RESult:CMSB<m>:FRAMe<n>:STOP? ..... 2087
SEARch:RESult:CMSB<m>:FRAMe<n>:TYPE? ..... 2088

---

**SEARch:RESult:CMSB<m>:FCOunt?** <SearchName>

Returns the count number of decoded result frames in a custom serial bus waveform. Basically, this is the maximum result index <n> when querying results by using `SEARch:RESult:CMSB<m>:FRAMe<n>:XXX`.

**Suffix:**

<m> *

**Query parameters:**

<SearchName>

**Return values:**

<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:CMSB<m>:FRAMe<n>:CCOunt?** <SearchName>

Returns the count number of cell in the frames, within the search result, in a custom serial bus waveform.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<NumWords> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:NAME?** <SearchName>

Returns the name of the specified cell within the search result. Cell names are not necessarily unique.

**Suffix:**
<m> *

<n> *

<o> *

**Query parameters:**
<SearchName>

**Return values:**
<Name>

**Usage:** Query only

---

**SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:STARt?** <SearchName>

Returns the start time of the cell within the search result.

**Suffix:**
<m> *

<n> *

<o> *

**Query parameters:**
<SearchName>

**Return values:**

<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:STATe?** <SearchName>

Returns the state of the cell within the search result.

**Suffix:**

<m> *

<n> *

<o> *

**Query parameters:**

<SearchName>

**Return values:**

<State> OK | LENGth | UNKNown | INComplete | CRC | PARity

**OK**
No error detected

**LENGth**
The length error indicates that the cell ended prematurely. The amount of bits that the software expected (based upon the user's frame description) was not found before the cell was terminated. This might occur because a new frame synchronized, or a gap appeared between the bits.

**UNKNown**
Unknown error

**INComplete**
The cell ended prematurely because it extends past the end of the record.

**CRC**
Checksum error in cyclic redundancy check (error in data)

**PARity**
Parity bit error, indicating a transmission error (only available if a parity is configured)

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:STOP?** <SearchName>

Returns the stop time of the cell within the search result.

**Suffix:**
<m> *
<n> *
<o> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:VALue?** <SearchName>

Data content of the specified cell within the search result.

**Suffix:**
<m> *
<n> *
<o> *

**Query parameters:**
<SearchName>

**Return values:**
<Value>

**Usage:** Query only

---

**SEARch:RESult:CMSB<m>:FRAMe<n>:CONE?** <SearchName>

Returns the 1st field content within the search result.

**Suffix:**
<m> *
<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Custom1>

**Usage:** Query only

**SEARch:RESult:CMSB<m>:FRAMe<n>:CTHRee?** <SearchName>

Returns the 3rd field content.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Custom3>

**Usage:** Query only

**SEARch:RESult:CMSB<m>:FRAMe<n>:CTWO?** <SearchName>

Returns the 2nd field content.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Custom2>

**Usage:** Query only

**SEARch:RESult:CMSB<m>:FRAMe<n>:STARt?** <SearchName>

Returns the start time of the frame within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:CMSB<m>:FRAMe<n>:STATe?** <SearchName>

Returns the overall state of the frame within the search result: either OK or the relevant error condition. R&S RTP-K50 marks each frame with a status that indicates whether the decode succeeded or not.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | LENGth | UNKNown | INComplete | CRC | PARity

**OK**
The frame was decoded normally and conforms to the frame description.

**LENGth**
The length error indicates that the frame ended prematurely, or an array in the frame had too few elements. The amount of bits that the software expected (based upon the user's frame description) was not found before the frame was terminated. This might occur because a new frame synchronized, or a gap appeared between the bits.

**UNKNown**
Unknown error

**INComplete**
The frame ended prematurely because it extends past the end of the record.

**CRC**
Checksum error in cyclic redundancy check (error in data)

*RST: OK

**Usage:** Query only

**SEARch:RESult:CMSB<m>:FRAMe<n>:STOP?** <SearchName>

Returns the stop time of the frame within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:CMSB<m>:FRAMe<n>:TYPE?** <SearchName>

Returns the name of the selected frame (n) from the user defined frame format description, labeled according to BUS<m>:CMSB:FRAMe<n>:TYPE on page 2055.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Type>

**Usage:** Query only

## 24.17.17 8B/10B (option R&S RTP-K52)

### 24.17.17.1 Configuration

In all BUS<m>:EBTB commands, the suffix <m> selects the serial bus.

BUS<m>:EBTB:BITRate....2089
BUS<m>:EBTB:EBTRate?....2089
BUS<m>:EBTB:DIFFerential:SOURce....2089
BUS<m>:EBTB:DIFFerential:THRHigh....2089
BUS<m>:EBTB:DIFFerential:THRLow....2090
BUS<m>:EBTB:DISF....2090
BUS<m>:EBTB:DMINus:SOURce....2090
BUS<m>:EBTB:DMINus:THReshold....2091
BUS<m>:EBTB:DPLus:SOURce....2091
BUS<m>:EBTB:DPLus:THReshold....2091
BUS<m>:EBTB:FCSY....2092
BUS<m>:EBTB:SCOunt?....2092
BUS<m>:EBTB:SCSY....2092
BUS<m>:EBTB:SYNC....2092
BUS<m>:EBTB:TYPE....2093
BUS<m>:EBTB:USCS....2093
BUS<m>:EBTB:BITDetermi....2093
BUS<m>:EBTB:FAUToscale....2093

**BUS<m>:EBTB:BITRate** <Bitrate>

Sets the number of transmitted bits per second.

**Suffix:**
<m> 1..4

**Parameters:**
<Bitrate> Range: 100 to 200E+9
Increment: 10
*RST: 1.25E+9
Default unit: bps

**Manual operation:** See "Bitrate" on page 784

**BUS<m>:EBTB:EBTRate?**

Queries the estimated bitrate value.

**Suffix:**
<m> 1..4

**Return values:**
<EstimatedBitrate> Range: 100 to 100E+9
Increment: 10
*RST: 1.25E+9
Default unit: bps

**Usage:** Query only

**Manual operation:** See "Estimate Bitrate" on page 784

**BUS<m>:EBTB:DIFFerential:SOURce** <SrcDiff>

Selects the source of the provided differential signal, if `BUS<m>:EBTB:TYPE DIFFerential` is set.

**Suffix:**
<m> 1..4

**Parameters:**
<SrcDiff> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 783

**BUS<m>:EBTB:DIFFerential:THRHigh** <ThresholdHigh>

Sets the high threshold of the signal, if `BUS<m>:EBTB:TYPE DIFFerential`.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdHigh> Range: -10 to 10
Increment: 1E-3
*RST: 0.15
Default unit: V

**Manual operation:** See "Threshold High" on page 784

---

**BUS<m>:EBTB:DIFFerential:THRLow** <ThresholdLow>

Sets the low threshold of the signal, if `BUS<m>:EBTB:TYPE DIFFerential`.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdLow> Range: -10 to 10
Increment: 1E-3
*RST: -0.15
Default unit: V

**Manual operation:** See "Threshold Low" on page 784

---

**BUS<m>:EBTB:DISF** <DisplayFormat>

Selects the display format for the results in the decode table and for the results of `BUS<m>:EBTB:SYMBol<n>:DATA?`.

**Suffix:**
<m> 1..4

**Parameters:**
<DisplayFormat> KD | EB | TB
KD: K/D codes
EB: 8-bit pattern
TB: 10-bit pattern
*RST: KD

**Manual operation:** See "Display Format" on page 786

---

**BUS<m>:EBTB:DMINus:SOURce** <SourceDminus>

Selects the D- source of the provided single ended signal, if `BUS<m>:EBTB:TYPE SINGle` is set.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceDminus> NONE | C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

**Usage:** Asynchronous command

**Manual operation:** See "Source D-" on page 783

---

**BUS<m>:EBTB:DMINus:THReshold** <ThresholdDminus>

Sets the low threshold (D-) of the signal, if `BUS<m>:EBTB:TYPE SINGle`.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdDminus>
Range: -10 to 10
Increment: 1E-3
*RST: 0
Default unit: V

**Manual operation:** See "Threshold D-" on page 784

---

**BUS<m>:EBTB:DPLus:SOURce** <SourceDplus>

Selects the D+ source of the provided single-ended signal, if `BUS<m>:EBTB:TYPE SINGle` is set.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceDplus> NONE | C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "Source D+" on page 783

---

**BUS<m>:EBTB:DPLus:THReshold** <ThresholdDplus>

Sets the high threshold (D +) of the signal, if `BUS<m>:EBTB:TYPE SINGle`.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdDplus>
Range: -10 to 10
Increment: 1E-3
*RST: 0
Default unit: V

**Manual operation:** See "Threshold D+" on page 784

**BUS<m>:EBTB:FCSY** <CustSync>

Sets a pattern value, if `BUS<m>:EBTB:SYNC CUS`.

**Suffix:**
<m> 1..4

**Parameters:**
<CustSync>

**Manual operation:** See "Pattern" on page 785

**BUS<m>:EBTB:SCOunt?**

Returns the symbol count for the selected serial bus, i.e. the number of symbols in the present acquisition.

**Suffix:**
<m> 1..4

**Return values:**
<Count> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:EBTB:SCSY** <CustSync2>

Sets the value for the second pattern, if `BUS<m>:EBTB:SYNC CUS` and `BUS<m>:EBTB:USCS ON`.

**Suffix:**
<m> 1..4

**Parameters:**
<CustSync2>

**Manual operation:** See "Pattern 2" on page 785

**BUS<m>:EBTB:SYNC** <CommaSymbol>

Sets the sync symbol, a control symbol used for low level control functions.

**Suffix:**
<m> 1..4

**Parameters:**
<CommaSymbol> K285 | K281 | K287 | CUS
*RST: K285

**Manual operation:** See "Sync symbol" on page 785

**BUS<m>:EBTB:TYPE** <SignalType>

Selects the signal type that is used for the decoding. You can choose between a differential and single-ended signal.

**Suffix:**
<m> 1..4

**Parameters:**
<SignalType> DIFFerential | SINGle
*RST: DIFFerential

**Manual operation:** See "Signal type" on page 783

**BUS<m>:EBTB:USCS** <IsSecondSync>

Selects whether the second pattern is enabled, if `BUS<m>:EBTB:SYNC CUS`.

**Suffix:**
<m> 1..4

**Parameters:**
<IsSecondSync> ON | OFF
*RST: OFF

**Manual operation:** See "Pattern 2" on page 785

**BUS<m>:EBTB:BITDetermi**

Starts a software algorithm for the automatic determination of the bitrate.

**Suffix:**
<m> 1..4

**Usage:** Event
Asynchronous command

**Manual operation:** See "Estimate Bitrate" on page 784

**BUS<m>:EBTB:FAUToscale**

Starts software algorithms for determining the signal threshold levels and bitrate.

**Suffix:**
<m> 1..4

**Usage:** Event
Asynchronous command

### 24.17.17.2 Trigger settings

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:EBTB:DISParityerr ..... 2094
TRIGger<m>:EBTB:DX ..... 2094
TRIGger<m>:EBTB:DY ..... 2094
TRIGger<m>:EBTB:EBPA ..... 2095
TRIGger<m>:EBTB:GLITcherror ..... 2095
TRIGger<m>:EBTB:SSTYpe ..... 2095
TRIGger<m>:EBTB:SYME ..... 2095
TRIGger<m>:EBTB:SYMType ..... 2095
TRIGger<m>:EBTB:TBPA ..... 2096
TRIGger<m>:EBTB:STYPe ..... 2096
TRIGger<m>:EBTB:UNK ..... 2096

**TRIGger<m>:EBTB:DISParityerr** <DispError>

Defines, if a trigger on a disparity error is activated or not.

**Parameters:**

<DispError> ON | OFF
*RST: OFF

**Manual operation:** See "Disparity error" on page 789

**TRIGger<m>:EBTB:DX** <DXSymbol>

Sets thex value of the data character Dx.y to be triggered on, if `TRIGger<m>:EBTB:SYMType` is set to `Dxy`.

**Parameters:**

<DXSymbol> Range: 0 to 31
Increment: 1
*RST: 0

**Manual operation:** See "x value" on page 788

**TRIGger<m>:EBTB:DY** <DYSymbol>

Sets the y value of the data character Dx.y to be triggered on, if `TRIGger<m>:EBTB:SYMType` is set to `Dxy`.

**Parameters:**

<DYSymbol> Range: 0 to 7
Increment: 1
*RST: 0

**Manual operation:** See "y value" on page 788

**TRIGger<m>:EBTB:EBPA** <EBPattern>

Sets the 8-bit pattern to be triggered on, if `TRIGger<m>:EBTB:STYPe` is set to `EB`.

**Parameters:**
<EBPattern>

**Manual operation:** See "8 bit pattern" on page 788

**TRIGger<m>:EBTB:GLITcherror** <GlitchError>

Defines, if a trigger on a glitch error is activated or not.

**Parameters:**
<GlitchError> ON | OFF
*RST: OFF

**Manual operation:** See "Glitching error" on page 789

**TRIGger<m>:EBTB:SSTYpe** <SymbolType>

Selects the symbol type to be triggered on. You can select a single symbol or an expression (defined series of symbols).

**Parameters:**
<SymbolType> SYMBol | EXPRession
*RST: SYMBol

**TRIGger<m>:EBTB:SYME** <Expression>

Selects the format of the symbol to be triggered on, if `TRIGger<m>:EBTB:SSTYpe` is set to `EXPRession`.

**Parameters:**
<Expression>

**Manual operation:** See "Expression" on page 788

**TRIGger<m>:EBTB:SYMType** <SymbolType>

Selects the data character (Dx.y) or control character to bea triggered on. You can specify he value of the data character to be triggered on with `TRIGger<m>:EBTB:DX` and `TRIGger<m>:EBTB:DY`.

**Parameters:**
<SymbolType> DXY | K280 | K281 | K282 | K283 | K284 | K285 | K286 | K287 | K237 | K277 | K297 | K307
*RST: K285

**Manual operation:** See "K/D Codes Symbol" on page 788

**TRIGger<m>:EBTB:TBPA** <TBPattern>

Sets the 10-bit pattern to be triggered on, if `TRIGger<m>:EBTB:STYPe` is set to `TB`.

**Parameters:**
<TBPattern>

**Manual operation:** See "10 bit pattern" on page 788

**TRIGger<m>:EBTB:STYPe** <Type>

Selects the type of condition to be triggered on for 8b10b.

**Parameters:**
<Type> KD | EB | TB | EXPRession | ERRor
`KD`: K/D symbol
`EB`: 8-bit pattern
`TB`: 10-bit pattern
*RST: KD

**Manual operation:** See "Type 8b/10b" on page 787

**TRIGger<m>:EBTB:UNK** <UnkError>

Defines, if a trigger on a unknown symbol error is activated or not.

**Parameters:**
<UnkError> ON | OFF
*RST: OFF

**Manual operation:** See "Unknown symbol error" on page 789

### 24.17.17.3 Decode results

In all `BUS<m>:EBTB:SYMBol<n>` commands, the suffix <m> selects the serial bus and the suffix <n> selects the symbol in the decode table.

BUS<m>:EBTB:SYMBol<n>:DATA? ..... 2096
BUS<m>:EBTB:SYMBol<n>:STARt? ..... 2097
BUS<m>:EBTB:SYMBol<n>:STATus? ..... 2097
BUS<m>:EBTB:SYMBol<n>:STOP? ..... 2098
BUS<m>:EBTB:SYMBol<n>:BYTE? ..... 2098
BUS<m>:EBTB:SYMBol<n>:KDCode? ..... 2098
BUS<m>:EBTB:SYMBol<n>:TENBit? ..... 2099

**BUS<m>:EBTB:SYMBol<n>:DATA?**

Returns the data of the specified symbol.

The format is determined by the remote command `BUS<m>:EBTB:DISF`.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<Data>

**Example:**
```
BUS<m>:EBTB:DISF EB
BUS:EBTB:SYMBol5:DATA?
<-- BC+
```

**Example:**
```
BUS<m>:EBTB:DISF KD
BUS:EBTB:SYMBol5:DATA?
<-- K28.5+
```

**Example:**
```
BUS<m>:EBTB:DISF TB
BUS:EBTB:SYMBol5:DATA?
<-- 305
```

**Usage:** Query only

---

**BUS<m>:EBTB:SYMBol<n>:STARt?**

Returns the start time of the specified symbol.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<Start>
Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:EBTB:SYMBol<n>:STATus?**

Returns the status of the specified symbol. It can show what kind of error was detected or no error.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<State> OK | UNK | DISPARITY | DISParity | GLITCH | GLITch

**OK**
There are no errors in the state of the specified symbol.

**UNK**
An unknown error was detected for the specified symbol.

**DISPARITY = DISParity**
A disparity error was detected for the specified symbol.

**GLITCH = GLITch**
A glitch error was detected for the specified symbol.

*RST: OK

**Usage:** Query only

---

**BUS<m>:EBTB:SYMBol<n>:STOP?**

Returns the stop time of the specified symbol.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:EBTB:SYMBol<n>:BYTE?**

Returns the symbol byte value.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Byte> Range: 0 to 0
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:EBTB:SYMBol<n>:KDCode?**

Returns the K/D code.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<KDCode>

**Usage:** Query only

**BUS<m>:EBTB:SYMBol<n>:TENBit?**

Returns the 10-bit symbol.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<TenBit>

**Usage:** Query only

### 24.17.17.4 Search settings

SEARch:TRIGger:EBTB:DISParityerr....2099
SEARch:TRIGger:EBTB:DX....2099
SEARch:TRIGger:EBTB:DY....2100
SEARch:TRIGger:EBTB:EBPA....2100
SEARch:TRIGger:EBTB:GLITcherror....2100
SEARch:TRIGger:EBTB:STYPe....2100
SEARch:TRIGger:EBTB:SYME....2101
SEARch:TRIGger:EBTB:SYMType....2101
SEARch:TRIGger:EBTB:TBPA....2101
SEARch:TRIGger:EBTB:UNK....2102

**SEARch:TRIGger:EBTB:DISParityerr** <SearchName>,<DispError>
**SEARch:TRIGger:EBTB:DISParityerr?** <SearchName>

Defines, if a search for any disparity error is activated or not.

**Parameters:**
<DispError> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Disparity error" on page 793

**SEARch:TRIGger:EBTB:DX** <SearchName>,<DXSymbol>
**SEARch:TRIGger:EBTB:DX?** <SearchName>

Sets the x value of the data character Dx.y to be searched for, if `SEARch:TRIGger:EBTB:SYMType` is set to `Dxy`.

**Parameters:**
<DXSymbol> Range: 0 to 31
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "x value" on page 792

---

**SEARch:TRIGger:EBTB:DY** <SearchName>,<DYSymbol>
**SEARch:TRIGger:EBTB:DY?** <SearchName>

Sets the y value of the data character Dx.y to be searched for, if `SEARch:TRIGger:EBTB:SYMType` is set to `Dxy`.

**Parameters:**
<DYSymbol> Range: 0 to 7
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "y value" on page 792

---

**SEARch:TRIGger:EBTB:EBPA** <SearchName>,<EBPattern>
**SEARch:TRIGger:EBTB:EBPA?** <SearchName>

Sets the 8-bit pattern to be searched for.

**Parameters:**
<EBPattern>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "8b Pattern" on page 793

---

**SEARch:TRIGger:EBTB:GLITcherror** <SearchName>,<GlitchError>
**SEARch:TRIGger:EBTB:GLITcherror?** <SearchName>

Defines, if a search for any glitch error is activated or not.

**Parameters:**
<GlitchError> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Glitching error" on page 793

---

**SEARch:TRIGger:EBTB:STYPe** <SearchName>,<Type>
**SEARch:TRIGger:EBTB:STYPe?** <SearchName>

Selects the search type.

**Parameters:**
<Type> KD | EB | TB | EXPRession | ERRor
`KD`: K/D symbol
`EB`: 8-bit pattern
`TB`: 10-bit pattern
*RST: KD

**Parameters for setting and query:**
<SearchName>

---

**SEARch:TRIGger:EBTB:SYME** <SearchName>,<Expression>
**SEARch:TRIGger:EBTB:SYME?** <SearchName>

Selects the format of the symbol to be searched for, if `SEARch:TRIGger:EBTB:STYPe` is set to `EXPRession`.

**Parameters:**
<Expression>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Expression" on page 793

---

**SEARch:TRIGger:EBTB:SYMType** <SearchName>,<SymbolType>
**SEARch:TRIGger:EBTB:SYMType?** <SearchName>

Selects the data character (Dx.y) or control character to bea searched for. You can specify he value of the data character to be searched for with `SEARch:TRIGger:EBTB:DX` and `SEARch:TRIGger:EBTB:DY`.

**Parameters:**
<SymbolType> DXY | K280 | K281 | K282 | K283 | K284 | K285 | K286 | K287 | K237 | K277 | K297 | K307
*RST: K285

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "K/D Codes Symbol" on page 792

---

**SEARch:TRIGger:EBTB:TBPA** <SearchName>,<TBPattern>
**SEARch:TRIGger:EBTB:TBPA?** <SearchName>

Sets the 10-bit pattern to be searched for, if `SEARch:TRIGger:EBTB:STYPe` is set to `TB`.

**Parameters:**
<TBPattern>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "10b Pattern" on page 793

---

**SEARch:TRIGger:EBTB:UNK** <SearchName>,<UnkError>
**SEARch:TRIGger:EBTB:UNK?** <SearchName>

Defines, if a search for any unknown symbol error is activated or not.

**Parameters:**
<UnkError> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Unknown symbol error" on page 793

### 24.17.17.5 Search results

In all `SEARch:RESult:EBTB:SYMBol<m>` commands, the suffix <m> selects the frame number in the list of search results.

SEARch:RESult:EBTB:SCOunt? 2102
SEARch:RESult:EBTB:SYMBol<m>:DATA? 2102
SEARch:RESult:EBTB:SYMBol<m>:STARt? 2103
SEARch:RESult:EBTB:SYMBol<m>:STATus? 2103
SEARch:RESult:EBTB:SYMBol<m>:STOP? 2103

---

**SEARch:RESult:EBTB:SCOunt?** <SearchName>

Returns the symbol count, i.e. the number of symbols found in the specified search result.

**Query parameters:**
<SearchName>

**Return values:**
<Count> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:EBTB:SYMBol<m>:DATA?** <SearchName>

Returns the data of the specified symbol.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

---

**SEARch:RESult:EBTB:SYMBol<m>:STARt?** <SearchName>

Returns the start time of the specified symbolwithin the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start>
Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:EBTB:SYMBol<m>:STATus?** <SearchName>

Returns the status of the specified symbol within the search result. It can show what kind of error was detected or no error.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | UNK | DISPARITY | DISParity | GLITCH | GLITch

**OK**
There are no errors in the state of the specified symbol.

**UNK**
An unknown error was detected for the specified symbol.

**DISPARITY = DISParity**
A disparity error was detected for the specified symbol.

**GLITCH = GLITch**
A glitch error was detected for the specified symbol.

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:EBTB:SYMBol<m>:STOP?** <SearchName>

Returns the stop time of the specified symbol within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

## 24.17.18 MDIO (option R&S RTP-K55)


- Configuration....2104
- Trigger....2107
- Decode results....2109
- Search settings....2113
- Search results....2116


### 24.17.18.1 Configuration

In all `BUS<m>:MDIO` commands, the suffix <m> selects the serial bus.


BUS<m>:MDIO:CLOCk:SOURce....2104
BUS<m>:MDIO:DATA:SOURce....2105
BUS<m>:MDIO:CLOCk:THReshold:HIGH....2105
BUS<m>:MDIO:CLOCk:THReshold:LOW....2105
BUS<m>:MDIO:DATA:THReshold:HIGH....2106
BUS<m>:MDIO:DATA:THReshold:LOW....2106
BUS<m>:MDIO:PRESet....2106
BUS<m>:MDIO:COUPling....2107


---

**BUS<m>:MDIO:CLOCk:SOURce** <SourceClock>

Selects the source for the clock line (management data clock, MDC). Permitted selections are the analog channels "C1"–"C4" and the digital channels "D0"–"D15".

**Suffix:**
<m> 1..4

**Parameters:**

<SourceClock> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

Digital and analog channels cannot be used at the same time for data and clock lines. For triggering on a serial bus, analog or digital input channels are required.
See Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C2W1

**Manual operation:** See "Source" on page 797

---

**BUS<m>:MDIO:DATA:SOURce** <SourceData>

Selects the source for the data signal. Permitted selections are the analog channels "C1"–"C4" and the digital channels "D0"–"D15", but not the same as for "Clock".

**Suffix:**

<m> 1..4

**Parameters:**

<SourceData> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

Digital and analog channels cannot be used at the same time for data and clock lines. For triggering on a serial bus, analog or digital input channels are required.
See Chapter 24.4.2, "Waveform parameter", on page 1193

*RST: C1W1

**Manual operation:** See "Source" on page 797

---

**BUS<m>:MDIO:CLOCk:THReshold:HIGH** <ThresClkHigh>

Defines the upper threshold level for the clock signal.

**Suffix:**

<m> 1..4

**Parameters:**

<ThresClkHigh> Range: -5 to 5
Increment: 0.1
*RST: 2
Default unit: V

**Manual operation:** See "Thresholds" on page 797

---

**BUS<m>:MDIO:CLOCk:THReshold:LOW** <ThresClkLow>

Defines the lower threshold level for the clock signal.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresClkLow> Range: -5 to 5
Increment: 0.1
*RST: 0.8
Default unit: V

**Manual operation:** See "Thresholds" on page 797

---

**BUS<m>:MDIO:DATA:THReshold:HIGH** <ThresDatHigh>

Defines the upper threshold level for the data signal.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresDatHigh> Range: -5 to 5
Increment: 0.1
*RST: 2
Default unit: V

**Manual operation:** See "Thresholds" on page 797

---

**BUS<m>:MDIO:DATA:THReshold:LOW** <ThresDatLow>

Defines the lower threshold level for the data signal.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresDatLow> Range: -5 to 5
Increment: 0.1
*RST: 0.8
Default unit: V

**Manual operation:** See "Thresholds" on page 797

---

**BUS<m>:MDIO:PRESet** <Preset>

Selects the default threshold settings according to the Ethernet standard: 2.0 V and 0.8 V.

**Suffix:**
<m> 1..4

**Parameters:**
<Preset> DEFault | MANual
*RST: DEFault

**Manual operation:** See "Thresholds" on page 797

**BUS<m>:MDIO:COUPling** <ThresCpl>

Overwrites the data thresholds with the clock thresholds.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresCpl> ON | OFF
*RST: ON

**Manual operation:** See "Thresholds" on page 797

### 24.17.18.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:MDIO:TYPE ..... 2107
TRIGger<m>:MDIO:ST ..... 2108
TRIGger<m>:MDIO:FRAMetype ..... 2108
TRIGger<m>:MDIO:PHYS ..... 2109
TRIGger<m>:MDIO:REGI ..... 2109
TRIGger<m>:MDIO:DATA ..... 2109

**TRIGger<m>:MDIO:TYPE** <Type>

Selects the trigger type for MDIO analysis.

**Parameters:**
<Type> STARt | STOP | DATA

**STARt**
Sets the trigger to the start of frame (SOF) field. The start of frame condition and the trigger instant is the end of the preamble. Trigger pattern: preamble (32 bits "1")

**STOP**
Sets the trigger to the end of frame (EOF) field. The trigger instant is after the last data bit. Trigger pattern: preamble (32 bits "1") + 32 bits "X"

**DATA**
Sets the trigger to the data field. For more information on the data condition, see MDIO in the Ethernet standard. The trigger instant is at the end of the frame after the last data bit, as indicated in the GUI. Trigger pattern: preamble (32 bits "1") + 2 bits "ST" (Start of Frame Code) + 2 bits "OP" (Frame Type Code, or "OpCode") + 5 bits "PHYAD/PRTAD" (Physical Layer Entity Address / Port Address) + 5 bits "REGAD/DEVAD" (Register Address / Device Address) + 2 bits "TA" (turnaround time, X bits) + 16 bits "DATA/ADDRESS"

*RST: STARt

**Manual operation:** See "MDIO Trigger Type" on page 801

---

**TRIGger<m>:MDIO:ST** <StartCode>

Selects the start of frame code of the frame pattern; available only in trigger type "Data".

Note that Clause 22 is coded by "01", while Clause 45 is coded by "00", thus the lower Clause number is represented by the higher parameter value.

**Parameters:**

<StartCode> ST00 | ST01 | ST0X

**ST00**
Clause 45

**ST01**
Clause 22

**ST0X**
Any permissible start pattern

*RST: ST0X

**Manual operation:** See "ST" on page 802

---

**TRIGger<m>:MDIO:FRAMetype** <FrameType>

Selects the Type of Frame code (or OP code, OpCode, operation code); available only in trigger type "Data".

Note that the same OpCode may have different meanings in Clause 22 and Clause 45.

**Parameters:**

<FrameType> OP00 | OP01 | OP10 | OP11 | OPXX

**OP00**
Address frame (in Clause 45, only)

**OP01**
Write frame (in Clause 22 or Clause 45)

**OP10**
Read frame (in Clause 22) or Post-Read increment address frame (in Clause 45)

**OP11**
Read frame (in Clause 45)

**OPXX**
Any frame type

*RST: OPXX
Note that the user interface shows interpretations of the numerical OpCode values corresponding to Clause 45. Clause 22 is not represented by this interpretation.

**Manual operation:** See "OP" on page 802

---

**TRIGger<m>:MDIO:PHYS** <PhyAddr>

Sets the physical address (in Clause 22) or port address (in Clause 45) of the frame pattern (5 bits); available only in trigger type "Data".

**Parameters:**
<PhyAddr>

**Manual operation:** See "PHYAD/PRTAD" on page 802

---

**TRIGger<m>:MDIO:REGI** <RegAddr>

Sets the register address (in Clause 22) or device address (in Clause 45) of the frame pattern (5 bits); available only in trigger type "Data".

**Parameters:**
<RegAddr>

**Manual operation:** See "REGAD/DEVAD" on page 803

---

**TRIGger<m>:MDIO:DATA** <Data>

Defines the 16-bit payload data pattern (both in Clause 22 or Clause 45) or the address pattern (in Clause 45, only) to trigger for; available only in trigger type "Data".

**Parameters:**
<Data>

**Manual operation:** See "DATA/ADDRESS" on page 803

### 24.17.18.3 Decode results

In all `BUS<m>:MDIO:WORD<n>` commands, the suffix <m> selects the serial bus and the suffix <n> selects the word number in the decode table.

BUS<m>:MDIO:WCOunt? ..... 2110
BUS<m>:MDIO:WORD<n>:DATA? ..... 2110
BUS<m>:MDIO:WORD<n>:PHYS? ..... 2110
BUS<m>:MDIO:WORD<n>:REGI? ..... 2111
BUS<m>:MDIO:WORD<n>:ST? ..... 2111

BUS<m>:MDIO:WORD<n>:STARt?....2111
BUS<m>:MDIO:WORD<n>:STATe?....2112
BUS<m>:MDIO:WORD<n>:STOP?....2112
BUS<m>:MDIO:WORD<n>:SYMBol?....2113
BUS<m>:MDIO:WORD<n>:TYPE?....2113

**BUS<m>:MDIO:WCOunt?**

Returns the word count for the selected serial bus, i.e. the number of words in the present acquisition.

**Suffix:**
<m> 1..4

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:MDIO:WORD<n>:DATA?**

Returns the 16-bit payload data field content (in Clause 22 or Clause 45), or the 16-bit address field content (in Clause 45, only) in the present acquisition of the selected word and the selected serial bus. The most significant bit (MSB) is transmitted first.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Data>

**Usage:** Query only

**BUS<m>:MDIO:WORD<n>:PHYS?**

Returns the 5-bit address field content (PHYAD/PRTAD) in the present acquisition of the selected word and the selected serial bus.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<PhyAd> Range: 0 to 32
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:MDIO:WORD<n>:REGI?**

Returns the 5-bit register or device address field content (REGAD/DEVAD) in the present acquisition of the selected word and the selected serial bus.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<RegAd> Range: 0 to 32
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:MDIO:WORD<n>:ST?**

Returns the Start Code (= start of frame code) in the present acquisition of the selected word and the selected serial bus.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<StartCode> Range: 0 to 3
Increment: 1
*RST: 0
The parameter value "0" represents Clause 45, and "1" stands for Clause 22.
The values "2" and "3" do not correspond with any legal parameters according to the standard, but they can be searched for.

**Usage:** Query only

**BUS<m>:MDIO:WORD<n>:STARt?**

Returns the start time of the frame in the selected word of the selected serial bus.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:MDIO:WORD<n>:STATe?**

Returns the state of the frame in the present acquisition of the selected serial bus.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<State> OK | UNSYN | UNSYncronized | OPCO | TA_ERROR | INComplete | SHORT | SHORt

**OK**
No error detected

**UNSYN = UNSYncronized**
UNSYncronized happens when bits are found, but they are not correlated with any synchronization sequence. We don't know what these bits are, but they are there. These bits receive a flag, but they are not decoded.

**OPCO**
OPcode Error

**TA_ERROR**
Tirnaround time error

**INComplete**
Incomplete Frame

**SHORT = SHORt**
Length Error

*RST: OK

**Usage:** Query only

**BUS<m>:MDIO:WORD<n>:STOP?**

Returns the stop time of the frame from the selected word within the search result.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:MDIO:WORD<n>:SYMBol?**

Returns a textual translation (called Register Name) of the PHY or port address label in the present acquisition of the selected word and the selected serial bus.

The translation is defined in the label list.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Translation>

**Usage:** Query only

**BUS<m>:MDIO:WORD<n>:TYPE?**

Returns the OpCode (= operation code or frame type) in the present acquisition of the selected word and the selected serial bus.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<FrameType> Range: 0 to 3
Increment: 1
*RST: 0
The parameter value "0" represents Clause 45, and "1" stands for Clause 22.
The values "2" and "3" do not correspond with any legal parameters according to the standard, but they can be searched for.

**Usage:** Query only

### 24.17.18.4 Search settings

SEARch:TRIGger:MDIO:DATA....2113
SEARch:TRIGger:MDIO:FRAMetype....2114
SEARch:TRIGger:MDIO:PHYS....2114
SEARch:TRIGger:MDIO:REGI....2115
SEARch:TRIGger:MDIO:ST....2115
SEARch:TRIGger:MDIO:TYPE....2115

**SEARch:TRIGger:MDIO:DATA** <SearchName>,<Data>
**SEARch:TRIGger:MDIO:DATA?** <SearchName>

Allows to define the 16-bit payload data pattern (both in Clause 22 or Clause 45) or the address pattern (in Clause 45, only) to search for; available only in search criteria type "Data".

**Parameters:**
<Data>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "DATA/ADDRESS" on page 806

---

**SEARch:TRIGger:MDIO:FRAMetype** <SearchName>,<FrameType>
**SEARch:TRIGger:MDIO:FRAMetype?** <SearchName>

Allows to select the Type of Frame code (or OP code, OpCode, operation code); available only in search criteria type "Data".

Note that the same OpCode may have different meanings in Clause 22 and Clause 45.

**Parameters:**
<FrameType> OP00 | OP01 | OP10 | OP11 | OPXX

**OP00**
Address frame (in Clause 45, only)

**OP01**
Write frame (in Clause 22 or Clause 45)

**OP10**
Read frame (in Clause 22) or Post-Read increment address frame (in Clause 45)

**OP11**
Read frame (in Clause 45)

**OPXX**
Any frame type

*RST: OPXX
Note that the user interface shows interpretations of the numerical OpCode values corresponding to Clause 45. Clause 22 is not represented by this interpretation.
Also, note that OPXX will never be a result of decoding, but it is still an option for triggering.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "OP" on page 806

---

**SEARch:TRIGger:MDIO:PHYS** <SearchName>,<PhyAddr>
**SEARch:TRIGger:MDIO:PHYS?** <SearchName>

Allows to set the physical address (in Clause 22) or port address (in Clause 45) of the frame pattern (5 bits); available only in search criteria type "Data".

**Parameters:**
<PhyAddr>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "PHYAD/PRTAD" on page 806

---

**SEARch:TRIGger:MDIO:REGI** <SearchName>,<RegAddr>
**SEARch:TRIGger:MDIO:REGI?** <SearchName>

Allows to set the register address (in Clause 22) or device address (in Clause 45) of the frame pattern (5 bits); available only in search criteria type "Data".

**Parameters:**
<RegAddr>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "REGAD/DEVAD" on page 806

---

**SEARch:TRIGger:MDIO:ST** <SearchName>,<StartCode>
**SEARch:TRIGger:MDIO:ST?** <SearchName>

Allows to select the start of frame code of the frame pattern; available only in search criteria type "Data".

**Parameters:**
<StartCode> ST00 | ST01 | ST0X

**ST00**
Clause 45

**ST01**
Clause 22

**ST0X**
Any permissible start pattern

*RST: ST0X

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "ST" on page 806

---

**SEARch:TRIGger:MDIO:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:MDIO:TYPE?** <SearchName>

Selects the event type to search for.

**Parameters:**
<Type> STARt | STOP | DATA

**STARt**
Searches for the start of frame (SOF).

**STARt**
Searches for the end of frame (EOF).

**DATA**
Allows to specify a payload data pattern (both in Clause 22 or Clause 45) or an address pattern (in Clause 45, only) to search for.

*RST: STARt

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search types" on page 805

### 24.17.18.5 Search results

In all `SEARch:RESult:MDIO:WORD<m>` commands, the suffix <m> selects the packet number in the list of search results.

SEARch:RESult:MDIO:WORD<m>:DATA? 2116
SEARch:RESult:MDIO:WORD<m>:PHYS? 2116
SEARch:RESult:MDIO:WORD<m>:REGI? 2117
SEARch:RESult:MDIO:WORD<m>:ST? 2117
SEARch:RESult:MDIO:WORD<m>:STARt? 2117
SEARch:RESult:MDIO:WORD<m>:STATe? 2118
SEARch:RESult:MDIO:WORD<m>:STOP? 2118
SEARch:RESult:MDIO:WORD<m>:SYMBol? 2119
SEARch:RESult:MDIO:WORD<m>:TYPE? 2119
SEARch:RESult:MDIO:WCOunt? 2119

---

**SEARch:RESult:MDIO:WORD<m>:DATA?** <SearchName>

Returns the 16-bit payload data field content (in Clause 22 or Clause 45), or the 16-bit address field content (in Clause 45, only) from the selected word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

---

**SEARch:RESult:MDIO:WORD<m>:PHYS?** <SearchName>

Returns the 5-bit address field content (PHYAD/PRTAD) from the selected word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<PhyAd> Range: 0 to 32
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:MDIO:WORD<m>:REGI?** <SearchName>

Returns the 5-bit register or device address field content (REGAD/DEVAD) from the selected word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<RegAd> Range: 0 to 32
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:MDIO:WORD<m>:ST?** <SearchName>

Returns the start of frame code from the selected word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<StartCode> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:MDIO:WORD<m>:STARt?** <SearchName>

Returns the start time of the frame from the selected word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**

<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:MDIO:WORD<m>:STATe?** <SearchName>

Returns the state of the frame from the selected word within the search result.

**Suffix:**

<m> *

**Query parameters:**

<SearchName>

**Return values:**

<State> OK | UNSYN | UNSYncronized | OPCO | TA_ERROR | INComplete | SHORT | SHORt

**OK**
No error detected

**UNSYN = UNSYncronized**
UNSYncronized happens when bits are found, but they are not correlated with any synchronization sequence. We don't know what these bits are, but they are there. These bits receive a flag, but they are not decoded.

**OPCO**
OPcode error

**TA_ERROR**
turnaround time error

**INComplete**
Incomplete Frame

**SHORT = SHORt**
Length Error

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:MDIO:WORD<m>:STOP?** <SearchName>

Returns the stop time of the frame from the selected word within the search result.

**Suffix:**

<m> *

**Query parameters:**

<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:MDIO:WORD<m>:SYMBol?** <SearchName>

Returns a textual translation (called Register Name) of the PHY or port address label from the selected word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Translation>

**Usage:** Query only

---

**SEARch:RESult:MDIO:WORD<m>:TYPE?** <SearchName>

Returns the frame type (= operation code or OpCode) for the selected word within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameType> Range: 0 to 3
Increment: 1
*RST: 0
Note that the user interface shows interpretations of the numerical OpCode values corresponding to Clause 45: "0" (= [bin]00) represents Address, "1" (= [bin]01" represents Write, "2" (= [bin]10) represents Post Read, "3" (= [bin]11) represents Read. Clause 22 is not represented by this interpretation.

**Usage:** Query only

---

**SEARch:RESult:MDIO:WCOunt?** <SearchName>

Returns the word count within the search result.

**Query parameters:**
<SearchName>

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

## 24.17.19 USB (option R&S RTP-K60)


- Configuration....2120
- Trigger....2122
- Decode results....2135
- Search settings....2141
- Search results....2157


### 24.17.19.1 Configuration

In all `BUS<m>:USB` commands, the suffix <m> selects the serial bus.


BUS<m>:USB:TECHnology....2120
BUS<m>:USB:DPLus:SOURce....2121
BUS<m>:USB:DMINus:SOURce....2121
BUS<m>:USB:DIFFerential:SOURce....2121
BUS<m>:USB:DPLus:THReshold....2121
BUS<m>:USB:DMINus:THReshold....2122
BUS<m>:USB:DIFFerential:THReshold....2122


---

**BUS<m>:USB:TECHnology** <ProtocolType>

Defines the USB protocol technology and transmission speed.

**Suffix:**
<m> 1..4

**Parameters:**
<ProtocolType> LOW | FULL | HIGH

**LOW**
USB low speed protocol (1.5 Mbit/s)

**FULL**
USB full speed protocol (12 Mbit/s)

**HIGH**
USB high speed protocol (480 Mbit/s)

*RST: LOW

**Manual operation:** See "Protocol type" on page 814

**BUS<m>:USB:DPLus:SOURce** <SourceDplus>

Selects the source for the D+ data signal (in USB low speed and USB full speed protocol, only).

**Suffix:**
<m> 1..4

**Parameters:**
<SourceDplus> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "D+, D-" on page 816

**BUS<m>:USB:DMINus:SOURce** <SourceDminus>

Selects the source for the D- data signal (in USB low speed and USB full speed protocol, only).

**Suffix:**
<m> 1..4

**Parameters:**
<SourceDminus> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

*RST: C2W1

**Usage:** Asynchronous command

**Manual operation:** See "D+, D-" on page 816

**BUS<m>:USB:DIFFerential:SOURce** <SrcDiff>

Selects the source for the differential signal in the USB high speed protocol.

**Suffix:**
<m> 1..4

**Parameters:**
<SrcDiff> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "Differential" on page 816

**BUS<m>:USB:DPLus:THReshold** <ThresholdDplus>

Defines the threshold level for the D+ data signal (in USB low speed and USB full speed protocol, only).

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdDplus> Range: -5 to 5
Increment: 0.01
*RST: 1.55
Default unit: V

**Manual operation:** See "Thresholds" on page 817

---

**BUS<m>:USB:DMINus:THReshold** <ThresholdDminus>

Defines the threshold level for the D- data signal (in USB low speed and USB full speed protocol, only).

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdDminus> Range: -5 to 5
Increment: 0.01
*RST: 1.55
Default unit: V

**Manual operation:** See "Thresholds" on page 817

---

**BUS<m>:USB:DIFFerential:THReshold** <ThresDiff>

Defines the threshold level for the differential signal in the USB high speed protocol.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresDiff> Range: -2 to 2
Increment: 0.01
*RST: 0
Default unit: V

**Manual operation:** See "Thresholds" on page 817

### 24.17.19.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:USB:TYPE ..... 2123
TRIGger<m>:USB:ACONdition ..... 2125
TRIGger<m>:USB:AMIN ..... 2125
TRIGger<m>:USB:AMAX ..... 2125
TRIGger<m>:USB:DATA ..... 2125
TRIGger<m>:USB:DCONdition ..... 2126
TRIGger<m>:USB:DPOPerator ..... 2126
TRIGger<m>:USB:DPOSition ..... 2126
TRIGger<m>:USB:ECONdition ..... 2127
TRIGger<m>:USB:EMIN ..... 2127
TRIGger<m>:USB:EMAX ..... 2127
TRIGger<m>:USB:ERRC ..... 2127
TRIGger<m>:USB:FCONdition ..... 2128
TRIGger<m>:USB:FMIN ..... 2128
TRIGger<m>:USB:FMAX ..... 2129
TRIGger<m>:USB:HAND ..... 2129
TRIGger<m>:USB:PATT ..... 2129
TRIGger<m>:USB:PCONdition ..... 2130
TRIGger<m>:USB:PMIN ..... 2130
TRIGger<m>:USB:PMAX ..... 2130
TRIGger<m>:USB:SCONdition ..... 2130
TRIGger<m>:USB:SMIN ..... 2131
TRIGger<m>:USB:SMAX ..... 2131
TRIGger<m>:USB:SPEC ..... 2131
TRIGger<m>:USB:STCO ..... 2132
TRIGger<m>:USB:TCONdition ..... 2132
TRIGger<m>:USB:TMIN ..... 2132
TRIGger<m>:USB:TMAX ..... 2133
TRIGger<m>:USB:TOKen ..... 2133
TRIGger<m>:USB:WADD ..... 2133
TRIGger<m>:USB:WEND ..... 2133
TRIGger<m>:USB:WETCheck ..... 2134
TRIGger<m>:USB:WFRN ..... 2134
TRIGger<m>:USB:WPAY ..... 2134
TRIGger<m>:USB:WPID ..... 2134
TRIGger<m>:USB:WPOR ..... 2134
TRIGger<m>:USB:WSEU ..... 2135
TRIGger<m>:USB:WSTC ..... 2135

**TRIGger<m>:USB:TYPE** <Type>

Selects the trigger type for USB analysis. The available trigger types depend on the activated USB protocol type.

**Parameters:**

<Type> SOP | EOP | RST | SUSPend | RESume | TOKen | DATA | HANDshake | SPECial | ERRCond

**SOP**
Sets the trigger to the SOP (start of packet) field. The start of packet condition is the end of the SYNC field. The trigger instant is the end of the SOP field.

**EOP**
Sets the trigger to the EOP (end of packet) field. Not available for USB High Speed and USB HSIC protocol types. The trigger instant is the beginning of the EOP field.

**RST**
Sets the trigger to the Reset field. Not available for USB High Speed and USB HSIC protocol types. For more information on the reset condition, see the USB standard. The trigger instant is the end of the 10 ms period after the SE0 field.

**SUSPend**
Sets the trigger to the Suspend field. Not available for USB High Speed and USB HSIC protocol types. For more information on the suspend condition, see the USB standard. The trigger instant will be declared after the defined 3 ms timeout.

**RESume**
Sets the trigger to the Resume field. Not available for USB High Speed and USB HSIC protocol types. For more information on the resume condition, see the USB standard. The trigger instant will be declared after the defined 20 ms timeout.

**TOKen**
Sets the trigger to one out of four different token trigger types: OUT, IN, SOF, or SETUP.

**DATA**
Sets the trigger to one out of four different data trigger types: DATA0, DATA1, DATA2, or MDATA.

**HANDshake**
Sets the trigger to one out of four different handshake trigger types: ACK, NAK, STALL, or NYET.

**SPECial**
Sets the trigger to one out of four different Special PID trigger types: PREamble, ERR, SPLIT, or PING.

**ERRCond**
Sets the trigger to one out of seven different error condition trigger types: PID error, CRC5 error, CRC16 error, Bitstuffing error, Unexpected PID error, SE1 error, or Glitching error.

*RST: SOP

**Manual operation:** See "Trigger type USB" on page 818

**TRIGger<m>:USB:ACONdition** <OperatorAddress>

Sets the operator to set a specific address or an address range. The address values are set with `TRIGger<m>:USB:AMIN` and `TRIGger<m>:USB:AMAX`.

**Parameters:**

<OperatorAddress> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one address value to be set using `TRIGger<m>:USB:AMIN`.

**INRange | OORange**
In range, out of range. These conditions require an address range to be set using `TRIGger<m>:USB:AMIN` and `TRIGger<m>:USB:AMAX`.

*RST: OFF

**Manual operation:** See "Address" on page 823

**TRIGger<m>:USB:AMIN** <Address>

Specifies the address, or sets the the start value of an address range. The string parameter does not accept the bit value X (don't care).

**Parameters:**

<Address>

**Manual operation:** See "Address" on page 823

**TRIGger<m>:USB:AMAX** <AddressTo>

Sets the the end value of an address range if `TRIGger<m>:USB:ACONdition` is set to `INRange` or `OORange`. The string parameter does not accept the bit value X (don't care).

**Parameters:**

<AddressTo>

**Manual operation:** See "Address" on page 823

**TRIGger<m>:USB:DATA** <USBDataType>

Sets the trigger to one out of four different payload data types: DATA0, DATA1, DATA2, or MDATA.

**Parameters:**

<USBDataType> D0 | D1 | D2 | MD

**D0**
Sets the trigger to the DATA0 field (even PID).

**D1**
Sets the trigger to the DATA1 field (odd PID).

**D2**
Sets the trigger to the DATA2 field (data packet PID for high-speed, high bandwidth isochronous transaction in a microframe).

**MD**
Sets the trigger to the MDATA field (high-speed data packet PID for split and high bandwidth isochronous transactions).

*RST: D0

**Manual operation:** See "Data" on page 820

---

**TRIGger<m>:USB:DCONdition** <OperatorData>

Sets the operator (equal or unequal) to set a specific payload data pattern.

**Parameters:**
<OperatorData> OFF | ANY | EQUal | NEQual
*RST: EQUal

**Manual operation:** See "Payload" on page 823

---

**TRIGger<m>:USB:DPOPerator** <DataPosOperator>

Sets the operator (any or equal) for the payload data index position.

**Parameters:**
<DataPosOperator> ANY | OFF | EQUal

**ANY = OFF**
The position of the bit pattern within the payload data is not relevant.

**EQUal**
Sets the operator for specifying a special start position for the bit pattern within the payload data.

*RST: ANY

**Manual operation:** See "Payload" on page 823

---

**TRIGger<m>:USB:DPOSition** <DataPosition>

Specifies the position within a payload data packet, in which a special data pattern is to be searched.

**Parameters:**
<DataPosition> Range: 0 to 1024
Increment: 1
*RST: 0

**Manual operation:** See "Payload" on page 823

**TRIGger<m>:USB:ECONdition** <EndpointOptor>

Sets the operator to set a specific endpoint or an endpoint range. The endpoint values are set with `TRIGger<m>:USB:EMIN` and `TRIGger<m>:USB:EMAX`.

**Parameters:**

<EndpointOptor> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one endpoint value to be set using `TRIGger<m>:USB:EMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of endpoint values to be set using `TRIGger<m>:USB:EMIN` and `TRIGger<m>:USB:EMAX`.

*RST: OFF

**Manual operation:** See "End Point" on page 823

---

**TRIGger<m>:USB:EMIN** <Endp>

Specifies the endpoint, or sets the the start value of an endpoint range.

**Parameters:**

<Endp>

**Manual operation:** See "End Point" on page 823

---

**TRIGger<m>:USB:EMAX** <EndpTo>

Sets the the end value of an endpoint range if `TRIGger<m>:USB:ECONdition` is set to `INRange` or `OORange`.

**Parameters:**

<EndpTo>

**Manual operation:** See "End Point" on page 823

---

**TRIGger<m>:USB:ERRC** <ErrorType>

Sets the trigger to an error condition type.

**Parameters:**

<ErrorType> ANY | PIDerror | CRC5error | CRC16error | BTST | UNEXpid | SE1error | GLITcherr

**ANY**
Triggers on any of the errors listed below.

**PIDerror**
Triggers on any packet identifier error.

**CRC5error**
Triggers on any CRC5 error event.

**CRC16error**
Triggers on any CRC16 error event.

**BTST**
Triggers on any bitstuffing error event (erroneous or missing bit stuffing sequence, see USB standard).

**UNEXpid**
Triggers on any unexpected PID error (illegal PID, that is not allowed in USB low speed and USB full speed protocols, especially PID's announcing packets such as SPLIT, DATA2, MDATA, or other noncompliant packets).

**SE1error**
Triggers on the illegal bus state Single Ended 1 (SE1 = both lines high).

**GLITcherr**
Triggers on any glitching error (illegal bit period, see USB standard for the definition of glitching).

*RST: ANY

**Manual operation:** See "Error condition" on page 822

---

**TRIGger<m>:USB:FCONdition** <FrameNoOperator>

Sets the operator to set a specific frame number or a frame number range. The frame number values are set with `TRIGger<m>:USB:FMIN` and `TRIGger<m>:USB:FMAX`.

**Parameters:**

<FrameNoOperator> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one frame number value to be set using `TRIGger<m>:USB:FMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of frame number values to be set using `TRIGger<m>:USB:FMIN` and `TRIGger<m>:USB:FMAX`.

*RST: OFF

**Manual operation:** See "Frame number" on page 824

---

**TRIGger<m>:USB:FMIN** <FrameNumber>

Specifies the frame number, or sets the the start value of a frame number range.

**Parameters:**

<FrameNumber>

**Manual operation:** See "Frame number" on page 824

---

**TRIGger<m>:USB:FMAX** <FrameNumberTo>

Sets the the end value of a frame number range if `TRIGger<m>:USB:FCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<FrameNumberTo>

**Manual operation:** See "Frame number" on page 824

---

**TRIGger<m>:USB:HAND** <HandshakeType>

Sets the trigger to one out of four different handshake types: ACK, NAK, STALl, or NYET.

**Parameters:**
<HandshakeType> ACK | NAK | STALl | NYET

**ACK**
Sets the trigger to the ACK field (acknowledgment of error-free data packet).

**NAK**
Sets the trigger to the NAK field (non-acknowledgment, no successful data transmission).

**STALl**
Sets the trigger to the STALL field (endpoint is halted or a control pipe request is not supported).

**NYET**
Sets the trigger to the NYET field (no response yet from receiver).

*RST: ACK

**Manual operation:** See "Handshake" on page 821

---

**TRIGger<m>:USB:PATT** <PayloadMuster>

Specifies the payload data pattern that is to be searched.

**Parameters:**
<PayloadMuster>

**Manual operation:** See "Payload" on page 823

**TRIGger<m>:USB:PCONdition** <OperatorPort>

Sets the operator to set a specific port number or a port number range. The port number values are set with `TRIGger<m>:USB:PMIN` and `TRIGger<m>:USB:PMAX`.

**Parameters:**

<OperatorPort> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one port number to be set using `TRIGger<m>:USB:PMIN`.

**INRange | OORange**
In range, out of range. These conditions require a port number range to be set using `TRIGger<m>:USB:PMIN` and `TRIGger<m>:USB:PMAX`.

*RST: OFF

**Manual operation:** See "Port" on page 824

**TRIGger<m>:USB:PMIN** <Port>

Specifies the port number, or sets the the start value of a port number range.

**Parameters:**

<Port>

**Manual operation:** See "Port" on page 824

**TRIGger<m>:USB:PMAX** <PortTo>

Sets the the end value of a port number range if `TRIGger<m>:USB:PCONdition` is set to `INRange` or `OORange`.

**Parameters:**

<PortTo>

**Manual operation:** See "Port" on page 824

**TRIGger<m>:USB:SCONdition** <OperatorSEU>

Sets the operator to set a specific SEU or an SEU range. The SEU values are set with `TRIGger<m>:USB:SMIN` and `TRIGger<m>:USB:SMAX`.

For SEU, see "SEU" on page 825.

**Parameters:**

<OperatorSEU> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one SEU value to be set using `TRIGger<m>:USB:SMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of SEU values to be set using `TRIGger<m>:USB:SMIN` and `TRIGger<m>:USB:SMAX`.

*RST: OFF

**Manual operation:** See "SEU" on page 825

---

**TRIGger<m>:USB:SMIN** <SEU>

Specifies the SEU, or sets the the start value of an SEU range.

**Parameters:**
<SEU>

**Manual operation:** See "SEU" on page 825

---

**TRIGger<m>:USB:SMAX** <SEUTo>

Sets the the end value of an SEU range if `TRIGger<m>:USB:SCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<SEUTo>

**Manual operation:** See "SEU" on page 825

---

**TRIGger<m>:USB:SPEC** <USBSpecialType>

Sets the trigger to one out of four different Special PID types: PREamble, ERR, SPLit, or PING.

**Parameters:**
<USBSpecialType> PREamble | ERR | SPLit | PING

**PREamble**
Sets the trigger to the PREamble PID

**ERR**
Sets the trigger to the ERRor PID

**SPLit**
Sets the trigger to the SPLIT PID (in USB high speed transactions)

**PING**
Sets the trigger to the PING PID (in USB high speed transactions, flow control probe for a bulk/control endpoint)

*RST: PREamble

**Manual operation:** See "Special PID" on page 821

---

**TRIGger<m>:USB:STCO** <SC>

Sets the trigger to a specific start-split or complete-split transaction endpoint.

**Parameters:**

<SC> ONE | ZERO | DC

**ONE**
SC = 1 represents a complete-split (CSPLIT) transaction.

**ZERO**
SC = 0 represents a start-split (SSPLIT) transaction.

**DC**
SC = X represents "don't care" (DC)

*RST: DC

**Manual operation:** See "SC" on page 824

---

**TRIGger<m>:USB:TCONdition** <OperatorET>

Sets the operator to set a specific endpoint type (ET) or an ET range. The ET values are set with `TRIGger<m>:USB:TMIN` and `TRIGger<m>:USB:TMAX`.

**Parameters:**

<OperatorET> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**OFF = ANY**
The position of the endpoint type is not relevant.

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one ET value to be set using `TRIGger<m>:USB:TMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of ET values to be set using `TRIGger<m>:USB:SMIN` and `TRIGger<m>:USB:SMAX`.

*RST: OFF

**Manual operation:** See "ET" on page 825

---

**TRIGger<m>:USB:TMIN** <ET>

Specifies the endpoint type, or sets the the start value of an endpoint type range.

**Parameters:**

<ET>

**Manual operation:** See "ET" on page 825

**TRIGger<m>:USB:TMAX** <ETTo>

Sets the the end value of an endpoint type range if `TRIGger<m>:USB:TCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<ETTo>

**Manual operation:** See "ET" on page 825

**TRIGger<m>:USB:TOKen** <USBTokenType>

Sets the trigger to one out of four different token types: OUT, IN, SOF, or SETup.

**Parameters:**
<USBTokenType> OUT | IN | SOF | SETup

**OUT**
Sets the trigger to the OUT token (OUT packet from host to device).

**IN**
Sets the trigger to the IN token (IN packet from device to host).

**SOF**
Sets the trigger to the SOF token (start of frame marker and frame number).

**SETup**
Sets the trigger to the SETUP token (address and endpoint number in OUT transaction for setup to a control pipe).

*RST: OUT

**Manual operation:** See "Token" on page 820

**TRIGger<m>:USB:WADD** <AddressCheck>

Defines, whether the address check shall be activated or not.

**Parameters:**
<AddressCheck> ON | OFF

*RST: OFF

**Manual operation:** See "Address" on page 823

**TRIGger<m>:USB:WEND** <EndpointCheck>

Defines, whether the endpoint check shall be activated or not.

**Parameters:**
<EndpointCheck> ON | OFF

*RST: OFF

**Manual operation:** See "End Point" on page 823

**TRIGger<m>:USB:WETCheck** <WithETCheck>

Defines, whether the Endpoint Type (ET) check shall be activated or not.

**Parameters:**
<WithETCheck> ON | OFF
*RST: OFF

**Manual operation:** See "ET" on page 825

**TRIGger<m>:USB:WFRN** <FrameNoChk>

Defines, whether the frame number check shall be activated or not.

**Parameters:**
<FrameNoChk> ON | OFF
*RST: OFF

**Manual operation:** See "Frame number" on page 824

**TRIGger<m>:USB:WPAY** <PayloadCheck>

Defines, whether the payload data check shall be activated or not.

**Parameters:**
<PayloadCheck> ON | OFF
*RST: OFF

**Manual operation:** See "Payload" on page 823

**TRIGger<m>:USB:WPID** <WithPIDCheck>

Defines, whether the packet ID error check shall be activated or not.

**Parameters:**
<WithPIDCheck> ON | OFF
*RST: OFF

**Manual operation:** See "PID error " on page 823

**TRIGger<m>:USB:WPOR** <WithPortCheck>

Defines, whether the port check shall be activated or not.

**Parameters:**
<WithPortCheck> ON | OFF
*RST: OFF

**Manual operation:** See "Port" on page 824

**TRIGger<m>:USB:WSEU** <WithSEUCheck>

Defines, whether the SEU check shall be activated or not.

For SEU, see "SEU" on page 825.

**Parameters:**
<WithSEUCheck> ON | OFF
*RST: OFF

**Manual operation:** See "SEU" on page 825

**TRIGger<m>:USB:WSTC** <WithSCCheck>

Defines, whether the Start / Complete (SC) check shall be activated or not.

**Parameters:**
<WithSCCheck> ON | OFF
*RST: OFF

**Manual operation:** See "SC" on page 824

### 24.17.19.3 Decode results

In all `BUS<m>:USB:PACKet<n>` commands, the suffix <m> selects the serial bus and the suffix <n> selects the packet number in the decode table.

► Query the status of packet #19: `BUS:USB:PACKet19:STAT?`

The result of this remote command query should be "OK".

BUS<m>:USB:PACKet<n>:PID? ..... 2135
BUS<m>:USB:PACKet<n>:ADDRess? ..... 2137
BUS<m>:USB:PACKet<n>:CRC? ..... 2137
BUS<m>:USB:PACKet<n>:DATA? ..... 2137
BUS<m>:USB:PACKet<n>:ENDPoint? ..... 2138
BUS<m>:USB:PACKet<n>:ET? ..... 2138
BUS<m>:USB:PACKet<n>:FRAMe? ..... 2138
BUS<m>:USB:PACKet<n>:PORT? ..... 2138
BUS<m>:USB:PACKet<n>:SC? ..... 2139
BUS<m>:USB:PACKet<n>:SEU? ..... 2139
BUS<m>:USB:PACKet<n>:STARt? ..... 2139
BUS<m>:USB:PACKet<n>:STATus? ..... 2140
BUS<m>:USB:PACKet<n>:STOP? ..... 2140
BUS<m>:USB:PCOunt? ..... 2140

**BUS<m>:USB:PACKet<n>:PID?**

Returns the packet PID for the selected serial bus and packet number.

**Suffix:**
<m> 1..4

<n> *

**Return values:**

<PID> RES | OUT | ACK | DATA0 | DATa0 | PING | SOF | NYET | DATA2 | DATa2 | SPLIT | SPLit | IN | NAK | DATA1 | DATa1 | PRE | SETUP | SETup | STALL | STALI | MDATA | MDATa | UNK

**RES**
RES = Reserved

**OUT**
OUT Token PID

**IN**
IN Token PID

**SOF**
Start Of Frame PID

**SETUP = SETup**
SETUP PID

**DATA0 = DATa0**
DATA0 PID, even PID

**DATA1 = DATa1**
DATA1 PID, odd PID

**DATA2 = DATa2**
DATA2 PID (only valid in USB high speed and USB HSIC protocols)

**MDATA = MDATa**
MDATA PID (only valid in USB high speed and USB HSIC protocols)

**ACK**
ACKnowledgment PID

**NAK**
Non-AcKnowledgment PID

**STALL = STALI**
STALL PID

**NYET**
Not ready YET (only valid in USB high speed and USB HSIC protocols)

**PRE**
PREamble PID (only valid in USB high speed and USB HSIC protocols)

**SPLIT = SPLit**
SPLIT PID (only valid in USB high speed and USB HSIC protocols)

**PING**
PING PID (only valid in USB high speed and USB HSIC protocols)

**UNK**
UNK = Unknown PID

*RST: RES

**Usage:** Query only

---

**BUS<m>:USB:PACKet<n>:ADDRess?**

Returns the packet address for the selected serial bus and packet number.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Addr> Range: 0 to 127
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:USB:PACKet<n>:CRC?**

Returns the packet CRC (Cyclic Redundancy Code) for the selected serial bus and packet number.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<CRC> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:USB:PACKet<n>:DATA?**

Returns the payload data from the packet with the selected packet number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *

**Parameters:**
<Data> Comma-separated sequence of integer values (N, D1, D2,…, DN). N is the number of bytes in the packet. and D1…DN are the values of the bytes.

**Example:**
```
BUS:USB:PACKet4:DATA?
<-- 6,18,52,86,120,154,188
```

**Usage:** Query only

---

**BUS<m>:USB:PACKet<n>:ENDPoint?**

Returns the endpoint for the selected serial bus and packet number.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Endp> Range: 0 to 15
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:USB:PACKet<n>:ET?**

Returns the endpoint type (ET) for the selected serial bus and packet number.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<ET> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:USB:PACKet<n>:FRAMe?**

Returns the frame number for the selected serial bus and packet number.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<FrameNo> Range: 0 to 2047
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:USB:PACKet<n>:PORT?**

Returns the port number for the selected serial bus and packet number.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<Port> Range: 0 to 127
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:USB:PACKet<n>:SC?**

Returns the Start- / Complete-split transaction (SSPLIT / CSPLIT) flag bits for the selected serial bus and packet number.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<SC>

**Usage:** Query only

---

**BUS<m>:USB:PACKet<n>:SEU?**

Returns the SEU values for the selected serial bus and packet number.

For SEU, see "SEU" on page 825.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<SEU> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:USB:PACKet<n>:STARt?**

Returns the start position of the packet with the selected packet number (for the selected serial bus).

**Suffix:**
<m> 1..4
<n> *

**Return values:**

<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:USB:PACKet<n>:STATus?**

Returns the status of the packet with the selected packet number (for the selected serial bus).

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<State> OK | PID | CRC | BTST | GLITCH | GLITch | BYTE
*RST: OK

**Usage:** Query only

---

**BUS<m>:USB:PACKet<n>:STOP?**

Returns the stop time of the packet with the selected packet number (for the selected serial bus).

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:USB:PCOunt?**

Returns the packet count for the selected serial bus, i.e. the number of packets in the present acquisition.

**Suffix:**

<m> 1..4

**Return values:**

<Count> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

### 24.17.19.4 Search settings

SEARch:TRIGger:USB:ACONdition....2142
SEARch:TRIGger:USB:AMIN....2142
SEARch:TRIGger:USB:AMAX....2142
SEARch:TRIGger:USB:BITSterror....2143
SEARch:TRIGger:USB:CRC16error....2143
SEARch:TRIGger:USB:CRC5error....2143
SEARch:TRIGger:USB:DATA....2144
SEARch:TRIGger:USB:DCONdition....2144
SEARch:TRIGger:USB:DPOPerator....2144
SEARch:TRIGger:USB:DPOSition....2145
SEARch:TRIGger:USB:ECONdition....2145
SEARch:TRIGger:USB:EMIN....2146
SEARch:TRIGger:USB:EMAX....2146
SEARch:TRIGger:USB:FCONdition....2146
SEARch:TRIGger:USB:FMIN....2147
SEARch:TRIGger:USB:FMAX....2147
SEARch:TRIGger:USB:GLITcherror....2147
SEARch:TRIGger:USB:HAND....2147
SEARch:TRIGger:USB:PATT....2148
SEARch:TRIGger:USB:PCONdition....2148
SEARch:TRIGger:USB:PMIN....2148
SEARch:TRIGger:USB:PMAX....2149
SEARch:TRIGger:USB:PIDerror....2149
SEARch:TRIGger:USB:SCONdition....2149
SEARch:TRIGger:USB:SMIN....2150
SEARch:TRIGger:USB:SMAX....2150
SEARch:TRIGger:USB:SDATa....2150
SEARch:TRIGger:USB:SERRor....2150
SEARch:TRIGger:USB:SHANdshake....2151
SEARch:TRIGger:USB:SSOP....2151
SEARch:TRIGger:USB:SSPE....2151
SEARch:TRIGger:USB:SPEC....2151
SEARch:TRIGger:USB:STCO....2152
SEARch:TRIGger:USB:STOKen....2152
SEARch:TRIGger:USB:TCONdition....2153
SEARch:TRIGger:USB:TMIN....2153
SEARch:TRIGger:USB:TMAX....2153
SEARch:TRIGger:USB:TOKen....2154
SEARch:TRIGger:USB:WADD....2154
SEARch:TRIGger:USB:WEND....2154
SEARch:TRIGger:USB:WETCheck....2155
SEARch:TRIGger:USB:WFRN....2155
SEARch:TRIGger:USB:WPAY....2155
SEARch:TRIGger:USB:WPID....2156

SEARch:TRIGger:USB:WPOR....2156
SEARch:TRIGger:USB:WSEU....2156
SEARch:TRIGger:USB:WSTC....2156

**SEARch:TRIGger:USB:ACONdition** <SearchName>,<OperatorAddress>
**SEARch:TRIGger:USB:ACONdition?** <SearchName>

Sets the operator to set a specific address or an address range.

**Parameters:**
<OperatorAddress> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one address value to be set using `SEARch:TRIGger:USB:AMIN`.

**INRange | OORange**
In range, out of range. These conditions require an address range to be set using `SEARch:TRIGger:USB:AMIN` and `SEARch:TRIGger:USB:AMAX`.

*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Address check" on page 830

**SEARch:TRIGger:USB:AMIN** <SearchName>,<Address>
**SEARch:TRIGger:USB:AMIN?** <SearchName>

Specifies an address, or sets the start value of an address range. The string parameter does not accept the bit value X (don't care).

**Parameters:**
<Address>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Address check" on page 830

**SEARch:TRIGger:USB:AMAX** <SearchName>,<AddressTo>
**SEARch:TRIGger:USB:AMAX?** <SearchName>

Sets the the end value of an address range if `TRIGger<m>:USB:ACONdition` is set to `INRange` or `OORange`. The string parameter does not accept the bit value X (don't care).

**Parameters:**
<AddressTo>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Address check" on page 830

---

**SEARch:TRIGger:USB:BITSterror** <SearchName>,<BitstuffError>
**SEARch:TRIGger:USB:BITSterror?** <SearchName>

Defines, whether a search for any bitstuffing error shall be activated or not.

**Parameters:**
<BitstuffError> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error condition" on page 830

---

**SEARch:TRIGger:USB:CRC16error** <SearchName>,<CRC16Error>
**SEARch:TRIGger:USB:CRC16error?** <SearchName>

Defines, whether a search for any CRC16 error shall be activated or not.

**Parameters:**
<CRC16Error> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error condition" on page 830

---

**SEARch:TRIGger:USB:CRC5error** <SearchName>,<CRC5Error>
**SEARch:TRIGger:USB:CRC5error?** <SearchName>

Defines, whether a search for any CRC5 error shall be activated or not.

**Parameters:**
<CRC5Error> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error condition" on page 830

**SEARch:TRIGger:USB:DATA** <SearchName>,<USBDataType>
**SEARch:TRIGger:USB:DATA?** <SearchName>

Defines, which data packet type is searched for: "DATA0", "DATA1", "DATA2", or "MDATA", as well as "Any" data packet.

**Parameters:**
<USBDataType> ANY | D0 | D1 | D2 | MD

**ANY**
Searches for any of the data packet types listed below

**D0**
Searches for a DATA0 packet (even PID)

**D1**
Searches for a DATA1 packet (odd PID)

**D2**
Searches for a DATA2 packet (high-speed data packet for high bandwidth isochronous transaction in a microframe)

**MD**
Searches for an MDATA packet (high-speed data packet for split and high bandwidth isochronous transactions)

*RST: D0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data" on page 829

**SEARch:TRIGger:USB:DCONdition** <SearchName>,<OperatorData>
**SEARch:TRIGger:USB:DCONdition?** <SearchName>

Sets the operator (equal or unequal) to set a specific payload data pattern.

**Parameters:**
<OperatorData> OFF | ANY | EQUal | NEQual

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Payload check" on page 831

**SEARch:TRIGger:USB:DPOPerator** <SearchName>,<DataPosOperator>
**SEARch:TRIGger:USB:DPOPerator?** <SearchName>

Sets the operator (any or equal) for the payload data index position.

**Parameters:**
<DataPosOperator> ANY | OFF | EQUal

**ANY = OFF**
The position of payload data is not relevant for the search condition.

*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Payload check" on page 831

---

**SEARch:TRIGger:USB:DPOSition** <SearchName>,<DataPosition>
**SEARch:TRIGger:USB:DPOSition?** <SearchName>

Specifies the position within a payload data packet, in which a special data pattern is to be searched.

**Parameters:**
<DataPosition> Range: 0 to 1024
Increment: 1
*RST: 0

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Payload check" on page 831

---

**SEARch:TRIGger:USB:ECONdition** <SearchName>,<EndpointOptor>
**SEARch:TRIGger:USB:ECONdition?** <SearchName>

Sets the operator to set a specific endpoint or an endpoint range. The endpoint values are set with `SEARch:TRIGger:USB:EMIN` and `SEARch:TRIGger:USB:EMAX`.

**Parameters:**
<EndpointOptor> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one endpoint value to be set using `SEARch:TRIGger:USB:EMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of endpoint values to be set using `SEARch:TRIGger:USB:EMIN` and `SEARch:TRIGger:USB:EMAX`.

*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "End Point check" on page 830

**SEARch:TRIGger:USB:EMIN** <SearchName>,<Endp>
**SEARch:TRIGger:USB:EMIN?** <SearchName>

Specifies an endpoint, or sets the start value of an endpoint range.

**Parameters:**
<Endp>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "End Point check" on page 830

**SEARch:TRIGger:USB:EMAX** <SearchName>,<EndpTo>
**SEARch:TRIGger:USB:EMAX?** <SearchName>

Sets the the end value of an endpoint range if `TRIGger<m>:USB:ECONdition` is set to `INRange` or `OORange`.

**Parameters:**
<EndpTo>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "End Point check" on page 830

**SEARch:TRIGger:USB:FCONdition** <SearchName>,<FrameNoOperator>
**SEARch:TRIGger:USB:FCONdition?** <SearchName>

Sets the operator to set a specific frame number or a frame number range. The frame number values are set with `SEARch:TRIGger:USB:FMIN` and `SEARch:TRIGger:USB:FMAX`.

**Parameters:**
<FrameNoOperator> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one frame number value to be set using `SEARch:TRIGger:USB:FMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of frame number values to be set using `SEARch:TRIGger:USB:FMIN` and `SEARch:TRIGger:USB:FMAX`.

*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame number check" on page 831

**SEARch:TRIGger:USB:FMIN** <SearchName>,<FrameNumber>
**SEARch:TRIGger:USB:FMIN?** <SearchName>

Specifies a frame number, or sets the start value of a frame number range.

**Parameters:**
<FrameNumber>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame number check" on page 831

**SEARch:TRIGger:USB:FMAX** <SearchName>,<FrameNumberTo>
**SEARch:TRIGger:USB:FMAX?** <SearchName>

Sets the the end value of a frame number range if `TRIGger<m>:USB:FCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<FrameNumberTo>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame number check" on page 831

**SEARch:TRIGger:USB:GLITcherror** <SearchName>,<GlitchError>
**SEARch:TRIGger:USB:GLITcherror?** <SearchName>

Defines, whether a search for any glitch error shall be activated or not.

**Parameters:**
<GlitchError> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error condition" on page 830

**SEARch:TRIGger:USB:HAND** <SearchName>,<HandshakeType>
**SEARch:TRIGger:USB:HAND?** <SearchName>

Defines, which handshake type is searched for.

**Parameters:**
<HandshakeType> ANY | ACK | NAK | STALl | NYET
*RST: ACK

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Handshake" on page 829

**SEARch:TRIGger:USB:PATT** <SearchName>,<PayloadMuster>
**SEARch:TRIGger:USB:PATT?** <SearchName>

Defines the payload data pattern to search for.

**Parameters:**
<PayloadMuster>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Payload check" on page 831

**SEARch:TRIGger:USB:PCONdition** <SearchName>,<OperatorPort>
**SEARch:TRIGger:USB:PCONdition?** <SearchName>

Sets the operator to set a specific port number or a port number range. The port number values are set with `SEARch:TRIGger:USB:PMIN` and `SEARch:TRIGger:USB:PMAX`.

**Parameters:**
<OperatorPort> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one port number to be set using `SEARch:TRIGger:USB:PMIN`.

**INRange | OORange**
In range, out of range. These conditions require a port number range to be set using `SEARch:TRIGger:USB:PMIN` and `SEARch:TRIGger:USB:PMAX`.

*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Port check" on page 832

**SEARch:TRIGger:USB:PMIN** <SearchName>,<Port>
**SEARch:TRIGger:USB:PMIN?** <SearchName>

Specifies a port number, or sets the start value of a port number range.

**Parameters:**
<Port>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Port check" on page 832

**SEARch:TRIGger:USB:PMAX** <SearchName>,<PortTo>
**SEARch:TRIGger:USB:PMAX?** <SearchName>

Sets the the end value of a port number range if `TRIGger<m>:USB:PCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<PortTo>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Port check" on page 832

**SEARch:TRIGger:USB:PIDerror** <SearchName>,<PIDError>
**SEARch:TRIGger:USB:PIDerror?** <SearchName>

Defines, whether a search for any PID error shall be activated or not.

**Parameters:**
<PIDError> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error condition" on page 830

**SEARch:TRIGger:USB:SCONdition** <SearchName>,<OperatorSEU>
**SEARch:TRIGger:USB:SCONdition?** <SearchName>

Sets the operator to set a specific SEU or an SEU range. The SEU values are set with `SEARch:TRIGger:USB:SMIN` and `SEARch:TRIGger:USB:SMAX`.

For SEU, see "SEU" on page 825.

**Parameters:**
<OperatorSEU> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one SEU value to be set using `SEARch:TRIGger:USB:SMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of SEU values to be set using `SEARch:TRIGger:USB:SMIN` and `SEARch:TRIGger:USB:SMAX`.

*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "SEU check" on page 832

**SEARch:TRIGger:USB:SMIN** <SearchName>,<SEU>
**SEARch:TRIGger:USB:SMIN?** <SearchName>

Specifies an SEU, or sets the start value of an SEU range.

**Parameters:**
<SEU>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "SEU check" on page 832

**SEARch:TRIGger:USB:SMAX** <SearchName>,<SEUTo>
**SEARch:TRIGger:USB:SMAX?** <SearchName>

Sets the the end value of an SEU range if `TRIGger<m>:USB:SCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<SEUTo>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "SEU check" on page 832

**SEARch:TRIGger:USB:SDATa** <SearchName>,<DataPacket>
**SEARch:TRIGger:USB:SDATa?** <SearchName>

Enables the search for any data packet.

**Parameters:**
<DataPacket> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search types" on page 828

**SEARch:TRIGger:USB:SERRor** <SearchName>,<ErrorCondition>
**SEARch:TRIGger:USB:SERRor?** <SearchName>

Enables the search for various errors.

**Parameters:**
<ErrorCondition> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search types" on page 828

**SEARch:TRIGger:USB:SHANdshake** <SearchName>,<HandshakePacket>
**SEARch:TRIGger:USB:SHANdshake?** <SearchName>

Defines, whether the search for any handshake packet shall be activated or not.

**Parameters:**
<HandshakePacket> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search types" on page 828

**SEARch:TRIGger:USB:SSOP** <SearchName>,<StartofPacket>
**SEARch:TRIGger:USB:SSOP?** <SearchName>

Defines, whether a start of packet (SOP) search shall be activated or not.

**Parameters:**
<StartofPacket> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search types" on page 828

**SEARch:TRIGger:USB:SSPE** <SearchName>,<SpecialPacket>
**SEARch:TRIGger:USB:SSPE?** <SearchName>

Defines, whether the search for any special PID packet shall be activated or not.

**Parameters:**
<SpecialPacket> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search types" on page 828

**SEARch:TRIGger:USB:SPEC** <SearchName>,<USBSpecialType>
**SEARch:TRIGger:USB:SPEC?** <SearchName>

Defines, which special PID packet type is searched for: "PREamble", "ERR", "SPLit", "PING", or "Any" special PID packet.

**Parameters:**
<USBSpecialType> ANY | PREamble | ERR | SPLit | PING

**ANY**
Searches for any of the special PID packet types listed below

**PREamble**
Searches for any host-issued preamble token (enables downstream bus traffic to low speed USB devices)

**ERR**
Searches for any SPLIT transaction error handshake token (reuses PRE value)

**SPLit**
Searches for any high speed SPLIT transaction token

**PING**
Searches for any high speed flow control probe for a bulk/control endpoint

*RST: PREamble

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Special PID" on page 829

---

**SEARch:TRIGger:USB:STCO** <SearchName>,<SC>
**SEARch:TRIGger:USB:STCO?** <SearchName>

Defines, which Start- / Complete- (SC) split transaction type is searched for: SSPLIT or CSPLIT.

**Parameters:**
<SC> ONE | ZERO | DC

**ONE**
SC = 1 represents a complete-split (CSPLIT) transaction

**ZERO**
SC = 0 represents a start-split (SSPLIT) transaction

**DC**
SC = X represents "don't care" (DC)

*RST: DC

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "SC check" on page 832

---

**SEARch:TRIGger:USB:STOKen** <SearchName>,<TokenPacket>
**SEARch:TRIGger:USB:STOKen?** <SearchName>

Defines, whether a search for any token packet shall be activated or not.

**Parameters:**
<TokenPacket> ON | OFF

*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Search types" on page 828

---

**SEARch:TRIGger:USB:TCONdition** <SearchName>,<OperatorET>
**SEARch:TRIGger:USB:TCONdition?** <SearchName>

Sets the operator to set a specific endpoint type (ET) or an ET range. The ET values are set with `SEARch:TRIGger:USB:TMIN` and `SEARch:TRIGger:USB:TMAX`.

**Parameters:**
<OperatorET> OFF | ANY | EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**OFF = ANY**
The endpoint type is not relevant for the search condition.

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one ET value to be set using `SEARch:TRIGger:USB:TMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of ET values to be set using `SEARch:TRIGger:USB:TMIN` and `SEARch:TRIGger:USB:TMAX`.

*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "ET check" on page 832

---

**SEARch:TRIGger:USB:TMIN** <SearchName>,<ET>
**SEARch:TRIGger:USB:TMIN?** <SearchName>

Specifies an endpoint type (ET), or sets the start value of an ET range.

**Parameters:**
<ET>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "ET check" on page 832

---

**SEARch:TRIGger:USB:TMAX** <SearchName>,<ETTo>
**SEARch:TRIGger:USB:TMAX?** <SearchName>

Sets the the end value of an endpoint type (ET) range if `TRIGger<m>:USB:TCONdition` is set to `INRange` or `OORange`.

**Parameters:**
<ETTo>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "ET check" on page 832

---

**SEARch:TRIGger:USB:TOKen** <SearchName>,<USBTokenType>
**SEARch:TRIGger:USB:TOKen?** <SearchName>

Defines, which token packet type is searched for: "OUT", "IN", "SOF", "SETUP", or "Any" token packet.

**Parameters:**
<USBTokenType> ANY | OUT | IN | SOF | SETup

**ANY**
Searches for any of the token packet types listed below.

**OUT**
Searches for an OUT token.

**IN**
Searches for an IN token.

**SOF**
Searches for a start of frame (SOF) token.

**SETup**
Searches for a SETup token.

*RST: OUT

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Token" on page 829

---

**SEARch:TRIGger:USB:WADD** <SearchName>,<AddressCheck>
**SEARch:TRIGger:USB:WADD?** <SearchName>

Defines, whether a search for any address field shall be activated or not.

**Parameters:**
<AddressCheck> ON | OFF

*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Address check" on page 830

---

**SEARch:TRIGger:USB:WEND** <SearchName>,<EndpointCheck>
**SEARch:TRIGger:USB:WEND?** <SearchName>

Defines, whether a search for any endpoint shall be activated or not.

**Parameters:**
<EndpointCheck> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "End Point check" on page 830

---

**SEARch:TRIGger:USB:WETCheck** <SearchName>,<WithETCheck>
**SEARch:TRIGger:USB:WETCheck?** <SearchName>

Defines, whether a search for any Endpoint Type (ET) shall be activated or not.

**Parameters:**
<WithETCheck> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "ET check" on page 832

---

**SEARch:TRIGger:USB:WFRN** <SearchName>,<FrameNoCheck>
**SEARch:TRIGger:USB:WFRN?** <SearchName>

Defines, whether a search for any frame number shall be activated or not.

**Parameters:**
<FrameNoCheck> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Frame number check" on page 831

---

**SEARch:TRIGger:USB:WPAY** <SearchName>,<PayloadCheck>
**SEARch:TRIGger:USB:WPAY?** <SearchName>

Defines, whether a search for any payload data shall be activated or not.

**Parameters:**
<PayloadCheck> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Payload check" on page 831

**SEARch:TRIGger:USB:WPID** <SearchName>,<WithPIDCheck>
**SEARch:TRIGger:USB:WPID?** <SearchName>

Defines, whether a search for any packet ID error shall be activated or not.

**Parameters:**
<WithPIDCheck> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**SEARch:TRIGger:USB:WPOR** <SearchName>,<WithPortCheck>
**SEARch:TRIGger:USB:WPOR?** <SearchName>

Defines, whether a search for any port shall be activated or not.

**Parameters:**
<WithPortCheck> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Port check" on page 832

**SEARch:TRIGger:USB:WSEU** <SearchName>,<WithSEUCheck>
**SEARch:TRIGger:USB:WSEU?** <SearchName>

Defines, whether a search for any SEU shall be activated or not.

For SEU, see "SEU" on page 825.

**Parameters:**
<WithSEUCheck> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "SEU check" on page 832

**SEARch:TRIGger:USB:WSTC** <SearchName>,<WithSCCheck>
**SEARch:TRIGger:USB:WSTC?** <SearchName>

Defines, whether a search for any Start / Complete (SC) shall be activated or not.

**Parameters:**
<WithSCCheck> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "SC check" on page 832

### 24.17.19.5 Search results

In all `SEARch:RESult:USB:PACKet<m>` commands, the suffix <m> selects the packet number in the list of search results.

SEARch:RESult:USB:PACKet<m>:ADDRess? ..... 2157
SEARch:RESult:USB:PACKet<m>:CRC? ..... 2157
SEARch:RESult:USB:PACKet<m>:DATA? ..... 2158
SEARch:RESult:USB:PACKet<m>:ENDPoint? ..... 2158
SEARch:RESult:USB:PACKet<m>:ET? ..... 2158
SEARch:RESult:USB:PACKet<m>:FRAMe? ..... 2159
SEARch:RESult:USB:PACKet<m>:PID? ..... 2159
SEARch:RESult:USB:PACKet<m>:PORT? ..... 2159
SEARch:RESult:USB:PACKet<m>:SC? ..... 2160
SEARch:RESult:USB:PACKet<m>:SEU? ..... 2160
SEARch:RESult:USB:PACKet<m>:STARt? ..... 2160
SEARch:RESult:USB:PACKet<m>:STATus? ..... 2161
SEARch:RESult:USB:PACKet<m>:STOP? ..... 2161
SEARch:RESult:USB:PCOunt? ..... 2161

---

**SEARch:RESult:USB:PACKet<m>:ADDRess?** <SearchName>

Returns the packet address for the selected packet number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Addr> Range: 0 to 127
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:USB:PACKet<m>:CRC?** <SearchName>

Returns the packet CRC (Cyclic Redundancy Code) for the selected packet number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<CRC> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:USB:PACKet<m>:DATA?** <SearchName>

Returns the payload data from the packet with the selected packet number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

---

**SEARch:RESult:USB:PACKet<m>:ENDPoint?** <SearchName>

Returns the endpoint for the selected packet number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Endp> Range: 0 to 15
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:USB:PACKet<m>:ET?** <SearchName>

Returns the endpoint type (ET) for the selected packet number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<ET> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:USB:PACKet<m>:FRAMe?** <SearchName>

Returns the frame number for the selected packet number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameNo> Range: 0 to 2047
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:USB:PACKet<m>:PID?** <SearchName>

Returns the packet PID for the selected packet number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<PID> RES | OUT | ACK | DATA0 | DATa0 | PING | SOF | NYET | DATA2 | DATa2 | SPLIT | SPLit | IN | NAK | DATA1 | DATa1 | PRE | SETUP | SETup | STALL | STALI | MDATA | MDATa | UNK

For a description of the return values, see BUS<m>:USB:PACKet<n>:PID? on page 2135.

*RST: RES

**Usage:** Query only

---

**SEARch:RESult:USB:PACKet<m>:PORT?** <SearchName>

Returns the port number for the selected packet number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Port> Range: 0 to 127
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:USB:PACKet<m>:SC?** <SearchName>

Returns the Start- / Complete-split transaction (SSPLIT / CSPLIT) flag bits for the selected packet number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<SC>

**Usage:** Query only

---

**SEARch:RESult:USB:PACKet<m>:SEU?** <SearchName>

Returns the SEU values for the selected packet number within the search result.

For SEU, see "SEU" on page 825.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<SEU> Range: 0 to 3
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:USB:PACKet<m>:STARt?** <SearchName>

Returns the start time of the packet with the selected packet number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:USB:PACKet<m>:STATus?** <SearchName>

Returns the status of the packet with the selected packet number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | PID | CRC | BTST | GLITCH | GLITch | BYTE
*RST: OK

**Usage:** Query only

**SEARch:RESult:USB:PACKet<m>:STOP?** <SearchName>

Returns the stop time of the packet with the selected packet number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:USB:PCOunt?** <SearchName>

Returns the search result's packet count, i.e. the number of packets found in the search result.

**Query parameters:**
<SearchName>

**Return values:**
<Count> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

## 24.17.20 USB 3.1 (option R&S RTP-K61/K62)


- Configuration....2162
- Digital signal processing (DSP)....2164
- Trigger....2167
- Decode results....2185
- Search settings....2190
- Search results....2196


### 24.17.20.1 Configuration

In all `BUS<m>:USBT` commands, the suffix <m> selects the serial bus.


BUS<m>:USBThree:SOURce....2162
BUS<m>:USBThree:POLarity....2162
BUS<m>:USBThree:THRHigh....2163
BUS<m>:USBThree:THRLow....2163
BUS<m>:USBThree:SCRMode....2163
BUS<m>:USBThree:TYPE....2164


---

**BUS<m>:USBThree:SOURce** <SourceData>

Selects the source for the data signal in the USB 3.1 protocol.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceData> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 845

---

**BUS<m>:USBThree:POLarity** <Polarity>

Defines the polarity of the data signal for the active (U0) state of the port.

**Suffix:**
<m> 1..4

**Parameters:**
<Polarity> NORMal | INVert

**NORMal**
Normal (non-inverted) data signal polarity

**INVert**
Inverted data signal polarity

*RST: NORMal

**Manual operation:** See "Polarity" on page 845

**BUS<m>:USBThree:THRHigh** <ThresholdUpper>

Defines the upper threshold level for digitization of the data signal in the USB 3.1 protocol.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdUpper> Range: -2 to 2
Increment: 0.01
*RST: 0.05
Default unit: V

**Manual operation:** See "Thresholds" on page 845

**BUS<m>:USBThree:THRLow** <ThresholdLower>

Defines the lower threshold level for digitization of the data signal in the USB 3.1 protocol.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdLower> Range: -2 to 2
Increment: 0.01
*RST: -0.05
Default unit: V

**Manual operation:** See "Thresholds" on page 845

**BUS<m>:USBThree:SCRMode** <ScrambleMode>

Selects if the data pattern is scrambled or unscambled.

**Suffix:**
<m> 1..4

**Parameters:**
<ScrambleMode> SCRM | NSCR

**SCRM**
Scrambling is enabled as specified in the standard, to minimize EMI emissions.

**NSCR**
No scrambling: To simplify testing and debugging, scrambling is disabled.

*RST: SCRM

**Manual operation:** See "Scramble Mode" on page 845

**BUS<m>:USBThree:TYPE** <Protocol>

Selects the USB 3.1 protocol type.

**Suffix:**
<m> 1..4

**Parameters:**
<Protocol> SS | SSP
*RST: SS

**Usage:** Asynchronous command

**Manual operation:** See "Protocol type" on page 844

### 24.17.20.2 Digital signal processing (DSP)

BUS<m>:CTLE:DCGain....2164
BUS<m>:CTLE:ENBLe....2164
BUS<m>:CTLE:P1FReq....2165
BUS<m>:CTLE:P2FReq....2165
BUS<m>:CTLE:Z1FReq....2165
BUS<m>:USBThree:BITR....2166
BUS<m>:USBThree:DFE:COEFficient....2166
BUS<m>:USBThree:DFE:ENBLe....2166
BUS<m>:USBThree:DFE:ESTCoeff....2166
BUS<m>:USBThree:DFE:TCONt....2167
BUS<m>:USBThree:PROTocol....2167

**BUS<m>:CTLE:DCGain** <DCGain>

Sets the adjustable DC gain. This value can be chosen according to the channel loss characteristics. For low-loss channels, you can set a value close to 0 dB.

**Suffix:**
<m> 1..4

**Parameters:**
<DCGain> Range: -100 to 100
Increment: 0.01
*RST: 0
Default unit: dB

**Manual operation:** See "DC Gain ($A_{DC}$)" on page 846

**BUS<m>:CTLE:ENBLe** <EnableCTLE>

Enables the continuous time linear equalization (CTLE) equalization.

**Suffix:**
<m> 1..4

**Parameters:**
<EnableCTLE> ON | OFF
*RST: OFF

**Manual operation:** See "Enable" on page 846

---

**BUS<m>:CTLE:P1FReq** <PoleOneFreq>

Sets the pole 1 frequency for the transfer function of the CTLE algorithm.

**Suffix:**
<m> 1..4

**Parameters:**
<PoleOneFreq> Range: 1 to 40E+9
Increment: 1
*RST: 0
Default unit: Hz

**Manual operation:** See "Pole frequency (wp1)/(wp2)" on page 846

---

**BUS<m>:CTLE:P2FReq** <PoleTwoFreq>

Sets the pole 2 frequency for the transfer function of the CTLE algorithm.

**Suffix:**
<m> 1..4

**Parameters:**
<PoleTwoFreq> Range: 1 to 40E+9
Increment: 1
*RST: 0
Default unit: Hz

**Manual operation:** See "Pole frequency (wp1)/(wp2)" on page 846

---

**BUS<m>:CTLE:Z1FReq** <ZroOneFreq>

Sets the zero frequency for the transfer function of the CTLE algorithm.

**Suffix:**
<m> 1..4

**Parameters:**
<ZroOneFreq> Range: 1 to 40E+9
Increment: 1
*RST: 0
Default unit: Hz

**Manual operation:** See "Zero frequency (fz)" on page 846

**BUS<m>:USBThree:BITR** <HighSpdBitrate>

Sets the bit rate.

**Suffix:**
<m> 1..4

**Parameters:**
<HighSpdBitrate> Range: 1000 to 3E+9
Increment: 1000
*RST: 3E+9
Default unit: bps

**BUS<m>:USBThree:DFE:COEFficient** <DFEcoeff>

Sets the value of the coefficient for the DFE .

**Suffix:**
<m> 1..4

**Parameters:**
<DFEcoeff> Range: -1 to 1
Increment: 1E-3
*RST: 0
Default unit: V

**Manual operation:** See "Coefficient" on page 847

**BUS<m>:USBThree:DFE:ENBLe** <EnableDFE>

Enables the use of the decision feedback equalization for USB 3.1.

**Suffix:**
<m> 1..4

**Parameters:**
<EnableDFE> ON | OFF
*RST: ON

**Manual operation:** See "Enable" on page 847

**BUS<m>:USBThree:DFE:ESTCoeff** <AutoChEstimate>

Enables the use of estimate coefficients for the decision feedback equalization.

**Suffix:**
<m> 1..4

**Parameters:**
<AutoChEstimate> ON | OFF
*RST: ON

**Manual operation:** See "Estimate coefficients" on page 847

**BUS<m>:USBThree:DFE:TCONt** <CIRLength>

Sets a number of taps for the decision feedback filter.

**Suffix:**
<m> 1..4

**Parameters:**
<CIRLength> Range: 1 to 20
Increment: 1
*RST: 5

**Manual operation:** See "Number of Taps" on page 847

**BUS<m>:USBThree:PROTocol** <ProtHSSPTSt>

Selects the USB Super Speed channel length.

**Suffix:**
<m> 1..4

**Parameters:**
<ProtHSSPTSt> OFF | LOWFLTR | HIGHFLTR
*RST: LOWFLTR

**Manual operation:** See "Channel Length" on page 847

### 24.17.20.3 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

There are two commands for each parameter, that you can use for defining the USB 3.1 settings.

For example, to set the *Frame type =ACK > Field =Route String >Data* value you can use one of the following commands:

- `TRIGger:USBThree:FRAMe13:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `TRIGger:USBThree:DMIN "ACK", "Route String", 01100`
  Defines the parameter by using the frame and field name.

Table 24-14 and table 24-15 give an overview of the available frame and field numbers. Those numbers follow the GUI numbers.

*Table 24-14: Overview of frame and field numbers for USB 3.1, Protocol Type > Super Speed*

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| 1 | TSEQ | 1 | Symbol |
| 2 | TS1 | 1 | Train |
| | | 2 | Loop |
| | | 3 | Scramble |
| | | 4 | LL |
| | | 5 | Retimer |
| 3 | TS1A | - | - |
| 4 | TS1B | - | - |
| 5 | TS2 | 1 | Train |
| | | 2 | Loop |
| | | 3 | Scramble |
| | | 4 | LL |
| | | 5 | Retimer |
| 6 | Set Link Function | 1 | SLF |
| | | 2 | HDr # |
| | | 3 | Hub # |
| | | 4 | DLY |
| | | 5 | DFR |
| 7 | U2 Inactivity Timeout | 1 | U2 IT |
| | | 2 | HDr # |
| | | 3 | Hub # |
| | | 4 | DLY |
| | | 5 | DFR |
| 8 | Vendor Device Test | 1 | VDT |
| | | 2 | Vendor Def |
| | | 3 | HDr # |
| | | 4 | Hub # |
| | | 5 | DLY |
| | | 6 | DFR |
| 9 | Port Capability | 1 | Link Speed |
| | | 2 | Num HP Buf |
| | | 3 | Dir |
| | | 4 | OTG |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| | | 5 | Tiebreake |
| | | 6 | HDr # |
| | | 7 | Hub # |
| | | 8 | DLY |
| | | 9 | DFR |
| 10 | Port Config | 1 | Link Speed |
| | | 2 | HDr # |
| | | 3 | Hub # |
| | | 4 | DLY |
| | | 5 | DFR |
| 11 | Port Config Rsp | 1 | Response Code |
| | | 2 | HDr # |
| | | 3 | Hub # |
| | | 4 | DLY |
| | | 5 | DFR |
| 12 | Link Delay Meas | 1 | LDM Type |
| | | 2 | LDMS |
| | | 3 | Response Delay |
| | | 4 | HDr # |
| | | 5 | Hub # |
| | | 6 | DLY |
| | | 7 | DFR |
| 13 | ACK | 1 | Route String |
| | | 2 | Device Address |
| | | 3 | rty |
| | | 4 | D |
| | | 5 | EPTNum |
| | | 6 | TT |
| | | 7 | HE |
| | | 8 | NumP |
| | | 9 | SeqNum |
| | | 10 | TPF |
| | | 11 | StreamId |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| | | 12 | SSI |
| | | 13 | WPA |
| | | 14 | DBI |
| | | 15 | PP |
| | | 16 | NBI |
| | | 17 | HDr # |
| | | 18 | Hub # |
| | | 19 | DLY |
| | | 20 | DFR |
| 14 | NRDY | 1 | Device Address |
| | | 2 | D |
| | | 3 | EPTNum |
| | | 4 | StreamId |
| | | 5 | HDr # |
| | | 6 | Hub # |
| | | 7 | DLY |
| | | 8 | DFR |
| 15 | ERDY | 1 | Device Address |
| | | 2 | D |
| | | 3 | EPTNum |
| | | 4 | NumP |
| | | 5 | StreamId |
| | | 6 | HDr # |
| | | 7 | Hub # |
| | | 8 | DLY |
| | | 9 | DFR |
| 16 | Status | 1 | Route String |
| | | 2 | Device Address |
| | | 3 | D |
| | | 4 | EPTNum |
| | | 5 | PP |
| | | 6 | HDr # |
| | | 7 | Hub # |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| | | 8 | DLY |
| | | 9 | DFR |
| 17 | Stall | 1 | Device Address |
| | | 2 | D |
| | | 3 | EPTNum |
| | | 4 | HDr # |
| | | 5 | Hub # |
| | | 6 | DLY |
| | | 7 | DFR |
| 18 | Function Wake | 1 | Device Address |
| | | 2 | Interface |
| | | 3 | HDr # |
| | | 4 | Hub # |
| | | 5 | DLY |
| | | 6 | DFR |
| 19 | Latency Tolerance | 1 | Device Address |
| | | 2 | BELT |
| | | 3 | HDr # |
| | | 4 | Hub # |
| | | 5 | DLY |
| | | 6 | DFR |
| 20 | Bus Interval Adjust | 1 | Device Address |
| | | 2 | BusIntAdj |
| | | 3 | HDr # |
| | | 4 | Hub # |
| | | 5 | DLY |
| | | 6 | DFR |
| 21 | Host Role Req | 1 | Device Address |
| | | 2 | RSP |
| | | 3 | HDr # |
| | | 4 | Hub # |
| | | 5 | DLY |
| | | 6 | DFR |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| 22 | Sublink Speed | 1 | Device Address |
| | | 2 | TPF |
| | | 3 | LSE |
| | | 4 | ST |
| | | 5 | Lanes |
| | | 6 | LP |
| | | 7 | LSM |
| | | 8 | HDr # |
| | | 9 | Hub # |
| | | 10 | DLY |
| | | 11 | DFR |
| 23 | Ping | 1 | Route String |
| | | 2 | Device Address |
| | | 3 | D |
| | | 4 | EPTNum |
| | | 5 | HDr # |
| | | 6 | Hub # |
| | | 7 | DLY |
| | | 8 | DFR |
| 24 | Ping Rsp | 1 | Device Address |
| | | 2 | D |
| | | 3 | EPTNum |
| | | 4 | HDr # |
| | | 5 | Hub # |
| | | 6 | DLY |
| | | 7 | DFR |
| 25 | Data Packet Header | 1 | Route String |
| | | 2 | Device Address |
| | | 3 | SeqNum |
| | | 4 | EOB |
| | | 5 | D |
| | | 6 | EPTNum |
| | | 7 | TT |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| | | 8 | S |
| | | 9 | Length |
| | | 10 | StreamId |
| | | 11 | SSI |
| | | 12 | WPA |
| | | 13 | DBI |
| | | 14 | PP |
| | | 15 | NBI |
| | | 16 | HDr # |
| | | 17 | Hub # |
| | | 18 | DLY |
| | | 19 | DFR |
| 26 | Data Packet Payload | 1 | Data |
| 27 | DPP Aborted | 1 | Data |
| 28 | ITP | 1 | Counter |
| | | 2 | Delta |
| | | 3 | Adj Ctrl |
| | | 4 | Corr |
| | | 5 | HDr # |
| | | 6 | Hub # |
| | | 7 | DLY |
| | | 8 | DFR |
| 29 | Link Command | 1 | Sub-type |
| 30 | Skip | - | - |
| 31 | BRST | - | - |
| 32 | BDAT | - | - |
| 33 | BERC | - | - |
| 34 | BCNT | 1 | EC |
| 35 | Idle | - | - |

*Table 24-15: Overview of frame and field numbers for USB 3.1, Protocol Type > Super Speed Plus*

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| 1 | TSEQ | - | - |
| 2 | TS1 | 1 | Train |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| | | 2 | Loop |
| | | 3 | Scramble |
| | | 4 | LL |
| | | 5 | Retimer |
| 3 | TS1A | 1 | TS1A id |
| | | 2 | TS1 id |
| 4 | TS1B | 1 | TS1B id |
| | | 2 | TS1 id |
| 5 | TS2 | 1 | Train |
| | | 2 | Loop |
| | | 3 | Scramble |
| | | 4 | LL |
| | | 5 | Retimer |
| 6 | Set Link Function | 1 | SLF |
| | | 2 | HDr # |
| | | 3 | Hub # |
| | | 4 | DLY |
| | | 5 | DFR |
| 7 | U2 Inactivity Timeout | 1 | U2 IT |
| | | 2 | HDr # |
| | | 3 | Hub # |
| | | 4 | DLY |
| | | 5 | DFR |
| 8 | Vendor Device Test | 1 | VDT |
| | | 2 | Vendor Def |
| | | 3 | HDr # |
| | | 4 | Hub # |
| | | 5 | DLY |
| | | 6 | DFR |
| 9 | Port Capability | 1 | Link Speed |
| | | 2 | Num HP Buf |
| | | 3 | Dir |
| | | 4 | OTG |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| | | 5 | Tiebreake |
| | | 6 | HDr # |
| | | 7 | Hub # |
| | | 8 | DLY |
| | | 9 | DFR |
| 10 | Port Config | 1 | Link Speed |
| | | 2 | HDr # |
| | | 3 | Hub # |
| | | 4 | DLY |
| | | 5 | DFR |
| 11 | Port Config Rsp | 1 | Response Code |
| | | 2 | HDr # |
| | | 3 | Hub # |
| | | 4 | DLY |
| | | 5 | DFR |
| 12 | Link Delay Meas | 1 | LDM Type |
| | | 2 | LDMS |
| | | 3 | Response Delay |
| | | 4 | HDr # |
| | | 5 | Hub # |
| | | 6 | DLY |
| | | 7 | DFR |
| 13 | ACK | 1 | Route String |
| | | 2 | Device Address |
| | | 3 | rty |
| | | 4 | D |
| | | 5 | EPTNum |
| | | 6 | TT |
| | | 7 | HE |
| | | 8 | NumP |
| | | 9 | SeqNum |
| | | 10 | TPF |
| | | 11 | StreamId |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| | | 12 | SSI |
| | | 13 | WPA |
| | | 14 | DBI |
| | | 15 | PP |
| | | 16 | NBI |
| | | 17 | HDr # |
| | | 18 | Hub # |
| | | 19 | DLY |
| | | 20 | DFR |
| 14 | NRDY | 1 | Device Address |
| | | 2 | D |
| | | 3 | EPTNum |
| | | 4 | StreamId |
| | | 5 | HDr # |
| | | 6 | Hub # |
| | | 7 | DLY |
| | | 8 | DFR |
| 15 | ERDY | 1 | Device Address |
| | | 2 | D |
| | | 3 | EPTNum |
| | | 4 | NumP |
| | | 5 | StreamId |
| | | 6 | HDr # |
| | | 7 | Hub # |
| | | 8 | DLY |
| | | 9 | DFR |
| 16 | Status | 1 | Route String |
| | | 2 | Device Address |
| | | 3 | D |
| | | 4 | EPTNum |
| | | 5 | PP |
| | | 6 | HDr # |
| | | 7 | Hub # |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| | | 8 | DLY |
| | | 9 | DFR |
| 17 | Stall | 1 | Device Address |
| | | 2 | D |
| | | 3 | EPTNum |
| | | 4 | HDr # |
| | | 5 | Hub # |
| | | 6 | DLY |
| | | 7 | DFR |
| 18 | Function Wake | 1 | Device Address |
| | | 2 | Interface |
| | | 3 | HDr # |
| | | 4 | Hub # |
| | | 5 | DLY |
| | | 6 | DFR |
| 19 | Latency Tolerance | 1 | Device Address |
| | | 2 | BELT |
| | | 3 | HDr # |
| | | 4 | Hub # |
| | | 5 | DLY |
| | | 6 | DFR |
| 20 | Bus Interval Adjust | 1 | Device Address |
| | | 2 | BusIntAdj |
| | | 3 | HDr # |
| | | 4 | Hub # |
| | | 5 | DLY |
| | | 6 | DFR |
| 21 | Host Role Req | 1 | Device Address |
| | | 2 | RSP |
| | | 3 | HDr # |
| | | 4 | Hub # |
| | | 5 | DLY |
| | | 6 | DFR |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| 22 | Sublink Speed | 1 | Device Address |
| | | 2 | TPF |
| | | 3 | LSE |
| | | 4 | ST |
| | | 5 | Lanes |
| | | 6 | LP |
| | | 7 | LSM |
| | | 8 | HDr # |
| | | 9 | Hub # |
| | | 10 | DLY |
| | | 11 | DFR |
| 23 | Ping | 1 | Route String |
| | | 2 | Device Address |
| | | 3 | D |
| | | 4 | EPTNum |
| | | 5 | HDr # |
| | | 6 | Hub # |
| | | 7 | DLY |
| | | 8 | DFR |
| 24 | Ping Rsp | 1 | Device Address |
| | | 2 | D |
| | | 3 | EPTNum |
| | | 4 | HDr # |
| | | 5 | Hub # |
| | | 6 | DLY |
| | | 7 | DFR |
| 25 | Data Packet Header | 1 | Route String |
| | | 2 | Device Address |
| | | 3 | SeqNum |
| | | 4 | EOB |
| | | 5 | D |
| | | 6 | EPTNum |
| | | 7 | TT |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| | | 8 | S |
| | | 9 | Length |
| | | 10 | StreamId |
| | | 11 | SSI |
| | | 12 | WPA |
| | | 13 | DBI |
| | | 14 | PP |
| | | 15 | NBI |
| | | 16 | HDr # |
| | | 17 | Hub # |
| | | 18 | DLY |
| | | 19 | DFR |
| 26 | Data Packet Header n-def | 1 | Route String |
| | | 2 | Device Address |
| | | 3 | SeqNum |
| | | 4 | EOB |
| | | 5 | D |
| | | 6 | EPTNum |
| | | 7 | TT |
| | | 8 | S |
| | | 9 | Length |
| | | 10 | StreamId |
| | | 11 | SSI |
| | | 12 | WPA |
| | | 13 | DBI |
| | | 14 | PP |
| | | 15 | NBI |
| | | 16 | HDr # |
| | | 17 | Hub # |
| | | 18 | DLY |
| | | 19 | DFR |
| | | 20 | Length Rep |
| 27 | Data Packet Payload | 1 | Data |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| 28 | DPP Aborted | 1 | Data |
| 29 | ITP | 1 | Counter |
| | | 2 | Delta |
| | | 3 | Adj Ctrl |
| | | 4 | Corr |
| | | 5 | HDr # |
| | | 6 | Hub # |
| | | 7 | DLY |
| | | 8 | DFR |
| 30 | Link Command | 1 | Sub-type |
| 31 | Skip | 1 | SKP |
| | | 2 | SKPEND |
| | | 3 | LFSR: 3 |
| | | 4 | P |
| | | 5 | LFSR: 2 |
| | | 6 | LFSR: 1 |
| 32 | Idle | - | - |
| 33 | SYNc | 1 | Symbol |
| 34 | SDS | 1 | Symbol |
| | | 2 | Body |

TRIGger<m>:USBThree:ERENable ..... 2181
TRIGger<m>:USBThree:ERRor<n>:ENABle ..... 2181
TRIGger<m>:USBThree:FRENable ..... 2181
TRIGger<m>:USBThree:FRAMe<n>:ENABle ..... 2181
TRIGger<m>:USBThree:FIENable ..... 2181
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:ENABle ..... 2181
TRIGger<m>:USBThree:DOPerator ..... 2182
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DOPerator ..... 2182
TRIGger<m>:USBThree:DMIN ..... 2183
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DMIN ..... 2183
TRIGger<m>:USBThree:DMAX ..... 2183
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DMAX ..... 2183
TRIGger<m>:USBThree:BIT ..... 2183
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:BIT ..... 2183
TRIGger<m>:USBThree:IOPerator ..... 2184
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IOPerator ..... 2184
TRIGger<m>:USBThree:IMIN ..... 2184
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IMIN ..... 2184

TRIGger<m>:USBThree:IMAX.....2185
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IMAX..... 2185

**TRIGger<m>:USBThree:ERENable** <ErrorName>, <Enabler>
**TRIGger<m>:USBThree:ERRor<n>:ENABle** <Enable>

Defines the error type to be triggered on. You can trigger on all error types in parallel.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the error number.

**Parameters:**
<Enable> ON | OFF
*RST: ON

**Example:** Trigger on CRC and length errors:
```
TRIGger:USBThree:ERRor1:ENABle ON
```
Enables CRC Error.
```
TRIGger:USBThree:ERENable "Length Error",ON
```
Enables Length Error.

**Manual operation:** See "Error types: Item name, Enable" on page 849

**TRIGger<m>:USBThree:FRENable** <Frame>, <Enabler>
**TRIGger<m>:USBThree:FRAMe<n>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the frame number.

**Parameters:**
<Enable> ON | OFF
*RST: OFF

**Example:**
```
TRIGger:USBThree:FRAMe1:ENABle ON
```
Enables the checking condition for the TSEQ frame.
```
TRIGger:USBThree:FRENable "TS1" ON
```
Enables the checking condition for the TS1 frame.

**Manual operation:** See "Frame type" on page 849

**TRIGger<m>:USBThree:FIENable** <Frame>, <Field>, <Enabler>
**TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:ENABle** <CondEnabler>

Enables or disables the checking condition for the selected field of the selected frame.

**Suffix:**

<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the frame number.

<o> *
Specifies the field number within the frame.

**Parameters:**

<CondEnabler> ON | OFF

**ON**
Checking condition enabled

**OFF**
Checking condition disabled

*RST: OFF

**Example:** `TRIGger:USBThree:FRAMe1:FLD1:ENABle ON`
Enables the checking condition for the Symbol field of the TSEQ frame.
`TRIGger:USBThree:FRENable "TS1", "Loop" ON`
Enables the checking condition for the Loop field of the TS1 frame.

**Manual operation:** See "Field name / Condition Summary / Enable" on page 850

---

**TRIGger<m>:USBThree:DOPerator** <Frame>, <Field>, <Operator>
**TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DOPerator** <DataOperator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**

<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the field number within the frame.

<o> *

**Parameters:**

<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one endpoint value to be set using `TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of end-point values to be set using `TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DMIN` and `TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DMAX`.

*RST: EQUal

**Manual operation:** See "Data" on page 850

---

**TRIGger<m>:USBThree:DMIN** <Frame>, <Field>, <Data>
**TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DMIN** <DataMin>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the field number within the frame.

<o> *

**Parameters:**
<DataMin>

**Manual operation:** See "Data" on page 850

---

**TRIGger<m>:USBThree:DMAX** <Frame>, <Field>, <Data>
**TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DMAX** <DataMax>

Sets the end value of a data pattern range if `TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DOPerator` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the field number within the frame.

<o> *

**Parameters:**
<DataMax>

**Manual operation:** See "Data" on page 850

---

**TRIGger<m>:USBThree:BIT** <Frame>, <Field>, <Bit>
**TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the field number within the frame.

<o> *

**Parameters:**
<BitState> ONE | ZERO | DC

**ONE**
1

**ZERO**
0

**DC**
"Don't care" (DC) = X

*RST: DC

**Manual operation:** See "Data BitState" on page 851

---

**TRIGger<m>:USBThree:IOPerator** <Frame>, <Field>, <Operator>
**TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IOPerator** <IndexOperator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the field number within the frame.

<o> *

**Parameters:**
<IndexOperator> EQUal | INRange | RANGe

**EQUal**
This condition requires one endpoint value to be set using `TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IMIN`.

**INRange | RANGe**
This condition requires a range of endpoint values to be set using `TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IMIN` and `TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IMAX`.

*RST: INRange

**Manual operation:** See "Index" on page 851

---

**TRIGger<m>:USBThree:IMIN** <Frame>, <Field>, <Data>
**TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IMIN** <IndexMin>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<m> 1..3
<n> *
Specifies the field number within the frame.
<o> *

**Parameters:**
<IndexMin> Specifies the index value or index start value within the field. The index range, increment and *RST values depend on the field type.

**Manual operation:** See "Index" on page 851

---

**TRIGger<m>:USBThree:IMAX** <Frame>, <Field>, <Data>
**TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IMAX** <IndexMax>

Sets the end value of an index range if `TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IOPerator` is set to `INRange` or `RANGe`.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.
<n> *
Specifies the field number within the frame.
<o> *

**Parameters:**
<IndexMax> Specifies the end value for the index range within the field. The index range, increment and *RST values depend on the field type.

**Manual operation:** See "Index" on page 851

### 24.17.20.4 Decode results

In all `BUS<m>:USBThree:RESult:FRAMe<n>` commands, the suffix <m> selects the serial bus and the suffix <n> selects the frame number in the decode table.

BUS<m>:USBThree:RESult:FCOunt? 2186
BUS<m>:USBThree:RESult:FRAMe<n>:INFO? 2186
BUS<m>:USBThree:RESult:FRAMe<n>:TYPE? 2186
BUS<m>:USBThree:RESult:FRAMe<n>:STATe? 2187
BUS<m>:USBThree:RESult:FRAMe<n>:STARt? 2187
BUS<m>:USBThree:RESult:FRAMe<n>:STOP? 2187
BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:TYPE? 2188
BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:STATus? 2188
BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:STARt? 2188
BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:STOP? 2189
BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:VAL? 2189
BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:FVAL? 2189

**BUS<m>:USBThree:RESult:FCOunt?**

Returns the number of decoded frames in the selected serial bus.

**Suffix:**
<m> 1..4

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1

**Usage:** Query only

**BUS<m>:USBThree:RESult:FRAMe<n>:INFO?**

Returns specific frame information details of the selected frame. This information is also shown in the "Info" column of the decode results table.

**Suffix:**
<m> 1..4

<n> *
Specifies the frame number.

**Return values:**
<Info>

**Usage:** Query only

**BUS<m>:USBThree:RESult:FRAMe<n>:TYPE?**

Returns the frame type for the selected serial bus and frame number.

**Suffix:**
<m> 1..4

<n> *
Selects the frame number.

**Return values:**
<FrameType> TSEQ | TS1 | TS1A | TS1B | TS2 | SLF | U2IT | VDT | PCAP | PCNF | PCNR | LDM | ACK | NRDY | ERDY | STATus | STALl | FWAKe | LTM | BIAM | HRR | SSPD | PING | PNGR | DPH | DPHN | DPP | DPPA | ITP | LC | SKIP | BRST | BDAT | BERC | BCNT | IDLE | CP0 | CP1 | CP2 | CP3 | CP5or6 | UHP | ULMP | UTP | UDVN | UDEF | ERRor | CTLB | DATB | SYNC | SDS

For a description of the frame types, see "Frame types" on page 837.

*RST: UDEF

**Usage:** Query only

**BUS<m>:USBThree:RESult:FRAMe<n>:STATe?**

Returns the status of the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *
Selects the frame number.

**Return values:**
<State> OK | INComplete | UNCorrelated | UNKNown | AMB | CRC | LENGth | VOOR | SYMB

**VOOR**
Value out of range

**Usage:** Query only

**BUS<m>:USBThree:RESult:FRAMe<n>:STARt?**

Returns the start time of the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *
Selects the frame number.

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
Default unit: s

**Usage:** Query only

**BUS<m>:USBThree:RESult:FRAMe<n>:STOP?**

Returns the stop time of the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *
Selects the frame number.

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
Default unit: s

**Usage:** Query only

**BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:TYPE?**

Returns the type of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**

<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**

<FieldType>

**Usage:** Query only

**BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:STATus?**

Returns the status of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**

<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**

<FieldState> OK | INComplete | UNCorrelated | UNKNown | AMB | CRC | LENGth | VOOR | SYMB

**VOOR**
Value out of range

**Usage:** Query only

**BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:STARt?**

Returns the start time of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**

<m> 1..4

<n> *

<o> *

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:STOP?**

Returns the stop time of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *

<o> *

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:VAL?**

Returns the value of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**
<FieldValue>

**Usage:** Query only

---

**BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:FVAL?**

Returns the formatted value of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**
<ValueFormatted>

**Usage:** Query only

### 24.17.20.5 Search settings

There are two commands for each parameter, that you can use for defining the SSIC search settings.

For example, to set the *Frame type =ACK > Field =Route String >Data* value you can use one of the following commands:

- `SEARch:TRIGger:USBThree:FRAMe13:FLD1:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number. For an overview, see Table 24-14.
- `SEARch:TRIGger:USBThree:DMIN "ACK", "Route String", 01100`
  Defines the parameter by using the frame and field name.

SEARch:TRIGger:USBThree:ERENable....2190
SEARch:TRIGger:USBThree:ERRor<m>:ENABle....2190
SEARch:TRIGger:USBThree:FRENable....2191
SEARch:TRIGger:USBThree:FRAMe<m>:ENABle....2191
SEARch:TRIGger:USBThree:FIENable....2191
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:ENABle....2191
SEARch:TRIGger:USBThree:DOPerator....2192
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DOPerator....2192
SEARch:TRIGger:USBThree:DMIN....2193
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DMIN....2193
SEARch:TRIGger:USBThree:DMAX....2193
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DMAX....2193
SEARch:TRIGger:USBThree:BIT....2194
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:BIT....2194
SEARch:TRIGger:USBThree:IOPerator....2194
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IOPerator....2194
SEARch:TRIGger:USBThree:IMIN....2195
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMIN....2195
SEARch:TRIGger:USBThree:IMAX....2195
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMAX....2195

---

**SEARch:TRIGger:USBThree:ERENable** <SearchName>,<ErrorName>, <Enabler>
**SEARch:TRIGger:USBThree:ERRor<m>:ENABle** <SearchName>,<Enable>
**SEARch:TRIGger:USBThree:ERRor<m>:ENABle?** <SearchName>

Enables the search for errors. You can search for all error types in parallel.

**Suffix:**
<m> *
Specifies the error.

**Parameters:**
<Enable> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Example:** Searches for CRC and length errors:
`SEARch:TRIGger:USBThree:ERRor1:ENABle ON`
Enables searching for CRC Error.
`SEARch:TRIGger:USBThree:ERENable "Length Error",ON`
Enables searching for Length Error.

**Manual operation:** See "Search criteria" on page 854

---

**SEARch:TRIGger:USBThree:FRENable** <SearchName>,<Frame>, <Enabler>
**SEARch:TRIGger:USBThree:FRAMe<m>:ENABle** <SearchName>,<Enable>
**SEARch:TRIGger:USBThree:FRAMe<m>:ENABle?** <SearchName>

Defines the frame type to be searched for.

**Suffix:**
<m> *

**Parameters:**
<Enable> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Example:** `SEARch:TRIGger:USBThree:FRAMe1:ENABle ON`
Enables searching for the TSEQ frame.
`SEARch:TRIGger:USBThree:FRENable "TS1" ON`
Enables searching for the TS1 frame.

**Manual operation:** See "Search criteria" on page 854

---

**SEARch:TRIGger:USBThree:FIENable** <SearchName>,<Frame>, <Field>, <Enabler>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:ENABle** <SearchName>,<CondEnabler>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:ENABle?** <SearchName>

Enables or disables the checking condition for searching a specific field of the selected frame.

**Suffix:**
<m> *
Specifies the frame.

<n> *
Specifies the field number within the frame.

**Parameters:**
<CondEnabler> ON | OFF

**ON**
Checking condition enabled

**OFF**
Checking condition disabled

*RST: OFF

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Example:** `SEARch:TRIGger:USBThree:FRAMe1:FLD1:ENABle ON`
Enables searching for the Symbol field of the TSEQ frame.
`SEARch:TRIGger:USBThree:FRENable "TS1", "Loop" ON`
Enables searching for the Loop field of the TS1 frame.

**Manual operation:** See "Search criteria" on page 854

---

**SEARch:TRIGger:USBThree:DOPerator** <SearchName>,<Frame>, <Field>, <Operator>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DOPerator** <SearchName>,<DataOperator>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DOPerator?** <SearchName>

Sets the operator to set a specific data pattern to be searched in the selected field of the selected frame.

**Suffix:**
<m> *
Specifies the field number within the frame.

<n> *

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one endpoint value to be set using `SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of end-point values to be set using `SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMIN` and `SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMAX` on page 2195.

*RST: EQUal

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search criteria" on page 854

---

**SEARch:TRIGger:USBThree:DMIN** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DMIN**
<SearchName>,<DataMin>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DMIN?** <SearchName>

Specifies the data pattern to be searched, or sets the start value of a data pattern range to be searched.

**Suffix:**
<m> *
Specifies the field number within the frame.

<n> *

**Parameters:**
<DataMin> Specifies the name of the generic trigger setting frame, see `SEARch:RESult:USBThree:FRAMe<m>:TYPE?` on page 2196.

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search criteria" on page 854

---

**SEARch:TRIGger:USBThree:DMAX** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DMAX**
<SearchName>,<DataMax>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DMAX?** <SearchName>

Sets the end value of a data pattern range if `SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DOPerator` is set to `INRange` or `OORange`.

**Suffix:**
<m> *
Specifies the field number within the frame.

<n> *

**Parameters:**
<DataMax> Specifies the name of the generic trigger setting frame, see `SEARch:RESult:USBThree:FRAMe<m>:TYPE?` on page 2196.

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search criteria" on page 854

---

**SEARch:TRIGger:USBThree:BIT** <SearchName>,<Frame>, <Field>, <Bit>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:BIT** <SearchName>,<BitState>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:BIT?** <SearchName>

Sets the bit state of a field to be searched that only consists of one bit.

**Suffix:**
<m> *
Specifies the frame.

<n> *
Specifies the field number within the frame.

**Parameters:**
<BitState> ONE | ZERO | DC

**ONE**
1

**ZERO**
0

**DC**
"Don't care" (DC) = X

*RST: DC

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search criteria" on page 854

---

**SEARch:TRIGger:USBThree:IOPerator** <SearchName>,<Frame>, <Field>, <Operator>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IOPerator** <SearchName>,<IndexOperator>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IOPerator?** <SearchName>

Sets the operator for the index for searching in the selected field of the selected frame.

**Suffix:**
<m> *
Specifies the field number within the frame.

<n> *

**Parameters:**
<IndexOperator> EQUal | INRange | RANGe

**EQUal**
This condition requires one endpoint value to be set using `SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMIN`.

**INRange | RANGe**
This condition requires a range of endpoint values to be set using `SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMIN` and `SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMAX`.

*RST: INRange

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search criteria" on page 854

---

**SEARch:TRIGger:USBThree:IMIN** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMIN** <SearchName>,<IndexMin>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMIN?** <SearchName>

Specifies the index at which the data is to be searched, or sets the start value of an index range in which the data is to be searched.

**Suffix:**
<m> *
Specifies the field number within the frame.

<n> *

**Parameters:**
<IndexMin> Specifies the index value or the start value of an index range within the field.
The index range, increment and *RST values depend on the field type.

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search criteria" on page 854

---

**SEARch:TRIGger:USBThree:IMAX** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMAX**
<SearchName>,<IndexMax>
**SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMAX?** <SearchName>

Sets the end value of an index range if `SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IOPerator` is set to `INRange` or `RANGe`.

**Suffix:**
<m> *
Specifies the field number within the frame.

<n> *

**Parameters:**
<IndexMax> Specifies the end value for the index range within the field.
The index range, increment and *RST values depend on the field type.

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search criteria" on page 854

### 24.17.20.6 Search results

In all `SEARch:RESult:USBThree:FRAMe<m>` commands, the suffix <m> selects the frame number in the list of search results.

SEARch:RESult:USBThree:FRAMe<m>:TYPE? 2196
SEARch:RESult:USBThree:FRAMe<m>:INFO? 2196
SEARch:RESult:USBThree:FCOunt? 2197
SEARch:RESult:USBThree:FRAMe<m>:STATe? 2197
SEARch:RESult:USBThree:FRAMe<m>:STARt? 2197
SEARch:RESult:USBThree:FRAMe<m>:STOP? 2198
SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:TYPE? 2198
SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:STATus? 2198
SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:STARt? 2199
SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:STOP? 2199
SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:VAL? 2199
SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:FVAL? 2200

---

**SEARch:RESult:USBThree:FRAMe<m>:TYPE?** <SearchName>

Returns the frame type for the selected frame number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameType> TSEQ | TS1T | TS2T | SLF | U2IT | VDT | PCAP | PCNF | PCNR | LDM | ACK | NRDY | ERDY | STATus | STALl | FWAKe | LTM | BIAM | HRR | SSPD | PING | PNGR | DPH | DPP | DPPA | ITP | LC | SKIP | BRST | BDAT | BERC | BCNT | IDLE | UHP | ULMP | UTP | UDVN | UDEF | ERRor

For a description of the frame types, see "Frame types" on page 837.

**Usage:** Query only

---

**SEARch:RESult:USBThree:FRAMe<m>:INFO?** <SearchName>

Returns specific frame information details of the selected frame in the search result. This information is also shown in the "Info" column of the decode results table.

**Suffix:**
<m> *

Specifies the frame number.

**Query parameters:**
<SearchName>

**Return values:**
<Info>

**Usage:** Query only

---

**SEARch:RESult:USBThree:FCOunt?** <SearchName>

Returns the number of decoded frames in the search result.

**Query parameters:**
<SearchName>

**Return values:**
<FrameCount>

**Usage:** Query only

---

**SEARch:RESult:USBThree:FRAMe<m>:STATe?** <SearchName>

Returns the status of the frame with the selected frame number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | INComplete | UNCorrelated | UNKNown | AMB | CRC | LENGth | VOOR | SYMB

**VOOR**
Value out of range

**Usage:** Query only

---

**SEARch:RESult:USBThree:FRAMe<m>:STARt?** <SearchName>

Returns the start time of the frame with the selected frame number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:USBThree:FRAMe<m>:STOP?** <SearchName>

Returns the stop time of the frame with the selected frame number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:TYPE?** <SearchName>

Returns the field name of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *
Selects the field number.

**Query parameters:**
<SearchName>

**Return values:**
<FieldType>

**Usage:** Query only

---

**SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:STATus?** <SearchName>

Returns the field status of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *
Selects the field number.

**Query parameters:**
<SearchName>

**Return values:**
<FieldState> OK | INComplete | UNCorrelated | UNKNown | AMB | CRC | LENGth | VOOR | SYMB

**VOOR**
Value out of range

**Usage:** Query only

---

**SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:STARt?** <SearchName>

Returns the start time of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:STOP?** <SearchName>

Returns the stop time of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:VAL?** <SearchName>

Returns the value of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *
Selects the field number.

**Query parameters:**
<SearchName>

**Return values:**
<ByteValue>

**Usage:** Query only

---

**SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:FVAL?** <SearchName>

Returns the formatted value of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *
Selects the field number.

**Query parameters:**
<SearchName>

**Return values:**
<ValueFormatted>

**Usage:** Query only

## 24.17.21 USBPD (option R&S RTP-K63)


- Configuration....2200
- Trigger....2203
- Decode results....2209
- Search settings....2214
- Search results....2220


### 24.17.21.1 Configuration

In all `BUS<m>:USBPd` commands, the suffix <m> selects the serial bus.


BUS<m>:USBPd:DETail....2201
BUS<m>:USBPd:HYSTeresis....2201
BUS<m>:USBPd:SOURce....2201
BUS<m>:USBPd:THReshold....2201
BUS<m>:USBPd:THRBottom....2202
BUS<m>:USBPd:THRMid....2202
BUS<m>:USBPd:THRTop....2202

**BUS<m>:USBPd:DETail** <Detail>

If enabled, the data words are broken down into subframes. If not enabled the data words are displayed as 32-bit data words.

**Suffix:**
<m> 1..4

**Parameters:**
<Detail> ON | OFF
*RST: OFF

**Manual operation:** See "Show decode detail" on page 860

**BUS<m>:USBPd:HYSTeresis** <Hysteresis>

Sets a value for the hysteresis of the data.

**Suffix:**
<m> 1..4

**Parameters:**
<Hysteresis> Range: -10 to 10
Increment: 0.01
*RST: 0.05
Default unit: V

**Manual operation:** See "Hysteresis" on page 859

**BUS<m>:USBPd:SOURce** <SourceData>

Selects the source for the data signal in the USBPD protocol.

For triggering on a serial bus, analog input channels are required.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceData> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "Data" on page 859

**BUS<m>:USBPd:THReshold** <Threshold>

Sets the threshold value of the data.

**Suffix:**
<m> 1..4

**Parameters:**

<Threshold> Range: -10 to 10
Increment: 0.01
*RST: 0.6
Default unit: V

**Manual operation:** See "vData" on page 859

---

**BUS<m>:USBPd:THRBottom** <ThresholdBot>

Sets the bottom threshold (for the low current ) for the current advertisment mode.

**Suffix:**

<m> 1..4

**Parameters:**

<ThresholdBot> Range: -10 to 10
Increment: 0.01
*RST: 0.2
Default unit: V

**Manual operation:** See "Current advertisment thresholds" on page 859

---

**BUS<m>:USBPd:THRMid** <ThresholdMid>

Sets the middle threshold (at default USB Type-C currnet) for the current advertisment mode.

**Suffix:**

<m> 1..4

**Parameters:**

<ThresholdMid> Range: -10 to 10
Increment: 0.01
*RST: 0.66
Default unit: V

**Manual operation:** See "Current advertisment thresholds" on page 859

---

**BUS<m>:USBPd:THRTop** <ThresholdTop>

Sets the top threshold (at USB Type-C currnet of 1.5 A ) for the current advertisment mode.

**Suffix:**

<m> 1..4

**Parameters:**

<ThresholdTop> Range: -10 to 10
Increment: 0.01
*RST: 1.23
Default unit: V

**Manual operation:** See "Current advertisment thresholds" on page 859

### 24.17.21.2 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

There are two commands for each parameter, that you can use for defining the USBPD settings.

For example, to set the *Frame type =Control Frame > Field =MsgID >Data* value you can use one of the following commands:

- `TRIGger:USBPD:FRAMe3:FLD3:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number. For an overview, see Overview of frame and field numbers for USBPD.
- `TRIGger:USBPD:DMIN "Control", "MsgID", 01100`
  Defines the parameter by using the frame and field name.

Table 24-16 gives an overview of the available frame and field numbers. Those numbers follow the GUI numbers.

***Table 24-16: Overview of frame and field numbers for USBPD***

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| 1 | Test Frame | - | - |
| 2 | Reset | - | - |
| 3 | Control frame | 1 | Extended |
| | | 2 | NumDataObjs |
| | | 3 | MsgID |
| | | 4 | PwrRole/Plug |
| | | 5 | Rev |
| | | 6 | DataRole |
| | | 7 | MsgType |
| 4 | Data Frame | 1 | Extended |
| | | 2 | NumDataObjs |
| | | 3 | MsgID |
| | | 4 | PwrRole/Plug |

| Frame number | Frame name | Field number | Field name |
|---|---|---|---|
| | | 5 | Rev |
| | | 6 | DataRole |
| | | 7 | MsgType |
| | | 8 | DATA |
| 5 | Extended | 1 | Extended |
| | | 2 | NumDataObjs |
| | | 3 | MsgID |
| | | 4 | PwrRole/Plug |
| | | 5 | Rev |
| | | 6 | DataRole |
| | | 7 | MsgType |
| | | 8 | Chunked |
| | | 9 | Chunk Num |
| | | 10 | Req Chunk |
| | | 11 | Data Size |
| | | 12 | DATA |

TRIGger<m>:USBPd:ERENable....2205
TRIGger<m>:USBPd:ERRor<n>:ENABle....2205
TRIGger<m>:USBPd:FRENable....2205
TRIGger<m>:USBPd:FRAMe<n>:ENABle....2205
TRIGger<m>:USBPd:FIENable....2205
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:ENABle....2205
TRIGger<m>:USBPd:BIT....2206
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:BIT....2206
TRIGger<m>:USBPd:DMAX....2207
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DMAX....2207
TRIGger<m>:USBPd:DMIN....2207
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DMIN....2207
TRIGger<m>:USBPd:DOPerator....2207
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DOPerator....2207
TRIGger<m>:USBPd:IMAX....2208
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IMAX....2208
TRIGger<m>:USBPd:IMIN....2208
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IMIN....2208
TRIGger<m>:USBPd:IOPerator....2209
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IOPerator....2209

**TRIGger<m>:USBPd:ERENable** <ErrorName>, <Enabler>
**TRIGger<m>:USBPd:ERRor<n>:ENABle** <Enable>

Enables or disables the checking condition for a specific error in the selected field of the selected frame.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the error number.

**Parameters:**
<Enable> ON | OFF
*RST: ON

**Example:** `TRIGger:USBPd:ERRor1:ENABle ON`
Enables CRC Error.
`TRIGger:USBPd:ERENable "Length Error",ON`
Enables Length Error.

**Manual operation:** See "Error > Enable /Item" on page 863

**TRIGger<m>:USBPd:FRENable** <Frame>, <Enabler>
**TRIGger<m>:USBPd:FRAMe<n>:ENABle** <Enable>

Enables or disables the checking condition for the selected frame.

**Suffix:**
<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the frame number.

**Parameters:**
<Enable> ON | OFF
*RST: OFF

**Example:** `TRIGger:USBPd:FRAMe1:ENABle ON`
Enables searching for for the Test frame.
`TRIGger:USBPd:FRENable "Control Frame" ON`
Enables searching for the Control frame.

**Manual operation:** See "Frame Type" on page 862

**TRIGger<m>:USBPd:FIENable** <Frame>, <Field>, <Enabler>
**TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:ENABle** <CondEnabler>

Enables or disables the checking condition for a specific data pattern in the selected field of the selected frame.

**Suffix:**

<m> 1..3
Only 1 = A-trigger, 2 | 3 = not available. Can be omitted.

<n> *
Specifies the frame.

<o> *
Specifies the field number within the frame.

**Parameters:**

<CondEnabler> ON | OFF

**ON**
Checking condition enabled

**OFF**
Checking condition disabled

*RST: OFF

**Example:** `TRIGger:USBPd:FRAMe3:FLD1:ENABle ON`
Enables the checking condition for the Extended field of the Control frame.
`TRIGger:USBPd:FRENable "Data Frame", "Data Role" ON`
Enables the checking condition for the Data Role field of the Data frame.

**Manual operation:** See " Enable/ Field name/ Condition" on page 863

---

**TRIGger<m>:USBPd:BIT** <Frame>, <Field>, <Bit>
**TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:BIT** <BitState>

Sets the bit state of a field that only consists of one bit.

**Suffix:**

<m> 1..3

<n> *

<o> *
Specifies the field number within the frame.

**Parameters:**

<BitState> ONE | ZERO | DC

**ONE**
1

**ZERO**
0

**DC**
"Don't care" (DC) = X

*RST: DC

**Manual operation:** See "Data BitState" on page 864

**TRIGger<m>:USBPd:DMAX** <Frame>, <Field>, <Data>
**TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DMAX** <DataMax>

Sets the end value of a data pattern range if `TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DMAX` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3
<n> *
Specifies the field number within the frame.
<o> *

**Parameters:**
<DataMax> Specifies the name of the generic trigger setting frame, see `BUS<m>:USBPd:RESult:FRAMe<n>:TYPE?` on page 2214.

**Manual operation:** See "Data" on page 863

**TRIGger<m>:USBPd:DMIN** <Frame>, <Field>, <Data>
**TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DMIN** <DataMin>

Specifies the data pattern, or sets the start value of a data pattern range.

**Suffix:**
<m> 1..3
<n> *
Specifies the field number within the frame.
<o> *

**Parameters:**
<DataMin>

**Manual operation:** See "Data" on page 863

**TRIGger<m>:USBPd:DOPerator** <Frame>, <Field>, <Operator>
**TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DOPerator** <DataOperator>

Sets the operator for the data pattern in the selected field of the selected frame.

**Suffix:**
<m> 1..3
<n> *
Specifies the field number within the frame.
<o> *

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one endpoint value to be set using `TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of endpoint values to be set using `TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DMIN` and `TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DMAX`.

*RST: EQUal

**Manual operation:** See "Data" on page 863

---

**TRIGger<m>:USBPd:IMAX** <Frame>, <Field>, <Data>
**TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IMAX** <IndexMax>

Sets the end value of an index range if `TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IOPerator` on page 2209 is set to `INRange` or `RANGe`.

**Suffix:**
<m> 1..3
<n> *
Specifies the field number within the frame.
<o> *

**Parameters:**
<IndexMax> Specifies the end value for the index range within the field. The index range, increment and *RST values depend on the field type.

**Manual operation:** See "Index" on page 864

---

**TRIGger<m>:USBPd:IMIN** <Frame>, <Field>, <Data>
**TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IMIN** <IndexMin>

Specifies the index, or sets the start value of an index range.

**Suffix:**
<m> 1..3
<n> *
Specifies the field number within the frame.
<o> *

**Parameters:**
<IndexMin> Specifies the index value or index start value within the field. The index range, increment and *RST values depend on the field type.

**Manual operation:** See "Index" on page 864

**TRIGger<m>:USBPd:IOPerator** <Frame>, <Field>, <Operator>
**TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IOPerator** <IndexOperator>

Sets the operator for the index in the selected field of the selected frame.

**Suffix:**
<m> 1..3
<n> *
<o> *

**Parameters:**
<IndexOperator> EQUal | INRange | RANGe

**EQUal**
This condition requires one endpoint value to be set using `TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IMIN`.

**INRange | RANGe**
This condition requires a range of endpoint values to be set using `TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IMIN` and `TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IMAX`.

*RST: INRange

**Manual operation:** See "Index" on page 864

### 24.17.21.3 Decode results

In all `BUS<m>:USBPd:RESult:FRAMe<n>` commands, the suffix <m> selects the serial bus and the suffix <n> selects the frame number in the decode table.

BUS<m>:USBPd:RESult:FCOunt? 2209
BUS<m>:USBPd:RESult:FRAMe<n>:DATA? 2210
BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:FVAL? 2210
BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:STATus? 2210
BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:STARt? 2211
BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:STOP? 2211
BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:TYPE? 2212
BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:VAL? 2212
BUS<m>:USBPd:RESult:FRAMe<n>:INFO? 2212
BUS<m>:USBPd:RESult:FRAMe<n>:STARt? 2213
BUS<m>:USBPd:RESult:FRAMe<n>:STATe? 2213
BUS<m>:USBPd:RESult:FRAMe<n>:STOP? 2213
BUS<m>:USBPd:RESult:FRAMe<n>:TYPE? 2214

**BUS<m>:USBPd:RESult:FCOunt?**

Returns the number of decoded frames in the selected serial bus.

**Suffix:**
<m> 1..4

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1

**Usage:** Query only

---

**BUS<m>:USBPd:RESult:FRAMe<n>:DATA?**

Returns the data from the frame with the selected frame number (for the selected serial bus). This is the hex value of the first four bytes of data fields, prefixed by the total number of data fields (in square brackets), e.g. `[5]FF008041`.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Data>

**Usage:** Query only

---

**BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:FVAL?**

Returns the formatted value of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**
<FrameByteValue>

**Usage:** Query only

---

**BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:STATus?**

Returns the status of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**

<FieldState> OK | UNKNown | INComplete | PREamble | SOP | CRC | LENGth | FBFB

**SOP**
Start of packet warning

**CRC**
Cyclic redundancy check error

**FBFB**
4b5b error

*RST: OK

**Usage:** Query only

---

**BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:STARt?**

Returns the start time of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**

<m> 1..4

<n> *

<o> *

**Return values:**

<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:STOP?**

Returns the stop time of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**

<m> 1..4

<n> *

<o> *

**Return values:**

<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:TYPE?**

Returns the type of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**
<FieldType>

**Usage:** Query only

---

**BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:VAL?**

Returns the value of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**
<FrameByteValue>

**Usage:** Query only

---

**BUS<m>:USBPd:RESult:FRAMe<n>:INFO?**

Returns information on the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Info>

**Usage:** Query only

**BUS<m>:USBPd:RESult:FRAMe<n>:STARt?**

Returns the start time of the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:USBPd:RESult:FRAMe<n>:STATe?**

Returns the status of the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<State> OK | UNKNown | INComplete | PREamble | SOP | CRC | LENGth | FBFB

**SOP**
Start of packet warning

**CRC**
Cyclic redundancy check error

**FBFB**
4b5b error

*RST: OK

**Usage:** Query only

**BUS<m>:USBPd:RESult:FRAMe<n>:STOP?**

Returns the stop time of the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *

**Return values:**

<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:USBPd:RESult:FRAMe<n>:TYPE?**

Returns the frame type for the selected serial bus and frame number.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<FrameType> STARt | RESet | CTRL | DATA | BIST | RQST | SINK | SRC | VEND | BATT | ALRT | XMSG | XSRC | XSTA | XGBC | XGBS | XBAC | XGMI | XMFI | XSRQ | XSRS | XFRQ | XFRS | TEST | LOWP | UNKNown

For a description of the frame types, see "Frame packet types" on page 857.

*RST: UNKNown

**Usage:** Query only

### 24.17.21.4 Search settings

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

There are two commands for each parameter, that you can use for defining the USBPD settings.

For example, to set the *Frame type =Control Frame > Field =MsgID >Data* value you can use one of the following commands:

- `SEARch:TRIGger:USBPD:FRAMe3:FLD3:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number. For an overview, see Table 24-16.
- `SEARch:TRIGger:USBPD:DMIN "Control", "MsgID", 01100`
  Defines the parameter by using the frame and field name.

SEARch:TRIGger:USBPd:ERENable ..... 2215
SEARch:TRIGger:USBPd:ERRor<m>:ENABle ..... 2215
SEARch:TRIGger:USBPd:FRENable ..... 2216
SEARch:TRIGger:USBPd:FRAMe<m>:ENABle ..... 2216
SEARch:TRIGger:USBPd:FIENable ..... 2216
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:ENABle ..... 2216
SEARch:TRIGger:USBPd:BIT ..... 2217
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:BIT ..... 2217
SEARch:TRIGger:USBPd:DMAX ..... 2217
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMAX ..... 2217
SEARch:TRIGger:USBPd:DMIN ..... 2218
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMIN ..... 2218
SEARch:TRIGger:USBPd:DOPerator ..... 2218
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DOPerator ..... 2218
SEARch:TRIGger:USBPd:IMAX ..... 2219
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMAX ..... 2219
SEARch:TRIGger:USBPd:IMIN ..... 2219
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMIN ..... 2219
SEARch:TRIGger:USBPd:IOPerator ..... 2220
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IOPerator ..... 2220

---

**SEARch:TRIGger:USBPd:ERENable** <SearchName>,<ErrorName>, <Enabler>
**SEARch:TRIGger:USBPd:ERRor<m>:ENABle** <SearchName>,<Enable>
**SEARch:TRIGger:USBPd:ERRor<m>:ENABle?** <SearchName>

Enables or disables the checking condition for searching a specific error in the selected field of the selected frame.

**Suffix:**
<m> *
Specifies the error number.

**Parameters:**
<Enable> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Example:** `SEARch:TRIGger:USBPd:ERRor1:ENABle ON`
Enables searching for CRC Error.
`SEARch:TRIGger:USBPd:ERENable "Length Error",ON`
Enables searching for Length Error.

**Manual operation:** See "Search" on page 867

**SEARch:TRIGger:USBPd:FRENable** <SearchName>,<Frame>, <Enabler>
**SEARch:TRIGger:USBPd:FRAMe<m>:ENABle** <SearchName>,<Enable>
**SEARch:TRIGger:USBPd:FRAMe<m>:ENABle?** <SearchName>

**Suffix:**

<m> *

Specifies the frame number.

**Parameters:**

<Enable> ON | OFF

*RST: OFF

**Parameters for setting and query:**

<SearchName>

**Example:** `SEARch:TRIGger:USBPd:FRAMe1:ENABle ON`

Enables the checking condition for the Test frame.

`SEARch:TRIGger:USBPd:FRENable "Control Frame" ON`

Enables the checking condition for the Control frame.

---

**SEARch:TRIGger:USBPd:FIENable** <SearchName>,<Frame>, <Field>, <Enabler>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:ENABle**
<SearchName>,<CondEnabler>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:ENABle?** <SearchName>

Enables or disables the checking condition for searching a specific data pattern in the selected field of the selected frame.

**Suffix:**

<m> *

Specifies the frame.

<n> *

Specifies the field number within the frame.

**Parameters:**

<CondEnabler> ON | OFF

**ON**
Checking condition enabled

**OFF**
Checking condition disabled

*RST: OFF

**Parameters for setting and query:**

<SearchName> String with the name of the search.

**Example:** `SEARch:TRIGger:USBPd:FRAMe3:FLD1:ENABle ON`

Enables searching for the Extended field of the Control frame.

`SEARch:TRIGger:USBPd:FRENable "Data Frame", "Data Role" ON`

Enables searching for the Data Role field of the Data frame.

**SEARch:TRIGger:USBPd:BIT** <SearchName>,<Frame>, <Field>, <Bit>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:BIT** <SearchName>,<BitState>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:BIT?** <SearchName>

Sets the bit state of a field to be searched that only consists of one bit.

**Suffix:**
<m> *
Specifies the frame.

<n> *
Specifies the field number within the frame.

**Parameters:**
<BitState> ONE | ZERO | DC

**ONE**
1

**ZERO**
0

**DC**
"Don't care" = X

*RST: DC

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search" on page 867

---

**SEARch:TRIGger:USBPd:DMAX** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMAX** <SearchName>,<DataMax>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMAX?** <SearchName>

Sets the end value of a data pattern range if `SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DOPerator` is set to `INRange` or `OORange`.

**Suffix:**
<m> *
Specifies the frame.

<n> *
Specifies the field number within the frame.

**Parameters:**
<DataMax> Specifies the name of the generic trigger setting frame, see `SEARch:RESult:USBPd:FRAMe<m>:TYPE?` on page 2225.

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search" on page 867

**SEARch:TRIGger:USBPd:DMIN** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMIN** <SearchName>,<DataMin>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMIN?** <SearchName>

Specifies the data pattern to be searched, or sets the start value of a data pattern range to be searched.

**Suffix:**
<m> *
Specifies the frame.

<n> *
Specifies the field number within the frame.

**Parameters:**
<DataMin> Specifies the name of the generic trigger setting frame, see `SEARch:RESult:USBPd:FRAMe<m>:TYPE?` on page 2225.

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search" on page 867

**SEARch:TRIGger:USBPd:DOPerator** <SearchName>,<Frame>, <Field>, <Operator>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DOPerator**
<SearchName>,<DataOperator>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DOPerator?** <SearchName>

Sets the operator to set a specific data pattern to be searched in the selected field of the selected frame.

**Suffix:**
<m> *
Specifies the frame.

<n> *
Specifies the field number within the frame.

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one endpoint value to be set using `SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of endpoint values to be set using `SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMIN` and `SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search" on page 867

---

**SEARch:TRIGger:USBPd:IMAX** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMAX** <SearchName>,<IndexMax>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMAX?** <SearchName>

Sets the end value of an index range if `SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IOPerator` is set to `INRange` or `RANGe`.

**Suffix:**
<m> *
Specifies the frame.

<n> *
Specifies the field number within the frame.

**Parameters:**
<IndexMax> Specifies the end value for the index range within the field.
The index range, increment and *RST values depend on the field type.

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search" on page 867

---

**SEARch:TRIGger:USBPd:IMIN** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMIN** <SearchName>,<IndexMin>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMIN?** <SearchName>

Specifies the index at which the data is to be searched, or sets the start value of an index range in which the data is to be searched.

**Suffix:**
<m> *
Specifies the frame.

<n> *
Specifies the field number within the frame.

**Parameters:**
<IndexMin> Specifies the index value or the start value of an index range within the field.
The index range, increment and *RST values depend on the field type.

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search" on page 867

**SEARch:TRIGger:USBPd:IOPerator** <SearchName>,<Frame>, <Field>, <Operator>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IOPerator**
<SearchName>,<IndexOperator>
**SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IOPerator?** <SearchName>

Sets the operator for the index for searching in the selected field of the selected frame.

**Suffix:**
<m> *
Specifies the frame.

<n> *
Specifies the field number within the frame.

**Parameters:**
<IndexOperator> EQUal | INRange | RANGe

**EQUal**
This condition requires one endpoint value to be set using `SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMIN`.

**INRange | RANGe**
This condition requires a range of endpoint values to be set using `SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMIN` and `SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMAX`.

*RST: INRange

**Parameters for setting and query:**
<SearchName> String with the name of the search.

**Manual operation:** See "Search" on page 867

### 24.17.21.5 Search results

In all `SEARch:RESult:USBPd:FRAMe<m>` commands, the suffix <m> selects the frame number in the list of search results.

SEARch:RESult:USBPd:FCOunt? 2220
SEARch:RESult:USBPd:FRAMe<m>:DATA? 2221
SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:FVAL? 2221
SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:STATus? 2221
SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:STARt? 2222
SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:STOP? 2222
SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:TYPE? 2223
SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:VAL? 2223
SEARch:RESult:USBPd:FRAMe<m>:INFO? 2223
SEARch:RESult:USBPd:FRAMe<m>:STARt? 2224
SEARch:RESult:USBPd:FRAMe<m>:STATe? 2224
SEARch:RESult:USBPd:FRAMe<m>:STOP? 2225
SEARch:RESult:USBPd:FRAMe<m>:TYPE? 2225

**SEARch:RESult:USBPd:FCOunt?** <SearchName>

Returns the number of decoded frames in the search result.

**Query parameters:**
<SearchName>

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:USBPd:FRAMe<m>:DATA?** <SearchName>

Searches for data from the frame with the selected frame number (for the selected serial bus). This is the hex value of the first four bytes of data fields, prefixed by the total number of data fields (in square brackets), e.g. `[5]FF008041`.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

---

**SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:FVAL?** <SearchName>

Returns the formatted value of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *
Selects the field number.

**Query parameters:**
<SearchName>

**Return values:**
<ValueFormatted>

**Usage:** Query only

---

**SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:STATus?** <SearchName>

Returns the field status of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *
Selects the field number.

**Query parameters:**
<SearchName>

**Return values:**
<FieldState> OK | UNKNown | INComplete | PREamble | SOP | CRC | LENGth | FBFB

**SOP**
Start of packet warning

**CRC**
Cyclic redundancy check error

**FBFB**
4b5b error

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:STARt?** <SearchName>

Returns the start time of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:STOP?** <SearchName>

Returns the stop time of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:TYPE?** <SearchName>

Returns the field name of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *
Selects the field number.

**Query parameters:**
<SearchName>

**Return values:**
<FieldType>

**Usage:** Query only

---

**SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:VAL?** <SearchName>

Returns the value of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *
Selects the field number.

**Query parameters:**
<SearchName>

**Return values:**
<ByteValue>

**Usage:** Query only

---

**SEARch:RESult:USBPd:FRAMe<m>:INFO?** <SearchName>

Returns specific frame information details of the selected frame in the search result. This information is also shown in the "Info" column of the decode results table.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Info>

**Usage:** Query only

---

**SEARch:RESult:USBPd:FRAMe<m>:STARt?** <SearchName>

Returns the start time of the frame with the selected frame number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:USBPd:FRAMe<m>:STATe?** <SearchName>

Returns the status of the frame with the selected frame number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | UNKNown | INComplete | PREamble | SOP | CRC | LENGth | FBFB

**SOP**
Start of packet warning

**CRC**
Cyclic redundancy check error

**FBFB**
4b5b error

*RST: OK

**Usage:** Query only

**SEARch:RESult:USBPd:FRAMe<m>:STOP?** <SearchName>

Returns the stop time of the frame with the selected frame number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
Default unit: s

**Usage:** Query only

**SEARch:RESult:USBPd:FRAMe<m>:TYPE?** <SearchName>

Returns the frame type for the selected frame number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameType> STARt | RESet | CTRL | DATA | BIST | RQST | SINK | SRC | VEND | BATT | ALRT | XMSG | XSRC | XSTA | XGBC | XGBS | XBAC | XGMI | XMFI | XSRQ | XSRS | XFRQ | XFRS | TEST | LOWP | UNKNown

For a description of the frame types, see "Frame packet types" on page 857.

*RST: UNKNown

**Usage:** Query only

## 24.17.22 Space wire (option R&S RTP-K65)


- Configuration....2226
- Trigger....2229
- Decode results....2232
- Search settings....2237
- Search results....2241

### 24.17.22.1 Configuration

BUS<m>:SWIRe:BPOSition....2226
BUS<m>:SWIRe:DATA:HYSTeresis....2226
BUS<m>:SWIRe:DATA:SOURce....2226
BUS<m>:SWIRe:DATA:THReshold....2227
BUS<m>:SWIRe:MGAP....2227
BUS<m>:SWIRe:STRBe:HYSTeresis....2227
BUS<m>:SWIRe:STRBe:SOURce....2228
BUS<m>:SWIRe:STRBe:THReshold....2228
BUS<m>:SWIRe:SYSLect....2228
BUS<m>:SWIRe:COUPling....2228
BUS<m>:SWIRe:PRESet....2229

**BUS<m>:SWIRe:BPOSition** <SyncBitPos>

Sets the bit position, the align position for the manual synchronisation mode. This can be useful when parity errors exist in the signal, and parity check is the main indicator for the decoder to do packet alignment.

**Suffix:**
<m> 1..4

**Parameters:**
<SyncBitPos> Range: 0 to 10
Increment: 1
*RST: 0

**Manual operation:** See "Bit Position" on page 873

**BUS<m>:SWIRe:DATA:HYSTeresis** <HysteresisData>

Sets a value for the hysteresis of the data signal.

**Suffix:**
<m> 1..4

**Parameters:**
<HysteresisData> Range: 0 to 0.8
Increment: 0.01
*RST: 0.2
Default unit: V

**Manual operation:** See "Hysteresis" on page 872

**BUS<m>:SWIRe:DATA:SOURce** <SourceData>

Selects the source of the data signal.

**Suffix:**
<m> 1..4

**Parameters:**

<SourceData> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

*RST: C1W1

**Manual operation:** See "Data" on page 872

---

**BUS<m>:SWIRe:DATA:THReshold** <ThresholdData>

Sets the threshold value for the digitization of the data signal.

**Suffix:**

<m> 1..4

**Parameters:**

<ThresholdData>
Range: -10 to 10
Increment: 0.1
*RST: 1.25
Default unit: V

**Manual operation:** See "Threshold setup" on page 872

---

**BUS<m>:SWIRe:MGAP** <MinGapTime>

Sets the minimum duration of a gap. Any inactivity greater than this time will be interpreted as a gap and lead to a resynchronization to the signal.

**Suffix:**

<m> 1..4

**Parameters:**

<MinGapTime>
Range: 1E-9 to 10E-6
Increment: 100E-9
*RST: 200E-9
Default unit: s

**Firmware/software:** FW 3.30. Replaces `BUS<m>:SWIRe:MINGap`.

**Manual operation:** See "Min Gap" on page 873

---

**BUS<m>:SWIRe:STRBe:HYSTeresis** <HystStrobe>

Sets a value for the hysteresis of the strobe signal.

**Suffix:**

<m> 1..4

**Parameters:**

<HystStrobe>
Range: 0 to 0.8
Increment: 0.01
*RST: 0.2
Default unit: V

**Manual operation:** See "Hysteresis" on page 872

---

**BUS<m>:SWIRe:STRBe:SOURce** <SourceStrobe>

Selects the source of the strobe signal.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceStrobe> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

*RST: C2W1

**Manual operation:** See "Strobe" on page 872

---

**BUS<m>:SWIRe:STRBe:THReshold** <ThresholdStrobe>

Sets the threshold value for the digitization of the strobe signal.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdStrobe> Range: -10 to 10
Increment: 0.1
*RST: 1.25
Default unit: V

**Manual operation:** See "Threshold setup" on page 872

---

**BUS<m>:SWIRe:SYSLect** <SyncSelector>

Selects the mode for the synchronisation settings, i.e i.e. packet align.

**Suffix:**
<m> 1..4

**Parameters:**
<SyncSelector> AUTO | MANual

*RST: AUTO

**Manual operation:** See "Sync Settings" on page 873

---

**BUS<m>:SWIRe:COUPling** <ThresCpl>

Enables coupling, i.e. the same threshold and hysteresis value is used for the strob and the data signal.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresCpl> ON | OFF
*RST: ON

**Manual operation:** See "Threshold setup" on page 872

---

**BUS<m>:SWIRe:PRESet** <ThresholdPreset>

Prests the threshold and hysteresis value of the strobe and data signal.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdPreset> V25 | V165 | V125 | V09 | V12 | V0 | MANual
V25: 2.5 V (CMOS 5.0 V)
V165: 1.65 V (CMOS 3.5 V)
V125: 1.25 V (CMOS 1.5 V)
V09: 2.5 V (CMOS 1.8V)
V12: 1.2 V (LVDS single ended)
V0: 0 V (LVDS differential)
MANual: user-defined value
*RST: V12

**Manual operation:** See "Threshold setup" on page 872

### 24.17.22.2 Trigger

TRIGger<m>:SWIRe:CTYPe....2229
TRIGger<m>:SWIRe:DATA:CONDition....2230
TRIGger<m>:SWIRe:DATA:MAX....2230
TRIGger<m>:SWIRe:DATA:MIN....2230
TRIGger<m>:SWIRe:ERRor:ESC....2231
TRIGger<m>:SWIRe:ERRor:PARity....2231
TRIGger<m>:SWIRe:TIME:CONDition....2231
TRIGger<m>:SWIRe:TIME:MAX....2232
TRIGger<m>:SWIRe:TIME:MIN....2232
TRIGger<m>:SWIRe:TYPE....2232

---

**TRIGger<m>:SWIRe:CTYPe** <ControlType>

Triggers on a specific control type character.

**Suffix:**
<m> 1..3

**Parameters:**
<ControlType> ANY | FCT | EOP | EEP
**FCT**
Flow Control Token

**EOP**
Normal End of Packet

**EEP**
Error End of Packet

*RST: ANY

**Manual operation:** See "Trigger Control Type" on page 875

---

**TRIGger<m>:SWIRe:DATA:CONDition** <DataOperator>

Set the condition for the data value. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:SWIRe:DATA:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:SWIRe:DATA:MIN` and `TRIGger<m>:SWIRe:DATA:MAX`.

*RST: EQUal

**Manual operation:** See "Data (Data Pattern)" on page 876

---

**TRIGger<m>:SWIRe:DATA:MAX** <DataPatternMax>

Sets the the end value of a data for the data pattern range if `TRIGger<m>:SWIRe:DATA:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<DataPatternMax>

**Manual operation:** See "Data (Data Pattern)" on page 876

---

**TRIGger<m>:SWIRe:DATA:MIN** <DataPatternMin>

Specifies a data for the data pattern, or sets the the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<DataPatternMin>

**Manual operation:** See "Data (Data Pattern)" on page 876

---

**TRIGger<m>:SWIRe:ERRor:ESC** <ErrorESC>

Enables searching for an escape error.

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorESC> ON | OFF
*RST: ON

**Manual operation:** See "ESC Error" on page 876

---

**TRIGger<m>:SWIRe:ERRor:PARity** <ErrorParity>

Enables triggering on a parity error (even parity).

**Suffix:**
<m> 1..3

**Parameters:**
<ErrorParity> ON | OFF
*RST: ON

**Manual operation:** See "Parity Error" on page 876

---

**TRIGger<m>:SWIRe:TIME:CONDition** <TimeOperator>

Set the condition for the data value for the time code. You can define an exact value or a value range.

**Suffix:**
<m> 1..3

**Parameters:**
<TimeOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `TRIGger<m>:SWIRe:TIME:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `TRIGger<m>:SWIRe:TIME:MIN` and `TRIGger<m>:SWIRe:TIME:MAX`.

*RST: EQUal

**Manual operation:** See "Data (Time Code)" on page 876

---

**TRIGger<m>:SWIRe:TIME:MAX** <TimePatternMax>

Sets the the end value of a data value for the time code range if `TRIGger<m>:SWIRe:TIME:CONDition` is set to `INRange` or `OORange`.

**Suffix:**
<m> 1..3

**Parameters:**
<TimePatternMax>

**Manual operation:** See "Data (Time Code)" on page 876

---

**TRIGger<m>:SWIRe:TIME:MIN** <TimePatternMin>

Specifies a pattern for the data value for the time code, or sets the the start value of a pattern range.

**Suffix:**
<m> 1..3

**Parameters:**
<TimePatternMin>

**Manual operation:** See "Data (Time Code)" on page 876

---

**TRIGger<m>:SWIRe:TYPE** <Type>

Sets the trigger type for the SpaceWire analysis.

**Suffix:**
<m> 1..3

**Parameters:**
<Type> CTRL | DATA | NULL | TIME | ERRor
*RST: CTRL

**Manual operation:** See "Type" on page 875

### 24.17.22.3 Decode results

BUS<m>:SWIRe:RESults:FCOunt ..... 2233
BUS<m>:SWIRe:RESults:FRAMe<n>:DATA? ..... 2233
BUS<m>:SWIRe:RESults:FRAMe<n>:STARt? ..... 2233
BUS<m>:SWIRe:RESults:FRAMe<n>:STATe? ..... 2234
BUS<m>:SWIRe:RESults:FRAMe<n>:STOP? ..... 2234
BUS<m>:SWIRe:RESults:FRAMe<n>:TYPE? ..... 2234
BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:FVAL? ..... 2235
BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:STARt? ..... 2235

BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:STATus? ..... 2236
BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:STOP? ..... 2236
BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:TYPE? ..... 2237
BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:VAL? ..... 2237

**BUS<m>:SWIRe:RESults:FCOunt** <Key>

Returns the number of decoded frames for the selected serial bus.

**Suffix:**
<m> 1..4

**Setting parameters:**
<Key>

**Return values:**
<Count>

**BUS<m>:SWIRe:RESults:FRAMe<n>:DATA?**

Returns the data value.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<Data> 8-bit data value

**Example:**
```
BUS:SWIRe:RESults:FRAMe2:DATA?
<-- #H12
```

**Usage:** Query only

**BUS<m>:SWIRe:RESults:FRAMe<n>:STARt?**

Returns the start time of the specified frame.

**Suffix:**
<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:SWIRe:RESults:FRAMe<n>:STATe?**

Returns the overall state of the frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the serial bus.

**Return values:**

<State> OK | PAR | ESC | AMB | INComplete

**OK**
The frame is valid.

**PARity**
Parity error

**ESC**
Escape error

**AMB**
Ambiguous

**INComplete**
The frame is incomplete

*RST: OK

**Usage:** Query only

**BUS<m>:SWIRe:RESults:FRAMe<n>:STOP?**

Returns the end time of the specified frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:SWIRe:RESults:FRAMe<n>:TYPE?**

Returns the type of frame.

**Suffix:**

<m> 1..4
Selects the serial bus.

<n> *
Selects the frame.

**Return values:**

<Type> DATA | FCT | EOP | EEP | ESC | NULL | TCOD

**DATA**
Data frame

**FCT**
Flow control token

**EOP**
End of packet

**EEP**
Error end of packet

**ESC**
Escape

**NULL**
Null symbol

**TCOD**
Time code

*RST: DATA

**Usage:** Query only

---

**BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:FVAL?**

Returns the formatted value of the field with the selected field number within the frame with the selected frame number.

**Suffix:**

<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**

<FrameByteValue>

**Usage:** Query only

---

**BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:STARt?**

Returns the start time of the field with the selected field number within the frame with the selected frame number.

**Suffix:**

<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**

<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:STATus?**

Returns the status of the field with the selected field number within the frame with the selected frame number.

**Suffix:**

<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**

<State> OK | PAR | ESC | AMB | INComplete
PAR: parity error
ESC: escape
AMB: ambiguous
*RST: OK

**Usage:** Query only

---

**BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:STOP?**

Returns the stop time of the field with the selected field number within the frame with the selected frame number.

**Suffix:**

<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**

<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:TYPE?**

Returns the type of the field with the selected field number within the frame with the selected frame number.

**Suffix:**

<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**

<FrameByteValue>

**Usage:** Query only

---

**BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:VAL?**

Returns the value of the field with the selected field number within the frame with the selected frame number.

**Suffix:**

<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**

<FrameByteValue> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

### 24.17.22.4 Search settings

SEARch:TRIGger:SWIRe:CTYPe ..... 2238
SEARch:TRIGger:SWIRe:DATA:CONDition ..... 2238
SEARch:TRIGger:SWIRe:DATA:MAX ..... 2239
SEARch:TRIGger:SWIRe:DATA:MIN ..... 2239

SEARch:TRIGger:SWIRe:ERRor:ESC ..... 2239
SEARch:TRIGger:SWIRe:ERRor:PARity ..... 2239
SEARch:TRIGger:SWIRe:TIME:CONDition ..... 2240
SEARch:TRIGger:SWIRe:TIME:MAX ..... 2240
SEARch:TRIGger:SWIRe:TIME:MIN ..... 2241
SEARch:TRIGger:SWIRe:TYPE ..... 2241

---

**SEARch:TRIGger:SWIRe:CTYPe** <SearchName>,<ControlType>
**SEARch:TRIGger:SWIRe:CTYPe?** <SearchName>

Searches for a specific control type character.

**Parameters:**
<ControlType> ANY | FCT | EOP | EEP

**FCT**
Flow Control Token

**EOP**
Normal End of Packet

**EEP**
Error End of Packet

*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Trigger Control Type" on page 879

---

**SEARch:TRIGger:SWIRe:DATA:CONDition** <SearchName>,<DataOperator>
**SEARch:TRIGger:SWIRe:DATA:CONDition?** <SearchName>

Set the condition for the data pattern data value. You can define an exact value or a value range.

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:SWIRe:DATA:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:SWIRe:DATA:MIN` and `SEARch:TRIGger:SWIRe:DATA:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data (Data Pattern)" on page 880

---

**SEARch:TRIGger:SWIRe:DATA:MAX** <SearchName>,<DataPatternMax>
**SEARch:TRIGger:SWIRe:DATA:MAX?** <SearchName>

Sets the the end value of a data type range if `SEARch:TRIGger:SWIRe:DATA:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<DataPatternMax> Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data (Data Pattern)" on page 880

---

**SEARch:TRIGger:SWIRe:DATA:MIN** <SearchName>,<DataPatternMin>
**SEARch:TRIGger:SWIRe:DATA:MIN?** <SearchName>

Specifies a data bit pattern, or sets the the start value of a pattern range.

**Parameters:**
<DataPatternMin>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data (Data Pattern)" on page 880

---

**SEARch:TRIGger:SWIRe:ERRor:ESC** <SearchName>,<ErrorESC>
**SEARch:TRIGger:SWIRe:ERRor:ESC?** <SearchName>

Enables triggering on an escape error.

**Parameters:**
<ErrorESC> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "ESC Error" on page 880

---

**SEARch:TRIGger:SWIRe:ERRor:PARity** <SearchName>,<ErrorParity>
**SEARch:TRIGger:SWIRe:ERRor:PARity?** <SearchName>

Enables searching for a parity error (even parity).

**Parameters:**
<ErrorParity> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Parity Error" on page 880

---

**SEARch:TRIGger:SWIRe:TIME:CONDition** <SearchName>,<TimeOperator>
**SEARch:TRIGger:SWIRe:TIME:CONDition?** <SearchName>

Set the condition for the data value of the time code. You can define an exact value or a value range.

**Parameters:**
<TimeOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less or equal than, Greater Than, Greater or equal than. These conditions require one data pattern to be set with the corresponding `SEARch:TRIGger:SWIRe:TIME:MIN` command.

**INRange | OORange**
In range/Out of range: set the minimum and maximum value of the range with `SEARch:TRIGger:SWIRe:TIME:MIN` and `SEARch:TRIGger:SWIRe:TIME:MAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data (Time Code)" on page 879

---

**SEARch:TRIGger:SWIRe:TIME:MAX** <SearchName>,<TimePatternMax>
**SEARch:TRIGger:SWIRe:TIME:MAX?** <SearchName>

Sets the the end value of a data type range for the time code if `SEARch:TRIGger:SWIRe:TIME:CONDition` is set to `INRange` or `OORange`.

**Parameters:**
<TimePatternMax>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data (Time Code)" on page 879

**SEARch:TRIGger:SWIRe:TIME:MIN** <SearchName>,<TimePatternMin>
**SEARch:TRIGger:SWIRe:TIME:MIN?** <SearchName>

Specifies a data bit pattern for the time code, or sets the the start value of a pattern range.

**Parameters:**
<TimePatternMin>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Data (Time Code)" on page 879

**SEARch:TRIGger:SWIRe:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:SWIRe:TYPE?** <SearchName>

Sets the search type for the SpaceWire analysis.

**Parameters:**
<Type> CTRL | DATA | NULL | TIME | ERRor
*RST: CTRL

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Type" on page 879

### 24.17.22.5 Search results

In all `SEARch:RESult:DPHY:FRAMe<m>` commands, the suffix <m> selects the frame number in the list of search results.

SEARch:RESult:SWIRe:FCOunt? 2241
SEARch:RESult:SWIRe:FRAMe<m>:DATA? 2242
SEARch:RESult:SWIRe:FRAMe<m>:STARt? 2242
SEARch:RESult:SWIRe:FRAMe<m>:STATe? 2242
SEARch:RESult:SWIRe:FRAMe<m>:STOP? 2243
SEARch:RESult:SWIRe:FRAMe<m>:TYPE? 2243

**SEARch:RESult:SWIRe:FCOunt?** <SearchName>

Returns the number of frames within the search result for the selected serial bus.

**Query parameters:**
<SearchName>

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:SWIRe:FRAMe<m>:DATA?** <SearchName>

Returns the data value for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

**SEARch:RESult:SWIRe:FRAMe<m>:STARt?** <SearchName>

Returns the start time of the specified frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start>
Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:SWIRe:FRAMe<m>:STATe?** <SearchName>

Returns the overall state of the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | PAR | ESC | AMB | INComplete

**OK**
The frame is valid

**PARity**
Parity error

**ESC**
Escape Error

**AMB**
Ambiguous

**INComplete**
The frame is incomplete

*RST: OK

**Usage:** Query only

---

**SEARch:RESult:SWIRe:FRAMe<m>:STOP?** <SearchName>

Returns the end time for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:SWIRe:FRAMe<m>:TYPE?** <SearchName>

Returns the type of frame for the selected frame within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Type> DATA | FCT | EOP | EEP | ESC | NULL | TCOD

**DATA**
Data frame

**FCT**
Flow control token

**EOP**
End of packet

**EEP**
Error end of packet

**ESC**
Escape

**NULL**
Null symbol

**TCOD**
Time code

*RST: DATA

**Usage:** Query only

## 24.17.23 PCIe (option R&S RTPK72/K73)


- Configuration....2244
- Clock data recovery....2247
- Digital signal processing (DSP)....2249
- Trigger....2252
- Decode results....2263
- Search settings....2271
- Search results....2287


### 24.17.23.1 Configuration

In all `BUS<m>:PCIE` commands, the suffix <m> selects the serial bus.


BUS<m>:PCIE:DSCRambling....2244
BUS<m>:PCIE:FIDLe....2244
BUS<m>:PCIE:GEN....2245
BUS<m>:PCIE:LNKW....2245
BUS<m>:PCIE:LZER:SOURce....2245
BUS<m>:PCIE:LONE:SOURce....2246
BUS<m>:PCIE:LTWO:SOURce....2246
BUS<m>:PCIE:LTHRee:SOURce....2246
BUS<m>:PCIE:LZER:THRHigh....2246
BUS<m>:PCIE:LONE:THRHigh....2246
BUS<m>:PCIE:LTWO:THRHigh....2246
BUS<m>:PCIE:LTHRee:THRHigh....2246
BUS<m>:PCIE:LZER:THRLow....2246
BUS<m>:PCIE:LONE:THRLow....2246
BUS<m>:PCIE:LTWO:THRLow....2246
BUS<m>:PCIE:LTHRee:THRLow....2246
BUS<m>:PCIE:SMLanes....2246


**BUS<m>:PCIE:DSCRambling** <Active>

Enables descrambling of the data.

**Suffix:**
<m> 1..4

**Parameters:**
<Active> ON | OFF
*RST: ON

**Manual operation:** See "Descrambling" on page 885

**BUS<m>:PCIE:FIDLe** <Active>

Enables the filtering of idle packets.

**Suffix:**
<m> 1..4

**Parameters:**
<Active> ON | OFF
*RST: ON

**Manual operation:** See "Filter Idle" on page 888

---

**BUS<m>:PCIE:GEN** <Generation>

Sets the generation of the PCIe technology.

**Suffix:**
<m> 1..4

**Parameters:**
<Generation> V1 | V2 | V3

**V1**
PCIe Gen1 (2.5 Gbit/s)

**V2**
PCIe Gen2 (5 Gbit/s)

*RST: V1

**Manual operation:** See "Generation" on page 885

---

**BUS<m>:PCIE:LNKW** <LinkWidth>

Sets the link width, the number of lanes that are used for the transmission of the data.

**Suffix:**
<m> 1..4

**Parameters:**
<LinkWidth> X1 | X2 | X4
*RST: X1

**Manual operation:** See "Link" on page 885

---

**BUS<m>:PCIE:LZER:SOURce** <SourceL0>

Sets the signal sources for Lane 0.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceL0> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4
*RST: C1W1

**Manual operation:** See "Lane 0/1/2/3" on page 886

**BUS<m>:PCIE:LONE:SOURce** <SourceL1>
**BUS<m>:PCIE:LTWO:SOURce** <SourceL2>
**BUS<m>:PCIE:LTHRee:SOURce** <SourceL3>

Sets the signal sources for the corresponding logical lane.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceL3> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | NONE
*RST: LONe: C2W1, LTWo: C3W1, LTHree: C4W1

**Manual operation:** See "Lane 0/1/2/3" on page 886

**BUS<m>:PCIE:LZER:THRHigh** <ThresholdL0High>
**BUS<m>:PCIE:LONE:THRHigh** <ThresholdL1High>
**BUS<m>:PCIE:LTWO:THRHigh** <ThresholdL2High>
**BUS<m>:PCIE:LTHRee:THRHigh** <ThresholdL3High>

Sets the high threshold value for the respective lane.

**Suffix:**
<m> 1..4

**Parameters:**
<ThresholdL3High> Range: -5 to 5
Increment: 1E-3
*RST: 0.02
Default unit: V

**Manual operation:** See "High threshold" on page 887

**BUS<m>:PCIE:LZER:THRLow** <ThresholdL0Low>
**BUS<m>:PCIE:LONE:THRLow** <ThresholdL1Low>
**BUS<m>:PCIE:LTWO:THRLow** <ThresholdL2Low>
**BUS<m>:PCIE:LTHRee:THRLow** <ThresholdL3Low>
**BUS<m>:PCIE:SMLanes** <Active>

Enables the display of the result in multiple lanes.

**Suffix:**
<m> 1..4

**Parameters:**
<Active> ON | OFF
*RST: ON

**Manual operation:** See "Show multiple lanes" on page 888

### 24.17.23.2 Clock data recovery

BUS<m>:CDR:BITRate ..... 2247
BUS<m>:CDR:PLL:BWIDth ..... 2247
BUS<m>:CDR:PLL:DAMPing ..... 2247
BUS<m>:CDR:PLL:ORDer ..... 2248
BUS<m>:CDR:PLL:RELBwidth ..... 2248
BUS<m>:CDR:RESults ..... 2248
BUS<m>:CDR:SYNC ..... 2248

**BUS<m>:CDR:BITRate** <Bitrate>

Sets the quiescent frequency of the PLL. It corresponds to the data rate of the data stream from which the clock is to be recovered.

**Suffix:**
<m> 1..4

**Parameters:**
<Bitrate>
Range: 100 to 5E+9
Increment: 10
*RST: 1E+9
Default unit: bps

**Manual operation:** See "Nominal bit rate" on page 889

**BUS<m>:CDR:PLL:BWIDth** <PLLBandwidth>

Sets the PLL bandwidth. It defines the part of the spectrum that the PLL can follow during synchronization. The PLL bandwidth is usually defined by the transmission standard.

**Suffix:**
<m> 1..4

**Parameters:**
<PLLBandwidth>
Range: 20E+3 to 10E+6
Increment: 10
*RST: 599.88E+3
Default unit: Hz

**Manual operation:** See "PLL settings" on page 889

**BUS<m>:CDR:PLL:DAMPing** <DampingFactor>

Sets the damping factor, which is only relevant for second order PLL.

**Suffix:**
<m> 1..4

**Parameters:**

<DampingFactor> Range: 0.5 to 1
Increment: 0.01
*RST: 0.7

**Manual operation:** See "PLL settings" on page 889

---

**BUS<m>:CDR:PLL:ORDer** <PLLOrder>

Sets the order of the PLL: first or second order. PLL of higher order can compensate for more complex jitter behavior.

**Suffix:**

<m> 1..4

**Parameters:**

<PLLOrder> FIRSt | SECond
*RST: SECond

**Manual operation:** See "PLL settings" on page 889

---

**BUS<m>:CDR:PLL:RELBwidth** <PLLRelBw>

Sets the relative bandwidth, that is the ratio of the nominal bit rate to the PLL bandwidth.

**Suffix:**

<m> 1..4

**Parameters:**

<PLLRelBw> Range: 10 to 5000
Increment: 1
*RST: 1667

**Manual operation:** See "PLL settings" on page 889

---

**BUS<m>:CDR:RESults** <Results>

The PLL requires some time to synchronize to the phase of the data stream. You can select when the CDR algorithm returns clock edges.

**Suffix:**

<m> 1..4

**Parameters:**

<Results> ALL | AISYnc
*RST: ALL

**Manual operation:** See "Selected results" on page 890

---

**BUS<m>:CDR:SYNC** <InitialSync>

Defines the phase reference for the first clock edge.

**Suffix:**
<m> 1..4

**Parameters:**
<InitialSync> SAMPle | DATaedge

**SAMPle**
The first clock edge matches the first sample of the waveform at the left border of the display.

**DATaedge**
The first clock edge matches the first edge of the data signal.

*RST: DATaedge

**Manual operation:** See "Initial phase sync." on page 890

### 24.17.23.3 Digital signal processing (DSP)

BUS<m>:CTLE:DCGain....2249
BUS<m>:CTLE:ENBLe....2249
BUS<m>:CTLE:P1FReq....2250
BUS<m>:CTLE:P2FReq....2250
BUS<m>:CTLE:Z1FReq....2250
BUS<m>:PCIE:DFE:AUTO....2251
BUS<m>:PCIE:DFE:COEFficient....2251
BUS<m>:PCIE:DFE:ENBLe....2251
BUS<m>:PCIE:DFE:TCONt....2251

**BUS<m>:CTLE:DCGain** <DCGain>

Sets the adjustable DC gain. This value can be chosen according to the channel loss characteristics. For low-loss channels, you can set a value close to 0 dB.

**Suffix:**
<m> 1..4

**Parameters:**
<DCGain> Range: -100 to 100
Increment: 0.01
*RST: 0
Default unit: dB

**Manual operation:** See "DC Gain ($A_{DC}$)" on page 846

**BUS<m>:CTLE:ENBLe** <EnableCTLE>

Enables the continuous time linear equalization (CTLE) equalization.

**Suffix:**
<m> 1..4

**Parameters:**
<EnableCTLE> ON | OFF
*RST: OFF

**Manual operation:** See "Enable" on page 846

---

**BUS<m>:CTLE:P1FReq** <PoleOneFreq>

Sets the pole 1 frequency for the transfer function of the CTLE algorithm.

**Suffix:**
<m> 1..4

**Parameters:**
<PoleOneFreq> Range: 1 to 40E+9
Increment: 1
*RST: 0
Default unit: Hz

**Manual operation:** See "Pole frequency (wp1)/(wp2)" on page 846

---

**BUS<m>:CTLE:P2FReq** <PoleTwoFreq>

Sets the pole 2 frequency for the transfer function of the CTLE algorithm.

**Suffix:**
<m> 1..4

**Parameters:**
<PoleTwoFreq> Range: 1 to 40E+9
Increment: 1
*RST: 0
Default unit: Hz

**Manual operation:** See "Pole frequency (wp1)/(wp2)" on page 846

---

**BUS<m>:CTLE:Z1FReq** <ZroOneFreq>

Sets the zero frequency for the transfer function of the CTLE algorithm.

**Suffix:**
<m> 1..4

**Parameters:**
<ZroOneFreq> Range: 1 to 40E+9
Increment: 1
*RST: 0
Default unit: Hz

**Manual operation:** See "Zero frequency (fz)" on page 846

**BUS<m>:PCIE:DFE:AUTO** <AutoChEstimate>

Enables the auto channel estimation.

**Suffix:**
<m> 1..4

**Parameters:**
<AutoChEstimate> ON | OFF
*RST: ON

**Manual operation:** See "Estimate coefficients" on page 892

**BUS<m>:PCIE:DFE:COEFficient** <DFEcoeff>

Sets the value of the coefficient for the DFE .

**Suffix:**
<m> 1..4

**Parameters:**
<DFEcoeff> Range: -1 to 1
Increment: 1E-3
*RST: 0
Default unit: V

**Manual operation:** See "Coefficient" on page 892

**BUS<m>:PCIE:DFE:ENBLe** <EnableDFE>

Enables the use of the decision feedback equalization for PCIe.

**Suffix:**
<m> 1..4

**Parameters:**
<EnableDFE> ON | OFF
*RST: ON

**Manual operation:** See "Enable" on page 892

**BUS<m>:PCIE:DFE:TCONt** <CIRLength>

Sets a number of taps for the decision feedback filter.

**Suffix:**
<m> 1..4

**Parameters:**
<CIRLength> Range: 1 to 20
Increment: 1
*RST: 5

**Manual operation:** See "Number of Taps" on page 892

### 24.17.23.4 Trigger

The trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on serial buses.

To trigger on a serial bus, make sure that:

- `TRIGger<m>:SOURce[:SELect]` is set to `SBUS`.
- The sources of the serial bus are channel signals: use `BUS<m>:...:SOURce` commands.
- Decoding is enabled: `BUS<m>[:STATe]` is set to `ON`.

TRIGger<m>:PCIE:TRGLevel ... 2253
TRIGger<m>:PCIE:DLLP:FCTL:CRET ... 2253
TRIGger<m>:PCIE:DLLP:FCTL:VCID ... 2253
TRIGger<m>:PCIE:DLLP:MULT:AUTH ... 2253
TRIGger<m>:PCIE:DLLP:MULT:CRET ... 2254
TRIGger<m>:PCIE:DLLP:MULT:DVPT ... 2254
TRIGger<m>:PCIE:DLLP:MULT:HABS ... 2254
TRIGger<m>:PCIE:DLLP:MULT:MIXT ... 2254
TRIGger<m>:PCIE:DLLP:MULT:PHAS ... 2254
TRIGger<m>:PCIE:DLLP:MULT:RESA ... 2255
TRIGger<m>:PCIE:DLLP:MULT:TLPT ... 2255
TRIGger<m>:PCIE:DLLP:MULT:TYPE ... 2255
TRIGger<m>:PCIE:DLLP:MULT:VHFC ... 2255
TRIGger<m>:PCIE:DLLP:MULT:VHGR ... 2256
TRIGger<m>:PCIE:DLLP:MULT:VLNR ... 2256
TRIGger<m>:PCIE:DLLP:POWM ... 2256
TRIGger<m>:PCIE:DLLP:SEQ ... 2256
TRIGger<m>:PCIE:DLLP:TYPE ... 2256
TRIGger<m>:PCIE:DLLP:VPAT ... 2257
TRIGger<m>:PCIE:ERRC:CRC ... 2257
TRIGger<m>:PCIE:ERRC:DISP ... 2257
TRIGger<m>:PCIE:ERRC:ECRC ... 2257
TRIGger<m>:PCIE:ERRC:INVP ... 2257
TRIGger<m>:PCIE:ERRC:LCRC ... 2258
TRIGger<m>:PCIE:ERRC:LENGth ... 2258
TRIGger<m>:PCIE:ERRC:MALFormed ... 2258
TRIGger<m>:PCIE:ERRC:MXPFix ... 2258
TRIGger<m>:PCIE:ERRC:PFORder ... 2258
TRIGger<m>:PCIE:ERRC:SYCBit ... 2258
TRIGger<m>:PCIE:ERRC:UKNP ... 2259
TRIGger<m>:PCIE:ERRC:VOORange ... 2259
TRIGger<m>:PCIE:OSET:TYPE ... 2259
TRIGger<m>:PCIE:TLP:ADRT ... 2259
TRIGger<m>:PCIE:TLP:CFGT ... 2260
TRIGger<m>:PCIE:TLP:CPID ... 2260
TRIGger<m>:PCIE:TLP:CPLS ... 2260
TRIGger<m>:PCIE:TLP:DEID ... 2260
TRIGger<m>:PCIE:TLP:MERW ... 2260
TRIGger<m>:PCIE:TLP:MSGC ... 2261

TRIGger<m>:PCIE:TLP:MSGR ..... 2261
TRIGger<m>:PCIE:TLP:ORDE ..... 2261
TRIGger<m>:PCIE:TLP:REID ..... 2261
TRIGger<m>:PCIE:TLP:SNOO ..... 2262
TRIGger<m>:PCIE:TLP:SNUM ..... 2262
TRIGger<m>:PCIE:TLP:TCHN ..... 2262
TRIGger<m>:PCIE:TLP:TYPE ..... 2262
TRIGger<m>:PCIE:TLP:AT ..... 2262
TRIGger<m>:PCIE:TYPE ..... 2263

**TRIGger<m>:PCIE:TRGLevel** <TriggerLevel>

Sets the trigger level for PCIe Gen 3.

**Parameters:**

<TriggerLevel>
Range: -10 to 10
Increment: 1E-3
*RST: 0
Default unit: V

**Manual operation:** See "Trigger level" on page 893

**TRIGger<m>:PCIE:DLLP:FCTL:CRET** <CreditType>

Sets credit type value to be triggered on for the data link layer trigger.

**Parameters:**

<CreditType>
ANY | P | NP | CPL
*RST: ANY

**Manual operation:** See "Credit Type" on page 899

**TRIGger<m>:PCIE:DLLP:FCTL:VCID** <VirtualChnID>

Sets the virtual channel ID to be triggered on for the data link layer trigger.

**Parameters:**

<VirtualChnID>
ANY | ZERO | ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN
*RST: ANY

**Manual operation:** See "VC ID" on page 899

**TRIGger<m>:PCIE:DLLP:MULT:AUTH** <Authorized>

Sets the authorized presence state to be triggered on for the data link layer trigger.

**Parameters:**

<Authorized>
ANY | YES | NO
*RST: ANY

**Manual operation:** See "Authorized" on page 897

**TRIGger<m>:PCIE:DLLP:MULT:CRET** <CreditType>

Sets the credit type to be triggered on for the multi-root data link layer trigger.

**Parameters:**
<CreditType> ANY | DATA | HEADER
*RST: ANY

**Manual operation:** See "Credit Type" on page 898

**TRIGger<m>:PCIE:DLLP:MULT:DVPT** <DevicePortType>

Sets the device/port type to be triggered on for the data link layer trigger.

**Parameters:**
<DevicePortType> ANY | ZERO | ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN | EIGHT | NINE | TEN | ELEVEN | TWELVE | THIRTEEN | FOURTEEN | FIFTEEN
*RST: ANY

**Manual operation:** See "Device/Port Type" on page 897

**TRIGger<m>:PCIE:DLLP:MULT:HABS** <VHAbsent>

Sets the state of the absent virtual hiearchies (VH) to be triggered on for the data link layer trigger.

**Parameters:**
<VHAbsent> ANY | YES | NO
*RST: ANY

**Manual operation:** See "VH Absent" on page 898

**TRIGger<m>:PCIE:DLLP:MULT:MIXT** <MixedType>

Sets the mixed type presence state to be triggered on for the data link layer trigger.

**Parameters:**
<MixedType> ANY | YES | NO
*RST: ANY

**Manual operation:** See "Mixed Type" on page 897

**TRIGger<m>:PCIE:DLLP:MULT:PHAS** <Phase>

Sets the phase state to be triggered on for the data link layer trigger.

**Parameters:**
<Phase> ANY | YES | NO
*RST: ANY

**Manual operation:** See "Phase" on page 897

**TRIGger<m>:PCIE:DLLP:MULT:RESA** <ResetA>

Sets the on the value of the A bit to be triggered on for the data link layer trigger.

**Parameters:**

<ResetA> ANY | ACK | REQUEST

*RST: ANY

**Manual operation:** See "A" on page 899

**TRIGger<m>:PCIE:DLLP:MULT:TLPT** <TLPType>

Sets the multi-root type to be triggered on for the data link layer trigger.

**Parameters:**

<TLPType> ANY | P | NP | CPL

**P**
Posted credit.

**NP**
Non posted credit.

**CL**
Completion credit.

*RST: ANY

**Manual operation:** See "TLP Type" on page 898

**TRIGger<m>:PCIE:DLLP:MULT:TYPE** <Type>

Sets the multi-root type for the data link layer to be triggered on.

**Parameters:**

<Type> MRINIT | MRRESET | MRUPDATEFC | MRINITFC1 | MRINITFC2

*RST: MRINIT

**Manual operation:** See "MR DLLP" on page 896

**TRIGger<m>:PCIE:DLLP:MULT:VHFC** <InitVHFC>

Sets the virtual hierarchies flow control presence state to be triggered on for the data link layer trigger.

**Parameters:**

<InitVHFC> ANY | YES | NO

*RST: ANY

**Manual operation:** See "VH FC" on page 897

**TRIGger<m>:PCIE:DLLP:MULT:VHGR** <ResetVHGroup>

Sets the virtual hierarchies group to be triggered on for the data link layer trigger.

**Parameters:**
<ResetVHGroup> ANY | ZERO | ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN | EIGHT | NINE | TEN | ELEVEN | TWELVE | THIRTEEN | FOURTEEN | FIFTEEN

*RST: ANY

**Manual operation:** See "VH Group" on page 899

**TRIGger<m>:PCIE:DLLP:MULT:VLNR** <VirtualLink>

Sets the virtual link (VL) number to be triggered on for the data link layer trigger.

**Parameters:**
<VirtualLink> ANY | ZERO | ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN

*RST: ANY

**Manual operation:** See "VL Number" on page 898

**TRIGger<m>:PCIE:DLLP:POWM** <PowerMngtType>

Sets the power management type to be triggered on for the data link layer trigger.

**Parameters:**
<PowerMngtType> ANY | ENTERL1 | ENTERL23 | ACTREQL1 | REQUESTACK

*RST: ANY

**Manual operation:** See "PM Type" on page 900

**TRIGger<m>:PCIE:DLLP:SEQ** <ACKNAKSequence>

Sets the sequence field, indicating what TLPs are affected by the ACK/NAK, to be triggered on for the data link layer trigger.

**Parameters:**
<ACKNAKSequence>

**Manual operation:** See "Sequence" on page 899

**TRIGger<m>:PCIE:DLLP:TYPE** <Type>

Sets the DLLP t type to be triggered on for the data link layer trigger.

**Parameters:**
<Type> ANY | MRDLLP | ACK | NAK | INITFC1 | INITFC2 | UPDATEFC | PM | VENDOR

*RST: ANY

**Manual operation:** See "DLLP Type" on page 896

---

**TRIGger<m>:PCIE:DLLP:VPAT** <VendorBtPatt>

Sets the vendor pattern to be triggered on for the data link layer trigger.

**Parameters:**
<VendorBtPatt> 24-bit pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Vendor Pattern" on page 900

---

**TRIGger<m>:PCIE:ERRC:CRC** <CRC16>

Enables triggering on 16-bit cyclic redundancy check (CRC) errors.

**Parameters:**
<CRC16> ON | OFF
*RST: OFF

**Manual operation:** See "Error Condition PCIe Gen 1/2" on page 900

---

**TRIGger<m>:PCIE:ERRC:DISP** <Disparity>

Enables triggering on disparity errors.

**Parameters:**
<Disparity> ON | OFF
*RST: OFF

**Manual operation:** See "Error Condition PCIe Gen 1/2" on page 900

---

**TRIGger<m>:PCIE:ERRC:ECRC** <ECRC>

Enables triggering on end-to-end cyclic redundancy checksum (ECRC) errors.

**Parameters:**
<ECRC> ON | OFF
*RST: OFF

**Manual operation:** See "Error Condition PCIe Gen 1/2" on page 900

---

**TRIGger<m>:PCIE:ERRC:INVP** <InvalidSymbol>

Enables triggering on invalid symbol errors.

**Parameters:**
<InvalidSymbol> ON | OFF
*RST: OFF

**Manual operation:** See "Error Condition PCIe Gen 1/2" on page 900

**TRIGger<m>:PCIE:ERRC:LCRC** <LCRC>

Enables triggering on link cyclic redundancy check (LCRC) errors.

**Parameters:**
<LCRC> ON | OFF
*RST: OFF

**Manual operation:** See "Error Condition PCIe Gen 1/2" on page 900

**TRIGger<m>:PCIE:ERRC:LENGth** <LENG>

**Parameters:**
<LENG> ON | OFF
*RST: OFF

**Manual operation:** See "Error Condition PCIe Gen 3" on page 901

**TRIGger<m>:PCIE:ERRC:MALFormed** <MALF>

**Parameters:**
<MALF> ON | OFF
*RST: OFF

**Manual operation:** See "Error Condition PCIe Gen 3" on page 901

**TRIGger<m>:PCIE:ERRC:MXPFix** <MXPF>

**Parameters:**
<MXPF> ON | OFF
*RST: OFF

**Manual operation:** See "Error Condition PCIe Gen 3" on page 901

**TRIGger<m>:PCIE:ERRC:PFORder** <PFOR>

**Parameters:**
<PFOR> ON | OFF
*RST: OFF

**Manual operation:** See "Error Condition PCIe Gen 3" on page 901

**TRIGger<m>:PCIE:ERRC:SYCBit** <SYCB>

**Parameters:**
<SYCB> ON | OFF
*RST: OFF

**Manual operation:** See "Error Condition PCIe Gen 3" on page 901

**TRIGger<m>:PCIE:ERRC:UKNP** <UnknownPacket>

**Parameters:**
<UnknownPacket> ON | OFF
*RST: OFF

**Manual operation:** See "Error Condition PCIe Gen 3" on page 901

**TRIGger<m>:PCIE:ERRC:VOORange** <VOOR>

**Parameters:**
<VOOR> ON | OFF
*RST: OFF

**Manual operation:** See "Error Condition PCIe Gen 3" on page 901

**TRIGger<m>:PCIE:OSET:TYPE** <Type>

Sets the ordered set to be triggered on.

**Parameters:**
<Type> | SDS

**SKP**
SKP ordered sets

**TS1**
Training sequence 1

**TS2**
Training sequence 2

**FTS**
Fast training sequence

**EIOS**
Electrical idle ordered set

**EIEOS**
Electrical idle exit ordered set

**COMP**
Compliance pattern

**SDS**
Start of data sequence

*RST: SKP

**Manual operation:** See "Ordered set condition" on page 900

**TRIGger<m>:PCIE:TLP:ADRT** <AddressType>

Sets the address type to be triggered on for the transaction layer type trigger.

**Parameters:**
<AddressType> ANY | X32 | X64
*RST: ANY

**Manual operation:** See "Address Type" on page 895

---

**TRIGger<m>:PCIE:TLP:CFGT** <ConfigType>

Sets the configuration type to be triggered on for the transaction layer type trigger.

**Parameters:**
<ConfigType> ANY | TYPE0 | TYPE1
*RST: ANY

**Manual operation:** See "Type" on page 895

---

**TRIGger<m>:PCIE:TLP:CPID** <TLPCompleterID>

Sets the completer ID to be triggered on for the transaction layer type trigger.

**Parameters:**
<TLPCompleterID> 16-bit pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Completer ID" on page 896

---

**TRIGger<m>:PCIE:TLP:CPLS** <CplStatus>

Sets the completion status to be triggered on for the transaction layer type trigger.

**Parameters:**
<CplStatus> ANY | SUCCESS | UNSUPPORTED | RETRY | ABORT
*RST: ANY

**Manual operation:** See "Status" on page 895

---

**TRIGger<m>:PCIE:TLP:DEID** <TLPDeviceID>

Sets the device ID to be triggered on for the transaction layer type trigger.

**Parameters:**
<TLPDeviceID> 16-bit pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Device ID" on page 895

---

**TRIGger<m>:PCIE:TLP:MERW** <MemRW>

Sets the read/ write type to be triggered on for the transaction layer type trigger.

**Parameters:**
<MemRW> ANY | READ | WRITE
*RST: ANY

**Manual operation:** See "Read/Write" on page 895

---

**TRIGger<m>:PCIE:TLP:MSGC** <TLPMessageCode>

Sets the message code to be triggered on for the transaction layer type trigger.

**Parameters:**
<TLPMessageCode> 8-bit pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Message Code" on page 895

---

**TRIGger<m>:PCIE:TLP:MSGR** <MsgRouting>

Sets the message routing type to be triggered on for the transaction layer type trigger.

**Parameters:**
<MsgRouting> ANY | ROOTCOMPLEX | ADDRESS | DEVICEID | DEViceid | BROADCAST | LOCAL | GATHERED
DEVICEID = DEViceid
*RST: ANY

**Manual operation:** See "Routing" on page 895

---

**TRIGger<m>:PCIE:TLP:ORDE** <Ordering>

Sets the ordering type to be triggered on for the transaction layer type trigger.

**Parameters:**
<Ordering> ANY | STRONG | RELAX
*RST: ANY

**Manual operation:** See "Ordering" on page 894

---

**TRIGger<m>:PCIE:TLP:REID** <TLPRequesterID>

Sets the requester ID to be triggered on for the transaction layer type trigger.

**Parameters:**
<TLPRequesterID> 16-bit pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Requester ID" on page 895

**TRIGger<m>:PCIE:TLP:SNOO** <Snoop>

Sets the snoop state to be triggered on for the transaction layer type trigger.

**Parameters:**

<Snoop> ANY | YES | NO

*RST: ANY

**Manual operation:** See "Snoop" on page 894

**TRIGger<m>:PCIE:TLP:SNUM** <TLPSeqNo>

Sets the sequence numbe to be triggered on for the transaction layer type trigger.

**Parameters:**

<TLPSeqNo> 12-bit pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Manual operation:** See "Sequence Number" on page 894

**TRIGger<m>:PCIE:TLP:TCHN** <TrafficClass>

Sets the traffic class to be triggered on for the transaction layer type trigger.

**Parameters:**

<TrafficClass> ANY | ZERO | ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN

*RST: ANY

**Manual operation:** See "TC" on page 894

**TRIGger<m>:PCIE:TLP:TYPE** <Type>

Sets the transaction type to be triggered on for the transaction layer type trigger.

**Parameters:**

<Type> ANY | MEM | IO | CFG | MSG | CPL | FETCH | SWAP | CAS

*RST: ANY

**Manual operation:** See "TLP" on page 893

**TRIGger<m>:PCIE:TLP:AT** <TrafficClass>

Sets the AT Type.

**Parameters:**

<TrafficClass> ANY | UT | TRQ | TR | UR

`UT`: untranslated.
`TRQ`: translated request.
`TR`: translated.

UR: unrecognised request.

*RST: ANY

**TRIGger<m>:PCIE:TYPE** <Type>

Sets the trigger type for the PCI express analysis..

**Parameters:**

<Type> TLP | DLLP | ORD | ERR

*RST: TLP

**Manual operation:** See "Trigger type PCI express" on page 893

### 24.17.23.5 Decode results

In all BUS<m>:PCIE:RESult:PACKet<n> commands, the suffix <m> selects the serial bus and the suffix <n> selects the frame number in the decode table.

BUS<m>:PCIE:RESult:PACKet<n>:ADDR? 2263
BUS<m>:PCIE:RESult:PACKet<n>:CPID? 2264
BUS<m>:PCIE:RESult:PACKet<n>:CPS? 2264
BUS<m>:PCIE:RESult:PACKet<n>:DATA? 2264
BUS<m>:PCIE:RESult:PACKet<n>:DFC? 2265
BUS<m>:PCIE:RESult:PACKet<n>:FCOunt? 2265
BUS<m>:PCIE:RESult:PACKet<n>:FLD<o>:FVAL? 2265
BUS<m>:PCIE:RESult:PACKet<n>:FLD<o>:STATus? 2266
BUS<m>:PCIE:RESult:PACKet<n>:FLD<o>:TYPE? 2267
BUS<m>:PCIE:RESult:PACKet<n>:FLD<o>:VAL? 2267
BUS<m>:PCIE:RESult:PACKet<n>:HFC? 2267
BUS<m>:PCIE:RESult:PACKet<n>:LEN? 2268
BUS<m>:PCIE:RESult:PACKet<n>:RQID? 2268
BUS<m>:PCIE:RESult:PACKet<n>:SEQN? 2268
BUS<m>:PCIE:RESult:PACKet<n>:STARt? 2268
BUS<m>:PCIE:RESult:PACKet<n>:STATe? 2269
BUS<m>:PCIE:RESult:PACKet<n>:STOP? 2269
BUS<m>:PCIE:RESult:PACKet<n>:TAG? 2270
BUS<m>:PCIE:RESult:PACKet<n>:TYPE? 2270
BUS<m>:PCIE:RESult:PCOunt? 2271
BUS<m>:PCIE:RESult:PACKet<n>:BITRate? 2271

**BUS<m>:PCIE:RESult:PACKet<n>:ADDR?**

Returns the packet address for the specified packet.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<Address> Range: 0 to 0
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:PCIE:RESult:PACKet<n>:CPID?**

Returns the completer ID of the specified packet.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<CompleterID> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:PCIE:RESult:PACKet<n>:CPS?**

Returns the completion status field value of the specified packet.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<CplStatus> RESERVED | SC | UR | CRS | CA

**RESERVED**
Others

**SC**
Successful Completion

**UR**
Unsupported Request

**CRS**
Configuration Request Retry Status

**CA**
Completer Abort

*RST: RESERVED

**Usage:** Query only

---

**BUS<m>:PCIE:RESult:PACKet<n>:DATA?**

Returns the data value of the specified packet.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<Data>

**Usage:** Query only

---

**BUS<m>:PCIE:RESult:PACKet<n>:DFC?**

Returns the credit value of the payload data of the specified packet.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<DataFC> Range: 0 to 4095
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:PCIE:RESult:PACKet<n>:FCOunt?**

Returns the number of decoded fields in the selected packet.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<FieldCount> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:PCIE:RESult:PACKet<n>:FLD<o>:FVAL?**

Returns the formatted value of the field with the selected field number within the frame with the selected packet number (for the selected serial bus).

**Suffix:**
<m> 1..4
<n> *
<o> *
Selects the field number.

**Return values:**
<ValueFormatted> String parameter

**Usage:** Query only

---

**BUS<m>:PCIE:RESult:PACKet<n>:FLD<o>:STATus?**

Returns the status of the frame with the selected field number within the packet with the selected packet number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *

<o> *
Selects the field number.

**Return values:**
<State> OK | MALF | CRC16 | ECRC | LCRC | LEN | POE | MEE | UNCorrelated | FCRC | FPERror | UNKNown | OUTRange | INComplete | SYNCerror

**MALF**
Malformed packet

**CRC16**
CRC16 error

**ECRC**
ECRC error

**LCRC**
Link CRC error

**LEN**
Length error

**POE**
TLP prefix error

**MEE**
Max end-end TLP prefix error

**FCRC**
Fabric packet CRC

**FPERror**
Fabric packet error

**OUTRange**
Out of range

**INComplete**
Incomplete

**SYNCerror**
Synchronization error

*RST: OK

**Usage:** Query only

**BUS<m>:PCIE:RESult:PACKet<n>:FLD<o>:TYPE?**

Returns the type of the field with the selected field number within the frame with the selected packet number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *

<o> *
Selects the field number.

**Return values:**
<Type>

**Usage:** Query only

**BUS<m>:PCIE:RESult:PACKet<n>:FLD<o>:VAL?**

Returns the value of the field with the selected field number within the packet with the selected packet number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *

<o> *
Selects the field number.

**Return values:**
<FieldValue> Range: 0 to 0
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:PCIE:RESult:PACKet<n>:HFC?**

Returns the credit value of the header of the specified packet.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<HeaderFC> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:PCIE:RESult:PACKet<n>:LEN?**

Returns the length of the specified packet.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Length> Range: 0 to 1023
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:PCIE:RESult:PACKet<n>:RQID?**

Returns the requester ID of the specified packet.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<RequesterID> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:PCIE:RESult:PACKet<n>:SEQN?**

Returns the sequence number of the specified packet.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<SequenceNumber> Range: 0 to 4095
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:PCIE:RESult:PACKet<n>:STARt?**

Returns the start time of the specified packet.

**Suffix:**
<m> 1..4

<n> *

**Return values:**

<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:PCIE:RESult:PACKet<n>:STATe?**

Returns the state of the specified packet.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<State> OK | MALF | CRC16 | ECRC | LCRC | LEN | POE | MEE | UNCorrelated | FCRC | FPERror | UNKNown | OUTRange | INComplete | SYNCerror

**MALF**
Mal formated packet

**CRC16**
CRC16 error

**ECRC**
ECRC error

**LCRC**
LCRC error

**LEN**
Length error

**POE**
TLP prefix error

**MEE**
Max end-end TLP prefix error

*RST: OK

**Usage:** Query only

---

**BUS<m>:PCIE:RESult:PACKet<n>:STOP?**

Returns the stop time of the specified packet.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:PCIE:RESult:PACKet<n>:TAG?**

Returns the tag of the specified packet.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<Tag> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:PCIE:RESult:PACKet<n>:TYPE?**

Returns the type of the specified packet. For a description of the packet types, see Chapter 13.22.1, "The PCIe protocol", on page 882.

**Suffix:**

<m> 1..4

<n> *

**Return values:**

<Type> MRD32 | MRD64 | MRDLK32 | MRDLK64 | MWR32 | MWR64 | IORD | IOWR | CFGRD0 | CFGWR0 | CFGRD1 | CFGWR1 | MSG | MSGD | CPL | CPLD | CPLLK | CPLDLK | FA32 | FA64 | SWP32 | SWP64 | CAS32 | CAS64 | LPRFX | EPRFX | ACK | NAK | PMEL1 | PMEL23 | PMASRL1 | PMRA | VENDS | IFC1P | IFC1NP | IFC1CPL | IFC2P | IFC2NP | IFC2CPL | UPDFCP | UPDFCNP | UPDFCCPL | MRUPDFC | MRIFC1 | MRIFC2 | MRINIT | MRRESET | UNK | SKPOS | EIOS | FTSOS | TS1OS | TS2OS | EIEOS | COMPL | MCOMPL | DCOMPL | UDEF | IDLE | SDS | EDB | EDS | SDAT | SCTR | PMUX | UNKTlp | UNKDllp | SINV
*RST: UNK

**Usage:** Query only

**BUS<m>:PCIE:RESult:PCOunt?**

Returns the packet count for the selected serial bus, i.e. the number of packets in the present acquisition.

**Suffix:**
<m> 1..4

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

**BUS<m>:PCIE:RESult:PACKet<n>:BITRate?**

Returns the bit rate of the selected packet.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<PrimaryBitRate> Range: 0 to 100000000000
Increment: 1
*RST: 0
Default unit: bps

**Usage:** Query only

### 24.17.23.6 Search settings

SEARch:TRIGger:PCIE:DLLP:FCTL:CRET....2272
SEARch:TRIGger:PCIE:DLLP:FCTL:VCID....2273
SEARch:TRIGger:PCIE:DLLP:MULT:AUTH....2273
SEARch:TRIGger:PCIE:DLLP:MULT:CRET....2273
SEARch:TRIGger:PCIE:DLLP:MULT:DVPT....2273
SEARch:TRIGger:PCIE:DLLP:MULT:HABS....2274
SEARch:TRIGger:PCIE:DLLP:MULT:MIXT....2274
SEARch:TRIGger:PCIE:DLLP:MULT:PHAS....2274
SEARch:TRIGger:PCIE:DLLP:MULT:RESA....2274
SEARch:TRIGger:PCIE:DLLP:MULT:TLPT....2275
SEARch:TRIGger:PCIE:DLLP:MULT:TYPE....2275
SEARch:TRIGger:PCIE:DLLP:MULT:VHFC....2275
SEARch:TRIGger:PCIE:DLLP:MULT:VHGR....2276
SEARch:TRIGger:PCIE:DLLP:MULT:VLNR....2276
SEARch:TRIGger:PCIE:DLLP:POWM....2276
SEARch:TRIGger:PCIE:DLLP:SEQ....2277
SEARch:TRIGger:PCIE:DLLP:TYPE....2277
SEARch:TRIGger:PCIE:DLLP:VPAT....2277

SEARch:TRIGger:PCIE:ERRC:CRC ..... 2277
SEARch:TRIGger:PCIE:ERRC:DISP ..... 2278
SEARch:TRIGger:PCIE:ERRC:ECRC ..... 2278
SEARch:TRIGger:PCIE:ERRC:INVP ..... 2278
SEARch:TRIGger:PCIE:ERRC:LCRC ..... 2278
SEARch:TRIGger:PCIE:ERRC:LENGth ..... 2279
SEARch:TRIGger:PCIE:ERRC:MALFormed ..... 2279
SEARch:TRIGger:PCIE:ERRC:MXPFix ..... 2279
SEARch:TRIGger:PCIE:ERRC:PFORder ..... 2279
SEARch:TRIGger:PCIE:ERRC:SYCBit ..... 2280
SEARch:TRIGger:PCIE:ERRC:UKNP ..... 2280
SEARch:TRIGger:PCIE:ERRC:VOORange ..... 2280
SEARch:TRIGger:PCIE:OSET:SDS ..... 2280
SEARch:TRIGger:PCIE:OSET:COMP ..... 2281
SEARch:TRIGger:PCIE:OSET:EIDE ..... 2281
SEARch:TRIGger:PCIE:OSET:EIDL ..... 2281
SEARch:TRIGger:PCIE:OSET:FTS ..... 2281
SEARch:TRIGger:PCIE:OSET:SKIP ..... 2282
SEARch:TRIGger:PCIE:OSET:TSONe ..... 2282
SEARch:TRIGger:PCIE:OSET:TSTWo ..... 2282
SEARch:TRIGger:PCIE:TLP:ADRT ..... 2282
SEARch:TRIGger:PCIE:TLP:CFGT ..... 2283
SEARch:TRIGger:PCIE:TLP:CPID ..... 2283
SEARch:TRIGger:PCIE:TLP:CPLS ..... 2283
SEARch:TRIGger:PCIE:TLP:DEID ..... 2284
SEARch:TRIGger:PCIE:TLP:MERW ..... 2284
SEARch:TRIGger:PCIE:TLP:MSGC ..... 2284
SEARch:TRIGger:PCIE:TLP:MSGR ..... 2284
SEARch:TRIGger:PCIE:TLP:ORDE ..... 2285
SEARch:TRIGger:PCIE:TLP:REID ..... 2285
SEARch:TRIGger:PCIE:TLP:SNOO ..... 2285
SEARch:TRIGger:PCIE:TLP:SNUM ..... 2286
SEARch:TRIGger:PCIE:TLP:TCHN ..... 2286
SEARch:TRIGger:PCIE:TLP:TYPE ..... 2286
SEARch:TRIGger:PCIE:TYPE ..... 2286

---

**SEARch:TRIGger:PCIE:DLLP:FCTL:CRET** <SearchName>,<CreditType>
**SEARch:TRIGger:PCIE:DLLP:FCTL:CRET?** <SearchName>

Sets credit type value to be searched for, for the data link layer.

**Parameters:**
<CreditType> ANY | P | NP | CPL
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Credit Type" on page 910

**SEARch:TRIGger:PCIE:DLLP:FCTL:VCID** <SearchName>,<VirtualChnID>
**SEARch:TRIGger:PCIE:DLLP:FCTL:VCID?** <SearchName>

Sets the virtual channel ID to be searched for, for the data link layer.

**Parameters:**
<VirtualChnID> ANY | ZERO | ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "VC ID" on page 911

**SEARch:TRIGger:PCIE:DLLP:MULT:AUTH** <SearchName>,<Authorized>
**SEARch:TRIGger:PCIE:DLLP:MULT:AUTH?** <SearchName>

Sets the authorized presence state to be searched for, for the data link layer.

**Parameters:**
<Authorized> ANY | YES | NO
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Authorized" on page 909

**SEARch:TRIGger:PCIE:DLLP:MULT:CRET** <SearchName>,<CreditType>
**SEARch:TRIGger:PCIE:DLLP:MULT:CRET?** <SearchName>

Sets the credit type to be searched for, for the data link layer.

**Parameters:**
<CreditType> ANY | DATA | HEADER
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**SEARch:TRIGger:PCIE:DLLP:MULT:DVPT** <SearchName>,<DevicePortType>
**SEARch:TRIGger:PCIE:DLLP:MULT:DVPT?** <SearchName>

Sets the device/port type to be searched for, for the data link layer.

**Parameters:**
<DevicePortType> ANY | ZERO | ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN | EIGHT | NINE | TEN | ELEVEN | TWELVE | THIRTEEN | FOURTEEN | FIFTEEN
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Device/Port Type" on page 909

---

**SEARch:TRIGger:PCIE:DLLP:MULT:HABS** <SearchName>,<VHAbsent>
**SEARch:TRIGger:PCIE:DLLP:MULT:HABS?** <SearchName>

Sets the state of the absent virtual hiearchies (VH) to be searched for, for the data link layer.

**Parameters:**
<VHAbsent> ANY | YES | NO
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "VH Absent" on page 910

---

**SEARch:TRIGger:PCIE:DLLP:MULT:MIXT** <SearchName>,<MixedType>
**SEARch:TRIGger:PCIE:DLLP:MULT:MIXT?** <SearchName>

Sets the mixed type presence state to be searched for, for the data link layer.

**Parameters:**
<MixedType> ANY | YES | NO
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Mixed Type" on page 909

---

**SEARch:TRIGger:PCIE:DLLP:MULT:PHAS** <SearchName>,<Phase>
**SEARch:TRIGger:PCIE:DLLP:MULT:PHAS?** <SearchName>

Sets the phase state to be searched for, for the data link layer.

**Parameters:**
<Phase> ANY | YES | NO
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Phase" on page 909

---

**SEARch:TRIGger:PCIE:DLLP:MULT:RESA** <SearchName>,<ResetA>
**SEARch:TRIGger:PCIE:DLLP:MULT:RESA?** <SearchName>

Sets the on the value of the A bit to be searched for, for the data link layer.

**Parameters:**
<ResetA> ANY | ACK | REQUEST
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "A" on page 911

---

**SEARch:TRIGger:PCIE:DLLP:MULT:TLPT** <SearchName>,<Type>
**SEARch:TRIGger:PCIE:DLLP:MULT:TLPT?** <SearchName>

Sets the multi-root type to be searched for, for the data link layer.

**Parameters:**
<Type> ANY | P | NP | CPL

**P**
Posted credit.

**NP**
Non posted credit.

**CL**
Completion credit.

*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "TLP Type" on page 910

---

**SEARch:TRIGger:PCIE:DLLP:MULT:TYPE** <SearchName>,<MRType>
**SEARch:TRIGger:PCIE:DLLP:MULT:TYPE?** <SearchName>

Sets the multi-root type to be searched for, for the data link layer.

**Parameters:**
<MRType> MRINIT | MRRESET | MRUPDATEFC | MRINITFC1 | MRINITFC2

*RST: MRINIT

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "MR DLLP" on page 908

---

**SEARch:TRIGger:PCIE:DLLP:MULT:VHFC** <SearchName>,<InitVHFC>
**SEARch:TRIGger:PCIE:DLLP:MULT:VHFC?** <SearchName>

Sets the virtual hierarchies flow control presence state to be searched for, for the data link layer.

**Parameters:**
<InitVHFC> ANY | YES | NO
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "VH FC" on page 909

---

**SEARch:TRIGger:PCIE:DLLP:MULT:VHGR** <SearchName>,<ResetVHGroup>
**SEARch:TRIGger:PCIE:DLLP:MULT:VHGR?** <SearchName>

Sets the virtual hierarchies group to be searched for, for the data link layer.

**Parameters:**
<ResetVHGroup> ANY | ZERO | ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN | EIGHT | NINE | TEN | ELEVEN | TWELVE | THIRTEEN | FOURTEEN | FIFTEEN
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "VH Group" on page 911

---

**SEARch:TRIGger:PCIE:DLLP:MULT:VLNR** <SearchName>,<VirtualLink>
**SEARch:TRIGger:PCIE:DLLP:MULT:VLNR?** <SearchName>

Sets the virtual link (VL) number to be searched for, for the data link layer.

**Parameters:**
<VirtualLink> ANY | ZERO | ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "VL Number" on page 910

---

**SEARch:TRIGger:PCIE:DLLP:POWM** <SearchName>,<PowerMngtType>
**SEARch:TRIGger:PCIE:DLLP:POWM?** <SearchName>

Sets the power management type to be searched for, for the data link layer.

**Parameters:**
<PowerMngtType> ANY | ENTERL1 | ENTERL23 | ACTREQL1 | REQUESTACK
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "PM Type" on page 912

**SEARch:TRIGger:PCIE:DLLP:SEQ** <SearchName>,<ACKNAKSequence>
**SEARch:TRIGger:PCIE:DLLP:SEQ?** <SearchName>

Sets the sequence field, indicating what TLPs are affected by the ACK/NAK, to be searched for, for the data link layer.

**Parameters:**
<ACKNAKSequence>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Sequence" on page 911

**SEARch:TRIGger:PCIE:DLLP:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:PCIE:DLLP:TYPE?** <SearchName>

Sets the DLLP t type to be searched for, for the data link layer.

**Parameters:**
<Type> ANY | MRDLLP | ACK | NAK | INITFC1 | INITFC2 | UPDATEFC | PM | VENDOR
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "DLLP Type" on page 908

**SEARch:TRIGger:PCIE:DLLP:VPAT** <SearchName>,<VendorBtPatt>
**SEARch:TRIGger:PCIE:DLLP:VPAT?** <SearchName>

Sets the vendor pattern to be searched for, for the data link layer.

**Parameters:**
<VendorBtPatt> 24-bit pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Vendor Pattern" on page 912

**SEARch:TRIGger:PCIE:ERRC:CRC** <SearchName>,<CRC16>
**SEARch:TRIGger:PCIE:ERRC:CRC?** <SearchName>

Enables searching for 16-bit cyclic redundancy check (CRC) errors.

**Parameters:**
<CRC16> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 912

---

**SEARch:TRIGger:PCIE:ERRC:DISP** <SearchName>,<Disparity>
**SEARch:TRIGger:PCIE:ERRC:DISP?** <SearchName>

Enables searching for disparity errors.

**Parameters:**
<Disparity> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 912

---

**SEARch:TRIGger:PCIE:ERRC:ECRC** <SearchName>,<ECRC>
**SEARch:TRIGger:PCIE:ERRC:ECRC?** <SearchName>

Enables searching for end-to-end cyclic redundancy checksum (ECRC) errors.

**Parameters:**
<ECRC> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 912

---

**SEARch:TRIGger:PCIE:ERRC:INVP** <SearchName>,<InvalidSymbol>
**SEARch:TRIGger:PCIE:ERRC:INVP?** <SearchName>

Enables searching for invalid symbol errors.

**Parameters:**
<InvalidSymbol> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 912

---

**SEARch:TRIGger:PCIE:ERRC:LCRC** <SearchName>,<LCRC>
**SEARch:TRIGger:PCIE:ERRC:LCRC?** <SearchName>

Enables searching for link cyclic redundancy check (LCRC) errors.

**Parameters:**
<LCRC> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition" on page 912

---

**SEARch:TRIGger:PCIE:ERRC:LENGth** <SearchName>,<LENG>
**SEARch:TRIGger:PCIE:ERRC:LENGth?** <SearchName>

**Parameters:**
<LENG> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition PCIe Gen 3" on page 913

---

**SEARch:TRIGger:PCIE:ERRC:MALFormed** <SearchName>,<MALF>
**SEARch:TRIGger:PCIE:ERRC:MALFormed?** <SearchName>

**Parameters:**
<MALF> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition PCIe Gen 3" on page 913

---

**SEARch:TRIGger:PCIE:ERRC:MXPFix** <SearchName>,<MXPF>
**SEARch:TRIGger:PCIE:ERRC:MXPFix?** <SearchName>

**Parameters:**
<MXPF> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition PCIe Gen 3" on page 913

---

**SEARch:TRIGger:PCIE:ERRC:PFORder** <SearchName>,<PFOR>
**SEARch:TRIGger:PCIE:ERRC:PFORder?** <SearchName>

**Parameters:**
<PFOR> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition PCIe Gen 3" on page 913

---

**SEARch:TRIGger:PCIE:ERRC:SYCBit** <SearchName>,<SYCB>
**SEARch:TRIGger:PCIE:ERRC:SYCBit?** <SearchName>

**Parameters:**
<SYCB> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition PCIe Gen 3" on page 913

---

**SEARch:TRIGger:PCIE:ERRC:UKNP** <SearchName>,<UnknownPacket>
**SEARch:TRIGger:PCIE:ERRC:UKNP?** <SearchName>

**Parameters:**
<UnknownPacket> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition PCIe Gen 3" on page 913

---

**SEARch:TRIGger:PCIE:ERRC:VOORange** <SearchName>,<VOOR>
**SEARch:TRIGger:PCIE:ERRC:VOORange?** <SearchName>

**Parameters:**
<VOOR> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Error Condition PCIe Gen 3" on page 913

---

**SEARch:TRIGger:PCIE:OSET:SDS** <SearchName>,<Format>
**SEARch:TRIGger:PCIE:OSET:SDS?** <SearchName>

**Parameters:**
<Format> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Ordered set condition" on page 912

---

**SEARch:TRIGger:PCIE:OSET:COMP** <SearchName>,<Format>
**SEARch:TRIGger:PCIE:OSET:COMP?** <SearchName>

Enables searching forcompliance pattern ordered sets.

**Parameters:**
<Format> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Ordered set condition" on page 912

---

**SEARch:TRIGger:PCIE:OSET:EIDE** <SearchName>,<Format>
**SEARch:TRIGger:PCIE:OSET:EIDE?** <SearchName>

Enables searching for electrical idle exit ordered sets.

**Parameters:**
<Format> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Ordered set condition" on page 912

---

**SEARch:TRIGger:PCIE:OSET:EIDL** <SearchName>,<Format>
**SEARch:TRIGger:PCIE:OSET:EIDL?** <SearchName>

Enables searching for electrical idle ordered sets.

**Parameters:**
<Format> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Ordered set condition" on page 912

---

**SEARch:TRIGger:PCIE:OSET:FTS** <SearchName>,<Format>
**SEARch:TRIGger:PCIE:OSET:FTS?** <SearchName>

Enables searching for fast training sequence ordered sets.

**Parameters:**
<Format> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Ordered set condition" on page 912

---

**SEARch:TRIGger:PCIE:OSET:SKIP** <SearchName>,<Format>
**SEARch:TRIGger:PCIE:OSET:SKIP?** <SearchName>

Enables searching for SKP ordered sets.

**Parameters:**
<Format> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Ordered set condition" on page 912

---

**SEARch:TRIGger:PCIE:OSET:TSONe** <SearchName>,<Format>
**SEARch:TRIGger:PCIE:OSET:TSONe?** <SearchName>

Enables searching for training sequence 1 ordered sets.

**Parameters:**
<Format> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Ordered set condition" on page 912

---

**SEARch:TRIGger:PCIE:OSET:TSTWo** <SearchName>,<Format>
**SEARch:TRIGger:PCIE:OSET:TSTWo?** <SearchName>

Enables searching for training sequence 2 ordered sets.

**Parameters:**
<Format> ON | OFF
*RST: OFF

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Ordered set condition" on page 912

---

**SEARch:TRIGger:PCIE:TLP:ADRT** <SearchName>,<AddressType>
**SEARch:TRIGger:PCIE:TLP:ADRT?** <SearchName>

Sets the address type to be searched for in the transaction layer type.

**Parameters:**
<AddressType> ANY | X32 | X64
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Address Type" on page 907

---

**SEARch:TRIGger:PCIE:TLP:CFGT** <SearchName>,<CfgType>
**SEARch:TRIGger:PCIE:TLP:CFGT?** <SearchName>

Sets the configuration type to be searched for in the transaction layer type.

**Parameters:**
<CfgType> ANY | TYPE0 | TYPE1
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Type" on page 907

---

**SEARch:TRIGger:PCIE:TLP:CPID** <SearchName>,<TLPCompleterID>
**SEARch:TRIGger:PCIE:TLP:CPID?** <SearchName>

Sets the completer ID to be searched for in the transaction layer type.

**Parameters:**
<TLPCompleterID> 16-bit pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Completer ID" on page 908

---

**SEARch:TRIGger:PCIE:TLP:CPLS** <SearchName>,<CplStatus>
**SEARch:TRIGger:PCIE:TLP:CPLS?** <SearchName>

Sets the completion status to be searched for in the transaction layer type.

**Parameters:**
<CplStatus> ANY | SUCCESS | UNSUPPORTED | RETRY | ABORT
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Status" on page 908

**SEARch:TRIGger:PCIE:TLP:DEID** <SearchName>,<TLPDeviceID>
**SEARch:TRIGger:PCIE:TLP:DEID?** <SearchName>

Sets the device ID to be searched for in the transaction layer type.

**Parameters:**
<TLPDeviceID> 16-bit pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Device ID" on page 907

**SEARch:TRIGger:PCIE:TLP:MERW** <SearchName>,<MemRW>
**SEARch:TRIGger:PCIE:TLP:MERW?** <SearchName>

Sets the read/ write type to be searched for in the transaction layer type.

**Parameters:**
<MemRW> ANY | READ | WRITE
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Read/Write" on page 907

**SEARch:TRIGger:PCIE:TLP:MSGC** <SearchName>,<TLPMessageCode>
**SEARch:TRIGger:PCIE:TLP:MSGC?** <SearchName>

Sets the message code to be searched for in the transaction layer type.

**Parameters:**
<TLPMessageCode> 8-bit pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Message Code" on page 907

**SEARch:TRIGger:PCIE:TLP:MSGR** <SearchName>,<MsgRouting>
**SEARch:TRIGger:PCIE:TLP:MSGR?** <SearchName>

Sets the message routing type to be searched for in the transaction layer type.

**Parameters:**
<MsgRouting> ANY | ROOTCOMPLEX | ADDRESS | DEVICEID | DEViceid | BROADCAST | LOCAL | GATHERED
DEVICEID = DEViceid
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Routing" on page 907

---

**SEARch:TRIGger:PCIE:TLP:ORDE** <SearchName>,<Ordering>
**SEARch:TRIGger:PCIE:TLP:ORDE?** <SearchName>

Sets the ordering type to be searched for in the transaction layer type.

**Parameters:**
<Ordering> ANY | STRONG | RELAX
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Ordering" on page 906

---

**SEARch:TRIGger:PCIE:TLP:REID** <SearchName>,<TLPRequesterID>
**SEARch:TRIGger:PCIE:TLP:REID?** <SearchName>

Sets the requester ID to be searched for in the transaction layer type.

**Parameters:**
<TLPRequesterID> 16-bit pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Requester ID" on page 907

---

**SEARch:TRIGger:PCIE:TLP:SNOO** <SearchName>,<Snoop>
**SEARch:TRIGger:PCIE:TLP:SNOO?** <SearchName>

Sets the snoop state to be searched for in the transaction layer type.

**Parameters:**
<Snoop> ANY | YES | NO
*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Snoop" on page 907

**SEARch:TRIGger:PCIE:TLP:SNUM** <SearchName>,<TLPSeqNo>
**SEARch:TRIGger:PCIE:TLP:SNUM?** <SearchName>

Sets the sequence numbe to be searched for in the transaction layer type.

**Parameters:**
<TLPSeqNo> 12-bit pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The string parameter accepts the bit value X (don't care).

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Sequence Number" on page 906

**SEARch:TRIGger:PCIE:TLP:TCHN** <SearchName>,<TrafficClass>
**SEARch:TRIGger:PCIE:TLP:TCHN?** <SearchName>

Sets the traffic class to be searched for in the transaction layer type.

**Parameters:**
<TrafficClass> ANY | ZERO | ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN

*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "TC" on page 906

**SEARch:TRIGger:PCIE:TLP:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:PCIE:TLP:TYPE?** <SearchName>

Sets the transaction type to be searched for in the transaction layer type.

**Parameters:**
<Type> ANY | MEM | IO | CFG | MSG | CPL | FETCH | SWAP | CAS

*RST: ANY

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "TLP" on page 906

**SEARch:TRIGger:PCIE:TYPE** <SearchName>,<Type>
**SEARch:TRIGger:PCIE:TYPE?** <SearchName>

**Parameters:**
<Type> TLP | DLLP | ORD | ERR

*RST: TLP

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Searches for PCI express" on page 905

### 24.17.23.7 Search results

In all `SEARch:RESult:PCIE:PACKet<m>` commands, the suffix <m> selects the frame number in the list of search results.

In search result commands, you have to specify the `<SearchName>` parameter. It is a string parameter that contains the search definition name.

For a description of the returned values, see the corresponding commands in Chapter 24.17.23.5, "Decode results", on page 2263.

SEARch:RESult:PCIE:PACKet<m>:ADDR? 2287
SEARch:RESult:PCIE:PACKet<m>:CPID? 2287
SEARch:RESult:PCIE:PACKet<m>:CPS? 2288
SEARch:RESult:PCIE:PACKet<m>:DATA? 2288
SEARch:RESult:PCIE:PACKet<m>:DFC? 2288
SEARch:RESult:PCIE:PACKet<m>:FCOunt? 2289
SEARch:RESult:PCIE:PACKet<m>:FLD<n>:FVAL? 2289
SEARch:RESult:PCIE:PACKet<m>:FLD<n>:STATus? 2289
SEARch:RESult:PCIE:PACKet<m>:FLD<n>:TYPE? 2289
SEARch:RESult:PCIE:PACKet<m>:FLD<n>:VAL? 2290
SEARch:RESult:PCIE:PACKet<m>:HFC? 2290
SEARch:RESult:PCIE:PACKet<m>:LEN? 2290
SEARch:RESult:PCIE:PACKet<m>:RQID? 2291
SEARch:RESult:PCIE:PACKet<m>:SEQN? 2291
SEARch:RESult:PCIE:PACKet<m>:STARt? 2291
SEARch:RESult:PCIE:PACKet<m>:STATe? 2291
SEARch:RESult:PCIE:PACKet<m>:STOP? 2292
SEARch:RESult:PCIE:PACKet<m>:TAG? 2292
SEARch:RESult:PCIE:PACKet<m>:TYPE? 2292
SEARch:RESult:PCIE:PCOunt? 2293

---

**SEARch:RESult:PCIE:PACKet<m>:ADDR?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Address>

**Usage:** Query only

---

**SEARch:RESult:PCIE:PACKet<m>:CPID?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<CompleterID> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:PCIE:PACKet<m>:CPS?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<CompleteState> RESERVED | SC | UR | CRS | CA
*RST: RESERVED

**Usage:** Query only

---

**SEARch:RESult:PCIE:PACKet<m>:DATA?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Data>

**Usage:** Query only

---

**SEARch:RESult:PCIE:PACKet<m>:DFC?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<DataFC> Range: 0 to 4095
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:PCIE:PACKet<m>:FCOunt?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FieldCount> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:PCIE:PACKet<m>:FLD<n>:FVAL?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName> String parameter

**Return values:**
<ValueFormatted> String parameter

**Usage:** Query only

**SEARch:RESult:PCIE:PACKet<m>:FLD<n>:STATus?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | MALF | CRC16 | ECRC | LCRC | LEN | POE | MEE | UNCorrelated | FCRC | FPERror | UNKNown | OUTRange | INComplete | SYNCerror

*RST: OK

**Usage:** Query only

**SEARch:RESult:PCIE:PACKet<m>:FLD<n>:TYPE?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<Type>

**Usage:** Query only

---

**SEARch:RESult:PCIE:PACKet<m>:FLD<n>:VAL?** <SearchName>

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<FieldValue> Range: 0 to 0
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:PCIE:PACKet<m>:HFC?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<HeaderFC> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:PCIE:PACKet<m>:LEN?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Length> Range: 0 to 1023
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:PCIE:PACKet<m>:RQID?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<RequesterID> Range: 0 to 65535
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:PCIE:PACKet<m>:SEQN?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<SequenceNumber> Range: 0 to 4095
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:PCIE:PACKet<m>:STARt?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:PCIE:PACKet<m>:STATe?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | MALF | CRC16 | ECRC | LCRC | LEN | POE | MEE | UNCorrelated | FCRC | FPERror | UNKNown | OUTRange | INComplete | SYNCerror
*RST: OK

**Usage:** Query only

---

**SEARch:RESult:PCIE:PACKet<m>:STOP?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:PCIE:PACKet<m>:TAG?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Tag> Range: 0 to 255
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:PCIE:PACKet<m>:TYPE?** <SearchName>

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**

<Type> MRD32 | MRD64 | MRDLK32 | MRDLK64 | MWR32 | MWR64 | IORD | IOWR | CFGRD0 | CFGWR0 | CFGRD1 | CFGWR1 | MSG | MSGD | CPL | CPLD | CPLLK | CPLDLK | FA32 | FA64 | SWP32 | SWP64 | CAS32 | CAS64 | LPRFX | EPRFX | ACK | NAK | PMEL1 | PMEL23 | PMASRL1 | PMRA | VENDS | IFC1P | IFC1NP | IFC1CPL | IFC2P | IFC2NP | IFC2CPL | UPDFCP | UPDFCNP | UPDFCCPL | MRUPDFC | MRIFC1 | MRIFC2 | MRINIT | MRRESET | UNK | SKPOS | EIOS | FTSOS | TS1OS | TS2OS | EIEOS | COMPL | MCOMPL | DCOMPL | UDEF | IDLE | SDS | EDB | EDS | SDAT | SCTR | PMUX | UNKTlp | UNKDllp | SINV

*RST: UNK

**Usage:** Query only

---

**SEARch:RESult:PCIE:PCOunt?** <SearchName>

**Query parameters:**

<SearchName>

**Return values:**

<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

## 24.17.24 DDR (option R&S RTP-K91/K93)


- Configuration....2293
- Decode results....2296
- Search settings....2300
- Search results....2305


### 24.17.24.1 Configuration

In all `BUS<m>:DDR` commands, the suffix <m> selects the serial bus.


BUS<m>:DDR:STANdard....2294
BUS<m>:DDR:DATA:HYSTeresis....2294
BUS<m>:DDR:DATA:SOURce....2294
BUS<m>:DDR:DATA:THReshold....2294
BUS<m>:DDR:STRBe:HYSTeresis....2295
BUS<m>:DDR:STRBe:SOURce....2295
BUS<m>:DDR:STRBe:THReshold....2295
BUS<m>:DDR:BITRate....2296

**BUS<m>:DDR:STANdard** <Standard>

Selects the DDR standard.

**Suffix:**
<m> 1..4

**Parameters:**
<Standard> DDR3 | | DDR4
DDR3: requires option R&S RTP-K91.
DDR4: requires option R&S RTP-K93.
*RST: DDR3

**Usage:** Asynchronous command

**Manual operation:** See "Variant" on page 917

**BUS<m>:DDR:DATA:HYSTeresis** <DataHysteresis>

Sets a value for the hysteresis of the data.

**Suffix:**
<m> 1..4

**Parameters:**
<DataHysteresis> Range: 0 to 10
Increment: 0.01
*RST: 0.15
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis" on page 918

**BUS<m>:DDR:DATA:SOURce** <DataSource>

Selects the source for the data output signal.

**Suffix:**
<m> 1..4

**Parameters:**
<DataSource> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15
*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "DQ" on page 917

**BUS<m>:DDR:DATA:THReshold** <DataThreshold>

Sets the threshold value for the digitization of the data line.

**Suffix:**
<m> 1..4

**Parameters:**
<DataThreshold> Range: -10 to 10
Increment: 0.01
*RST: 0.75
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 918

---

**BUS<m>:DDR:STRBe:HYSTeresis** <StrobeHyst>

Sets a value for the hysteresis of the strobe.

**Suffix:**
<m> 1..4

**Parameters:**
<StrobeHyst> Range: 0 to 10
Increment: 0.01
*RST: 0.35
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Hysteresis" on page 918

---

**BUS<m>:DDR:STRBe:SOURce** <StrobeSource>

Selects the source for the strobe signal.

**Suffix:**
<m> 1..4

**Parameters:**
<StrobeSource> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15
*RST: C2W1

**Usage:** Asynchronous command

**Manual operation:** See "DQS" on page 917

---

**BUS<m>:DDR:STRBe:THReshold** <StrobeThreshold>

Sets the threshold value for the digitization of the strobe line.

**Suffix:**
<m> 1..4

**Parameters:**

<StrobeThreshold>

| | |
|---|---|
| Range: | -10 to 10 |
| Increment: | 0.01 |
| *RST: | 0 |
| Default unit: | V |

**Usage:** Asynchronous command

**Manual operation:** See "Threshold" on page 918

---

**BUS<m>:DDR:BITRate** <Bitrate>

Declares the nominal bit rate of the data signal for the DDR eye diagram analysis.

**Suffix:**

<m> 1..4

**Parameters:**

<Bitrate>

| | |
|---|---|
| Range: | 1000 to 10E+9 |
| Increment: | 100E+3 |
| *RST: | 800E+6 |
| Default unit: | bps |

**Usage:** Asynchronous command

**Manual operation:** See "Bit rate (for eye diagram)" on page 918

### 24.17.24.2 Decode results

In all `BUS<m>:DDR:RESult:FRAMe<n>` commands, the suffix <m> selects the serial bus and the suffix <n> selects the frame number in the decode table.

BUS<m>:DDR:RESult:FCOunt? 2296
BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:FVAL? 2297
BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:STARt? 2297
BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:STATus? 2297
BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:STOP? 2298
BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:VAL? 2298
BUS<m>:DDR:RESult:FRAMe<n>:NWRDs? 2299
BUS<m>:DDR:RESult:FRAMe<n>:STARt? 2299
BUS<m>:DDR:RESult:FRAMe<n>:STATe? 2299
BUS<m>:DDR:RESult:FRAMe<n>:STOP? 2300
BUS<m>:DDR:RESult:FRAMe<n>:TYPE? 2300

---

**BUS<m>:DDR:RESult:FCOunt?**

Returns the number of decoded frames in the selected serial bus.

**Suffix:**

<m> 1..4

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:FVAL?**

Returns the formatted value of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**
<FrameByteValue>

**Usage:** Query only

---

**BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:STARt?**

Returns the start time of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:STATus?**

Returns the status of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**

<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**

<State> OK | UNKNown | INComplete | LENGth
*RST: OK

**Usage:** Query only

---

**BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:STOP?**

Returns the stop time of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**

<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**

<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:VAL?**

Returns the value of the field with the selected field number within the frame with the selected frame number (for the selected serial bus).

**Suffix:**

<m> 1..4

<n> *
Selects the frame number.

<o> *
Selects the field number.

**Return values:**

<FrameByteValue> Range: 0 to 0
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:DDR:RESult:FRAMe<n>:NWRDs?**

Returns the number of words for the selected serial bus and frame number.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<NumWords> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

---

**BUS<m>:DDR:RESult:FRAMe<n>:STARt?**

Returns the start time of the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**BUS<m>:DDR:RESult:FRAMe<n>:STATe?**

Returns the status of the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<State> OK | UNKNown | INComplete | LENGth
*RST: OK

**Usage:** Query only

**BUS<m>:DDR:RESult:FRAMe<n>:STOP?**

Returns the stop time of the frame with the selected frame number (for the selected serial bus).

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**BUS<m>:DDR:RESult:FRAMe<n>:TYPE?**

Returns the frame type for the selected serial bus and frame number.

**Suffix:**
<m> 1..4
<n> *

**Return values:**
<FrameType> READ | WRITe | RDNC | WRNC | UNKNown
`RDNC`: read data not continue
`WRNC`: write data not continue
*RST: UNKNown

**Usage:** Query only

### 24.17.24.3 Search settings

There are two commands for each parameter, that you can use for defining the DDR search settings.

For example, to set the *Frame type =Read* > *Field =Data* >*Data* value you can use one of the following commands:

- `SEARch:TRIGger:DDR:FRAMe1:ENABle:DMIN 01100`
  Defines the parameter by using the index <m> for the frame number and <n> for the field number.
- `SEARch:TRIGger:DDR:DMIN "Read", "Data", 01100`
  Defines the parameter by using the frame and field name.

SEARch:TRIGger:DDR:ERENable....2301
SEARch:TRIGger:DDR:ERRor<m>:ENABle....2301
SEARch:TRIGger:DDR:FIENable....2301
SEARch:TRIGger:DDR:FRENable....2302
SEARch:TRIGger:DDR:DMAX....2302

SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMAX....2302
SEARch:TRIGger:DDR:DMIN....2302
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMIN....2302
SEARch:TRIGger:DDR:DOPerator....2303
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DOPerator....2303
SEARch:TRIGger:DDR:FRENable....2303
SEARch:TRIGger:DDR:FRAMe<m>:ENABle....2303
SEARch:TRIGger:DDR:IMAX....2304
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMAX....2304
SEARch:TRIGger:DDR:IMIN....2304
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMIN....2304
SEARch:TRIGger:DDR:IOPerator....2305
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IOPerator....2305

---

**SEARch:TRIGger:DDR:ERENable** <SearchName>,<ErrorName>, <Enabler>
**SEARch:TRIGger:DDR:ERRor<m>:ENABle** <SearchName>,<Enable>
**SEARch:TRIGger:DDR:ERRor<m>:ENABle?** <SearchName>

Defines the error type to be searched for. You can search for all error types in parallel.

**Suffix:**
<m> *
Specifies the error.

**Parameters:**
<Enable> ON | OFF
*RST: ON

**Parameters for setting and query:**
<SearchName> The following errors are available:
`Frame Incomplete`
`Length Error`

**Example:** `SEARch:TRIGger:DDR:ERRor1:ENABle ON`
Enables searching for Frame Incomplete error.
`SEARch:TRIGger:DDR:ERENable "Length Error",ON`
Enables searching for Length Error.

**Manual operation:** See "Error type" on page 921

---

**SEARch:TRIGger:DDR:FIENable** <SearchName>,<Frame>, <Field>, <Enabler>

**Setting parameters:**
<SearchName>

<Frame>

<Field>

<Enabler> ON | OFF

**Usage:** Setting only

**Manual operation:** See "Set details" on page 921

**SEARch:TRIGger:DDR:FRENable** <SearchName>,<Frame>, <Enabler>

**Setting parameters:**
<SearchName>

<Frame>

<Enabler> ON | OFF

**Usage:** Setting only

**Manual operation:** See "DDR search" on page 921

**SEARch:TRIGger:DDR:DMAX** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMAX** <SearchName>,<DataMax>
**SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMAX?** <SearchName>

Sets the end value of a data pattern range if `SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DOPerator` is set to `INRange` or `OORange`.

**Suffix:**
<m> *

<n> *

**Parameters:**
<DataMax>

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Value" on page 921

**SEARch:TRIGger:DDR:DMIN** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMIN** <SearchName>,<DataMin>
**SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMIN?** <SearchName>

Specifies the data pattern to be searched, or sets the start value of a data pattern range to be searched.

**Suffix:**
<m> *

<n> *

**Parameters:**
<DataMin> Specifies the name of the generic trigger setting frame.

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Value" on page 921

**SEARch:TRIGger:DDR:DOPerator** <SearchName>,<Frame>, <Field>, <Operator>
**SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DOPerator**
<SearchName>,<DataOperator>
**SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DOPerator?** <SearchName>

Sets the operator to set a specific data pattern to be searched in the selected field of the selected frame.

**Suffix:**
<m> *

<n> *

**Parameters:**
<DataOperator> EQUal | NEQual | LTHan | LETHan | GTHan | GETHan | INRange | OORange

**EQUal | NEQual | LTHan | LETHan | GTHan | GETHan**
Equal, Not equal, Less than, Less than or equal, Greater than, Greater than or equal. These conditions require one endpoint value to be set using `SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMIN`.

**INRange | OORange**
In range, out of range. These conditions require a range of endpoint values to be set using `SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMIN` and `SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMAX`.

*RST: EQUal

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Value" on page 921

---

**SEARch:TRIGger:DDR:FRENable** <SearchName>,<Frame>, <Enabler>
**SEARch:TRIGger:DDR:FRAMe<m>:ENABle** <SearchName>,<Enable>
**SEARch:TRIGger:DDR:FRAMe<m>:ENABle?** <SearchName>

Enables or disables the specific frame to be searched for.

**Suffix:**
<m> *

**Parameters:**
<Enable> ON | OFF

*RST: OFF

**Parameters for setting and query:**
<SearchName> The following frames are available:
`Read`
`Write`
`Read NC`
`Write NC`

**Manual operation:** See "DDR search" on page 921

---

**SEARch:TRIGger:DDR:IMAX** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMAX** <SearchName>,<IndexMax>
**SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMAX?** <SearchName>

Sets the end value of the bit index range (data position range), if `SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IOPerator` is set to `INRange` or `RANGe`.

**Suffix:**
<m> *

<n> *

**Parameters:**
<IndexMax> Range: 1 to 65534
Increment: 1
*RST: 65534

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Index" on page 922

---

**SEARch:TRIGger:DDR:IMIN** <SearchName>,<Frame>, <Field>, <Data>
**SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMIN** <SearchName>,<IndexMin>
**SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMIN?** <SearchName>

Sets the index at which the data is to be searched, or sets the start value of an index range in which the data is to be searched.

**Suffix:**
<m> *

<n> *

**Parameters:**
<IndexMin> Specifies the index value or the start value of an index range within the field.
The index range, increment and *RST values depend on the field type.

Range: 1 to 65534
Increment: 1
*RST: 1

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Index" on page 922

**SEARch:TRIGger:DDR:IOPerator** <SearchName>,<Frame>, <Field>, <Operator>
**SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IOPerator**
<SearchName>,<IndexOperator>
**SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IOPerator?** <SearchName>

Sets the operator for the index for searching in the selected field of the selected frame.

**Suffix:**
<m> *

<n> *

**Parameters:**
<IndexOperator> EQUal | INRange | RANGe

**EQUal**
This condition requires one endpoint value to be set using `SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMIN`.

**INRange | RANGe**
This condition requires a range of endpoint values to be set using `SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMIN` and `SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMAX`.

*RST: INRange

**Parameters for setting and query:**
<SearchName>

**Manual operation:** See "Index" on page 922

### 24.17.24.4 Search results

In all `SEARch:RESult:DDR:FRAMe<m>` commands, the suffix <m> selects the frame number in the list of search results.

SEARch:RESult:DDR:FCOunt?....2305
SEARch:RESult:DDR:FRAMe<m>:FLD<n>:FVAL?....2306
SEARch:RESult:DDR:FRAMe<m>:FLD<n>:STARt?....2306
SEARch:RESult:DDR:FRAMe<m>:FLD<n>:STATus?....2306
SEARch:RESult:DDR:FRAMe<m>:FLD<n>:STOP?....2307
SEARch:RESult:DDR:FRAMe<m>:FLD<n>:VAL?....2307
SEARch:RESult:DDR:FRAMe<m>:NWRDs?....2308
SEARch:RESult:DDR:FRAMe<m>:STARt?....2308
SEARch:RESult:DDR:FRAMe<m>:STATe?....2308
SEARch:RESult:DDR:FRAMe<m>:STOP?....2309
SEARch:RESult:DDR:FRAMe<m>:TYPE?....2309

**SEARch:RESult:DDR:FCOunt?** <SearchName>

Returns the number of decoded frames in the search result.

**Query parameters:**
<SearchName>

**Return values:**
<FrameCount> Range: 0 to 100000
Increment: 1
*RST: 0

**Usage:** Query only

---

**SEARch:RESult:DDR:FRAMe<m>:FLD<n>:FVAL?** <SearchName>

Returns the formatted value of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *
Selects the field number.

**Query parameters:**
<SearchName>

**Return values:**
<BteFormattedVal>

**Usage:** Query only

---

**SEARch:RESult:DDR:FRAMe<m>:FLD<n>:STARt?** <SearchName>

Returns the start time of the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *
Selects the field number.

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:DDR:FRAMe<m>:FLD<n>:STATus?** <SearchName>

Returns the field status of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | UNKNown | INComplete | LENGth
*RST: OK

**Usage:** Query only

---

**SEARch:RESult:DDR:FRAMe<m>:FLD<n>:STOP?** <SearchName>

Returns the stop time of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *
Selects the field number.

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

---

**SEARch:RESult:DDR:FRAMe<m>:FLD<n>:VAL?** <SearchName>

Returns the value of the field with the selected field number within the frame with the selected frame number within the search result.

**Suffix:**
<m> *

<n> *
Selects the field number.

**Query parameters:**
<SearchName>

**Return values:**
<ByteValue> Range: 0 to 0
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:DDR:FRAMe<m>:NWRDs?** <SearchName>

Returns the number of words of the selected frame in the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<NumWords> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only

**SEARch:RESult:DDR:FRAMe<m>:STARt?** <SearchName>

Returns the start time of the frame with the selected frame number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Start> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:DDR:FRAMe<m>:STATe?** <SearchName>

Returns the status of the frame with the selected frame number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<State> OK | UNKNown | INComplete | LENGth
*RST: OK

**Usage:** Query only

**SEARch:RESult:DDR:FRAMe<m>:STOP?** <SearchName>

Returns the stop time of the frame with the selected frame number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<Stop> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0
Default unit: s

**Usage:** Query only

**SEARch:RESult:DDR:FRAMe<m>:TYPE?** <SearchName>

Returns the frame type for the selected frame number within the search result.

**Suffix:**
<m> *

**Query parameters:**
<SearchName>

**Return values:**
<FrameType> READ | WRITe | RDNC | WRNC | UNKNown

`RDNC`: read data not continue
`WRNC`: write data not continue

*RST: UNKNown

**Usage:** Query only

## 24.18 Mixed signal option (MSO, R&S RTP-B1)

This chaptes decribes the remote commands of MSO option R&S RTP-B1.

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"

- Chapter 24.3, "Programming examples", on page 1172

- Digital channels....2310
- Parallel bus configuration....2313
- Digital resolution....2320
- Trigger settings for digital signals and parallel buses....2321
- MSO data....2332

## 24.18.1 Digital channels

All `DIGital:` commands affect only the settings of the first MSO bus (Bus1). The settings of all other parallel buses (Bus 2, 3,4) remain unchanged.

DIGital<m>:DISPlay....2310
DIGital<m>:TECHnology....2310
DIGital<m>:THReshold....2311
DIGital<m>:THCoupling....2311
DIGital<m>:HYSTeresis....2311
DIGital<m>:LABel....2312
DIGital<m>:DESKew....2312

---

**DIGital<m>:DISPlay** <Display>

Enables or disables the indicated digital channel, displays it, and enables the parallel Bus1 if the bus was disabled. That is, `BUS<m>:PARallel:DISPlay:SHDI` and `BUS<m>:PARallel:STATe` are set to ON automatically.

If another active bus already uses the selected digital channel, the instrument disables the other bus to avoid conflicts.

For Bus1, the `DIG:DISP` command has the same effect as `BUS<m>:PARallel:BIT<n>[:STATe]`. To enable digital channels for buses 2, 3 and 4, use the `BUS:PAR:BIT[:STAT]` command.

**Suffix:**
<m> 0..15
Number of the digital channel

**Parameters:**
<Display> ON | OFF

---

**DIGital<m>:TECHnology** <Technology>

Selects the threshold voltage for various types of integrated circuits and applies it to all digital channels.

The setting affects only the settings of the first MSO bus (Bus1). You can set the technology value for all buses with `BUS<m>:PARallel:TECHnology`.

**Suffix:**
<m> 0..15
The suffix is irrelevant.

**Parameters:**
<Technology> V15 | V25 | V165 | V125 | V09 | VM13 | V38 | V20 | V0 | MANual
See BUS<m>:PARallel:TECHnology

**Manual operation:** See "Technology, Threshold" on page 939

---

**DIGital<m>:THReshold** <Value>

Sets the logical threshold for the channel group to which the indicated digital channel belongs. The setting affects only the settings of the first MSO bus (Bus1).

The setting affects only the settings of the first MSO bus (Bus1). You can set the threshold for all buses with BUS<m>:PARallel:TECHnology or BUS<m>:PARallel:THReshold<n>.

See also: DIGital<m>:THCoupling

**Suffix:**
<m> 0..15
Number of the digital channel.
Channel groups: 0..3; 4..7; 8..11; 12..15

**Parameters:**
<Value> Range: -8.0 to 8.0
Increment: 200.0e-12
*RST: 0
Default unit: V

**Manual operation:** See "Technology, Threshold" on page 939

---

**DIGital<m>:THCoupling** <State>

Sets the threshold and the hysteresis for all digital channels of parallel bus 1 to the same value.

The command BUS<m>:PARallel:THCoupling is used to set all buses.

**Suffix:**
<m> 0..15
The suffix is irrelevant.

**Parameters:**
<State> ON | OFF

**Manual operation:** See "Couple thresholds" on page 940

---

**DIGital<m>:HYSTeresis** <Hysteresis>

Defines the size of the hysteresis to avoid the change of signal states due to noise for the channel group to which the indicated digital channel belongs.

The setting affects only the settings of the first MSO bus (Bus1). You can set the hysteresis for all buses with BUS<m>:PARallel:HYSTeresis<n>.

See also: DIGital<m>:THCoupling

**Suffix:**
<m> 0..15
Number of the digital channel
Channel groups: 0..3; 4..7; 8..11; 12..15

**Parameters:**
<Hysteresis> MAXIMUM | MAXimum | ROBUST | ROBust | NORMAL | NORMal

**MAXIMUM = MAXimum**
Maximum value that is possible and useful for the signal and its settings

**ROBUST = ROBust**
Different hysteresis values for falling and rising edges to avoid an undefined state of the trigger system.

**NORMAL = NORMal**
The instrument sets a value suitable for the signal and its settings.

**Manual operation:** See "Hysteresis" on page 941

---

**DIGital<m>:LABel** <Label>

Sets a name for the indicated digital channel. The name is displayed in the diagram.

The setting affects only the settings of the first MSO bus (Bus1). You can set the label for all buses with BUS<m>:PARallel:BIT<n>:LABel.

**Suffix:**
<m> 0..15
Number of the digital channel

**Parameters:**
<Label> String containing the channel name

---

**DIGital<m>:DESKew** <Deskew>

Sets an individual delay for each digital channel to time-align it with other digital channels. The deskew value compensates delays that are known from the circuit specifics or caused by the different length of cables.

The setting affects only the settings of the first MSO bus (Bus1). You can set the deskew for all buses with BUS<m>:PARallel:BIT<n>:DESKew.

**Suffix:**
<m> 0..15
Number of the digital channel

**Parameters:**

<Deskew> Range: -200.0E-09 to 200.0E-09
Increment: 200.0E-12
*RST: 0
Default unit: s

**Manual operation:** See "Skew" on page 944

## 24.18.2 Parallel bus configuration

The following commands configure the four parallel buses of R&S RTP-B1.

BUS<m>:PARallel:STATe....2313
BUS<m>:PARallel:BIT<n>[:STATe]....2314
BUS<m>:PARallel:THReshold<n>....2314
BUS<m>:PARallel:TECHnology....2315
BUS<m>:PARallel:THCoupling....2315
BUS<m>:PARallel:HYSTeresis<n>....2315
BUS<m>:PARallel:BIT<n>:DESKew....2316
BUS<m>:PARallel:DESoffset....2316
BUS<m>:PARallel:BIT<n>:LABel....2317
BUS<m>:PARallel:DATA:FORMat....2317
BUS<m>:PARallel:DISPlay:SHDI....2317
BUS<m>:PARallel:DISPlay:SHBU....2318
BUS<m>:PARallel:DISPlay:BTYP....2318
BUS<m>:PARallel:CLON....2318
BUS<m>:PARallel:CLOCk....2319
BUS<m>:PARallel:CLSLope....2319
BUS<m>:PARallel:CLEar....2319
BUS<m>:PARallel:DECTable<n>:SHOW....2320
BUS<m>:PARallel:DECTable<n>:COUNt?....2320
BUS<m>:PARallel:DECTable<n>:DATA?....2320

---

**BUS<m>:PARallel:STATe** <Enable>

Enables or disables the indicated parallel bus. The threshold settings of the bus take effect for all *active* parallel buses.

Dependencies: At least one digital channel must be enabled for the selected bus, otherwise the command does not work. The bus is enabled automatically if the first digital channel is enabled with `BUS<m>:PARallel:BIT<n>[:STATe]` or `DIGital<m>:DISPlay`.

**Suffix:**

<m> 1..4
Selects the parallel bus.

**Parameters:**

<Enable> ON | OFF
*RST: OFF

**Manual operation:** See "Enable bus" on page 939

**BUS<m>:PARallel:BIT<n>[:STATe]** <Assigned>

Assigns the selected digital channel to the indicated bus, displays it, and enables the bus if the bus was disabled. That is, `BUS<m>:PARallel:DISPlay:SHDI` and `BUS<m>:PARallel:STATe` are set to ON automatically.

If another active bus already uses the selected digital channel, the instrument disables the other bus to avoid conflicts.

For parallel bus 1, the `BUS:PAR:BIT[:STATe]` command has the same effect as `DIGital<m>:DISPlay`.

**Suffix:**

<m> 1..4
Selects the parallel bus.

<n> 0..15
Selects the bit of the bus word. Each bit corresponds to a digital channel.

**Parameters:**

<Assigned> ON | OFF
*RST: OFF

**BUS<m>:PARallel:THReshold<n>** <Threshold>

Sets the logical threshold for the indicated channel group.

Alternatively, you can set the threshold with `BUS<m>:PARallel:TECHnology`. For the parallel bus 1, you can also use `DIGital<m>:THReshold`.

See also: `DIGital<m>:THCoupling`

**Suffix:**

<m> 1..4
Selects the parallel bus.

<n> 1..4
Selects the channel group:
1 = dig. channels 0..3;
2 = dig. channels 4..7
3 = dig. channels 8..11
4 = dig. channels 12..15

**Parameters:**

<Threshold> Range: -8.0 to 8.0
Increment: 200.0e-12
*RST: 0
Default unit: V

**Manual operation:** See "Technology, Threshold" on page 939

**BUS<m>:PARallel:TECHnology** <Technology>

Selects the threshold voltage for various types of integrated circuits and applies it to all digital channels.

**Suffix:**
<m> 1..4
Selects the parallel bus.

**Parameters:**
<Technology> V15 | V25 | V165 | V125 | V09 | VM13 | V38 | V20 | V0 | MANual
V15: TTL
V25: CMOS 5.0 V
V165: CMOS 3.3 V
V125: CMOS 2.5 V
V09: CMOS 1.85 V
VM13: ECL, -1.3 V
V38: PECL
V20: LVPECL
V0: Ground
MANual: Set a user-defined threshold value with `DIGital<m>:THReshold`
*RST: V165

**Manual operation:** See "Technology, Threshold" on page 939

**BUS<m>:PARallel:THCoupling** <LevelCoupling>

Sets the threshold for all digital channels of the selected bus to the same value. Also the hysteresis value is applied to all digital channels.

Tor parallel bus 1, the command `DIGital<m>:THCoupling` has the same effect.

**Suffix:**
<m> 1..4
The suffix is irrelevant.

**Parameters:**
<LevelCoupling> ON | OFF
*RST: ON

**Manual operation:** See "Couple thresholds" on page 940

**BUS<m>:PARallel:HYSTeresis<n>** <Hysteresis>

Defines the size of the hysteresis for the channel group to avoid the change of signal states due to noise.

For the parallel bus 1, you can also use `DIGital<m>:HYSTeresis`.

**Suffix:**
<m> 1..4
Selects the parallel bus.

<n> 1..4
Selects the channel group:
1 = dig. channels 0..3;
2 = dig. channels 4..7
3 = dig. channels 8..11
4 = dig. channels 12..15

**Parameters:**

<Hysteresis> MAXIMUM | MAXimum | ROBUST | ROBust | NORMAL | NORMal

**MAXIMUM = MAXimum**
Maximum value that is possible and useful for the signal and its settings

**ROBUST = ROBust**
Different hysteresis values for falling and rising edges to avoid an undefined state of the trigger system.

**NORMAL = NORMal**
The instrument sets a value suitable for the signal and its settings.

**Manual operation:** See "Hysteresis" on page 941

---

**BUS<m>:PARallel:BIT<n>:DESKew** <Deskew>

Sets an individual delay for each digital channel to time-align it with other digital channels. The deskew value compensates delays that are known from the circuit specifics or caused by the different length of cables.

For the parallel bus 1, you can also use `DIGital<m>:DESKew`.

**Suffix:**

<m> 1..4
Selects the parallel bus.

<n> 0..15
Selects the bit of the bus word. Each bit corresponds to a digital channel.

**Parameters:**

<Deskew>
Range: -200E-9 to 200E-9
Increment: 200E-12
*RST: 0
Default unit: s

**Manual operation:** See "Skew" on page 944

---

**BUS<m>:PARallel:DESoffset** <DeskewOffset>

Sets a general delay for all digital channels. The resulting deskew of a digital channel is the sum of this general value and the individual value set with `BUS<m>:PARallel:BIT<n>:DESKew`.

**Suffix:**
<m> 1..4
Selects the parallel bus.

**Parameters:**
<DeskewOffset> Range: -200E-9 to 200E-9
Increment: 200E-12
*RST: 0
Default unit: s

**Manual operation:** See "Global skew" on page 944

---

**BUS<m>:PARallel:BIT<n>:LABel** <Label>

Sets a name for the indicated digital channel. The name is displayed in the diagram.

For the parallel bus 1, you can also use `DIGital<m>:LABel`.

**Suffix:**
<m> 1..4
Selects the parallel bus.

<n> 0..15
Selects the bit of the bus word. Each bit corresponds to a digital channel.

**Parameters:**
<Label> String containing the channel name

---

**BUS<m>:PARallel:DATA:FORMat** <DataFormat>

Sets the number format for decoded data values on the display for the indicated parallel bus.

It sets also the number representation for remote data transfer with `BUS<m>:PARallel:DATA[:VALues]?`.

**Suffix:**
<m> 1..4

**Parameters:**
<DataFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG
*RST: HEX

**Manual operation:** See "Data format" on page 942

---

**BUS<m>:PARallel:DISPlay:SHDI** <ShwDigSigns>

If enabled, the selected digital signals are shown in the diagram. Each channel is displayed as a logic signal.

See also: `DIGital<m>:DISPlay`

**Suffix:**
<m> 1..4
Selects the parallel bus.

**Parameters:**
<ShwDigSigns> ON | OFF
*RST: OFF

**Manual operation:** See "Show channels" on page 939

---

**BUS<m>:PARallel:DISPlay:SHBU** <ShowBus>

Shows or hides the indicated parallel bus. If enabled, the resulting bus signal and bus values are displayed in the diagram.

**Suffix:**
<m> 1..4
Selects the parallel bus.

**Parameters:**
<ShowBus> ON | OFF
*RST: OFF

**Manual operation:** See "Show bus" on page 942

---

**BUS<m>:PARallel:DISPlay:BTYP** <BusRepres>

Selects the display type of the indicated parallel bus.

**Suffix:**
<m> 1..4
Selects the parallel bus.

**Parameters:**
<BusRepres> COMB | ANALog

**COMB**
Displays the decoded bus signal with bus values.

**ANALog**
Displays the bus value as amplitude, similar to an analog waveform.

*RST: COMB

**Manual operation:** See "Bus display" on page 942

---

**BUS<m>:PARallel:CLON** <Clocked>

Defines if the bus is a clocked bus - one of the digital channels serves as clock of the bus.

**Suffix:**
<m> 1 | 2
Selects the parallel bus. The clocked bus is available on parallel buses 1 and 2.

**Parameters:**
<Clocked> ON | OFF
*RST: OFF

**Manual operation:** See "Enable" on page 942

---

**BUS<m>:PARallel:CLOCk** <ClockSource>

Selects the digital channel used as clock of the indicated parallel bus.

**Suffix:**
<m> 1 | 2
Selects the parallel bus. The clocked bus is available on parallel buses 1 and 2.

**Parameters:**
<ClockSource> D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15
Clock channel
*RST: D1

**Manual operation:** See "Clock source" on page 943

---

**BUS<m>:PARallel:CLSLope** <ClockSlope>

Selects the slope of the clock signal at which all digital channels of the bus are analyzed.

**Suffix:**
<m> 1 | 2
Selects the parallel bus. The clocked bus is available on parallel buses 1 and 2.

**Parameters:**
<ClockSlope> POSitive | NEGative | EITHer
*RST: POSitive

**Manual operation:** See "Clock slope" on page 943

---

**BUS<m>:PARallel:CLEar**

Removes all assigned digital channels from the bus

**Suffix:**
<m> 1..4
Selects the parallel bus.

**Usage:** Event

---

**BUS<m>:PARallel:DECTable<n>:SHOW** <ShowDecodeTable>

Available only if `BUS<m>:PARallel:CLON` is set to `ON`.

If enabled, a results box opens with decoded values of the bus signal and its time. You can query the number of values (`BUS<m>:PARallel:DECTable<n>:COUNt?`) and their data value (`BUS<m>:PARallel:DECTable<n>:DATA?`).

**Suffix:**
<m> 1..4

<n> 1..5

**Parameters:**
<ShowDecodeTable> ON | OFF

*RST: OFF

**Manual operation:** See "Show table" on page 942

---

**BUS<m>:PARallel:DECTable<n>:COUNt?**

Returns the number of clock edges in the decode table.

**Suffix:**
<m> 1..4

<n> 1..5

**Usage:** Query only

**Manual operation:** See "Show table" on page 942

---

**BUS<m>:PARallel:DECTable<n>:DATA?**

Returns the data of the decoded values of the bus signal and its time.

**Suffix:**
<m> 1..4

<n> 1..5

**Usage:** Query only

**Manual operation:** See "Show table" on page 942

## 24.18.3 Digital resolution

ACQuire:DRESolution?....2321
ACQuire:POINts:DVALue?....2321

**ACQuire:DRESolution?**

Returns the current digital resolution of the digital channels.

**Return values:**
<DigRes> Default unit: s

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Dig. resolution" on page 945

**ACQuire:POINts:DVALue?**

Returns the current digital record length used by each digital channel.

**Return values:**
<DigReclgth> Range: 1000 to 200E6
Increment: 2
*RST: 1000
Default unit: pts

**Usage:** Query only
Asynchronous command

## 24.18.4 Trigger settings for digital signals and parallel buses

In all `TRIGger<m>:PARallel` commands, the trigger suffix <m> is always 1 and can be omitted. It selects the trigger event: Only the A-trigger is available for triggering on digital signals and parallel buses.


- General commands....2321
- Edge trigger....2324
- Width trigger....2324
- Timeout trigger....2326
- Setup & Hold trigger....2326
- State trigger....2328
- Pattern trigger....2328
- Serial pattern trigger....2331


### 24.18.4.1 General commands


TRIGger<m>:SOURce[:SELect]....2322
TRIGger<m>:PARallel:TYPE....2323
TRIGger<m>:PARallel:SETHold:CSOurce[:VALue]....2323
TRIGger<m>:PARallel:STATe:CSOurce:VALue....2323
TRIGger<m>:PARallel:SPATtern:CSOurce[:VALue]....2323
TRIGger<m>:PARallel:EDGE:EXPRession[:DEFine]....2323
TRIGger<m>:PARallel:WIDTh:EXPRession[:DEFine]....2323
TRIGger<m>:PARallel:TIMeout:EXPRession[:DEFine]....2323

TRIGger<m>:PARallel:STATe:EXPRession[:DEFine]....2323
TRIGger<m>:PARallel:PATTern:EXPRession[:DEFine]....2323
TRIGger<m>:PARallel:SPATtern:EXPRession[:DEFine]....2323

**TRIGger<m>:SOURce[:SELect]** <SourceDetailed>

Selects the source of the trigger signal.

**Suffix:**

<m> 1..3
1 = A-trigger, 2 = B-trigger, 3 = R-trigger
Available values depend on the selected trigger source. For input channels CHAN1...4, a trigger sequence can be configured.
For all other trigger sources, only suffix 1 is allowed.

**Parameters:**

<SourceDetailed> CHAN1 | CHANnel1 | CHAN2 | CHANnel2 | CHAN3 | CHANnel3 | CHAN4 | CHANnel4 | EXTernanalog | SBUS | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | LOGIC | MSOB1 | MSOB2 | MSOB3 | MSOB4 | Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4 | DIFF1 | DIFF2 | COMMON1 | COMMON2 | LINE

**CHAN1 = CHANnel1, CHAN2 = CHANnel2, CHAN3 = CHANnel3, CHAN4 = CHANnel4**
Input channels

**EXTernanalog**
External analog signal connected to the External Trigger Input. For this source, only the analog edge trigger is available.

**LINE**
The instrument generates the trigger from the AC power input and synchronizes the signal to the AC power frequency.

**SBUS**
Serial bus

**D0...D15**
Digital channels (option R&S RTP-B1)
See also: Chapter 24.18.4, "Trigger settings for digital signals and parallel buses", on page 2321

**LOGIc**
Logic combination of digital channels, used as trigger source (option R&S RTP-B1)

**MSOB1 | MSOB2 | MSOB3 | MSOB4**
Parallel bus (option R&S RTP-B1)

**Z1V1 | Z1V2 | Z1V3 | Z1V4 | Z1I1 | Z1I2 | Z1I3 | Z1I4 | Z2V1 | Z2V2 | Z2V3 | Z2V4 | Z2I1 | Z2I2 | Z2I3 | Z2I4**
Input channels of R&S RT-ZVCmulti-channel power probe. Only available in the A-trigger with trigger type EDGE.

**DIFF1 | DIFF2 | COMMON1 | COMMON2**
Differential signals

*RST: CHAN1

**Usage:** Asynchronous command

**Manual operation:** See "[Source]" on page 37

---

**TRIGger<m>:PARallel:TYPE** <Type>

Selects the trigger type to trigger on digital channels and parallel buses.

To trigger on analog channels or the external trigger input, use `TRIGger<m>:TYPE`.

**Parameters:**

<Type> EDGE | WIDTh | TIMeout | SETHold | DATatoclock | STATe | PATTern | SERPattern |

SETHold = DATatoclock, setup & hold trigger
SERPattern = serial pattern trigger

*RST: EDGE

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 951

---

**TRIGger<m>:PARallel:SETHold:CSOurce[:VALue]** <ClockSource>
**TRIGger<m>:PARallel:STATe:CSOurce:VALue** <ClockSource>
**TRIGger<m>:PARallel:SPATtern:CSOurce[:VALue]** <ClockSource>

Selects the digital channel of the clock signal.

**Parameters:**

<ClockSource> D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

*RST: D0

**Usage:** Asynchronous command

**Manual operation:** See "Clock source" on page 956

---

**TRIGger<m>:PARallel:EDGE:EXPRession[:DEFine]** <LogicalExpr>
**TRIGger<m>:PARallel:WIDTh:EXPRession[:DEFine]** <LogicalExpr>
**TRIGger<m>:PARallel:TIMeout:EXPRession[:DEFine]** <LogicalExpr>
**TRIGger<m>:PARallel:STATe:EXPRession[:DEFine]** <LogicalExpr>
**TRIGger<m>:PARallel:PATTern:EXPRession[:DEFine]** <LogicalExpr>
**TRIGger<m>:PARallel:SPATtern:EXPRession[:DEFine]** <LogicalExpr>

Defines a logical combination of several digital channels as trigger condition if `TRIGger<m>:SOURce[:SELect]` is set to LOGIC.

**Parameters:**

<LogicalExpr> String with logical expression

**Example:** `TRIGger:PARallel:EDGE:EXPRession 'D1 and D2'`

**Usage:** Asynchronous command

**Manual operation:** See "Logical expression" on page 955

### 24.18.4.2 Edge trigger

See also:

- `TRIGger<m>:PARallel:EDGE:EXPRession[:DEFine]` on page 2323

TRIGger<m>:PARallel:EDGE:SLOPe....2324

**TRIGger<m>:PARallel:EDGE:SLOPe** <Slope>

Defines the edge - the state transition - of the signal to trigger on a single digital channel (a logic bit), or a logical combination of digital channels.

**Parameters:**

<Slope> POSitive | NEGative | EITHer

*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 952

### 24.18.4.3 Width trigger

See also:

- `TRIGger<m>:PARallel:WIDTh:EXPRession[:DEFine]` on page 2323

TRIGger<m>:PARallel:WIDTh:POLarity....2324
TRIGger<m>:PARallel:WIDTh:RANGe....2325
TRIGger<m>:PARallel:WIDTh:WIDTh....2325
TRIGger<m>:PARallel:WIDTh:DELTa....2325

**TRIGger<m>:PARallel:WIDTh:POLarity** <Polarity>

Sets the polarity of a pulse. When triggering on a positive pulse, the trigger event occurs on the high to low transition of the pulse if the timing condition is true. When triggering on a negative pulse, the trigger event occurs on the low to high transition of the pulse if the timing condition is true.

**Parameters:**

<Polarity> POSitive | NEGative | EITHer

*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Polarity" on page 953

**TRIGger<m>:PARallel:WIDTh:RANGe** <RangeMode>

Selects how the range of a pulse width is defined:

**Parameters:**

<RangeMode> WITHin | OUTSide | SHORter | LONGer

**WITHin | OUTSide**
Pulses inside or outside a given time range. The time range *Width ± Delta* is defined by `TRIGger<m>:PARallel:WIDTh:WIDTh` and `TRIGger<m>:PARallel:WIDTh:DELTa`.

**SHORter | LONGer**
Pulses shorter or longer than a given width defined by `TRIGger<m>:PARallel:WIDTh:WIDTh`

*RST: WITHin

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 953

**TRIGger<m>:PARallel:WIDTh:WIDTh** <Width>

Sets the limit for the pulse width.

The effect depends on `TRIGger<m>:PARallel:WIDTh:RANGe`.

- For the ranges SHORter and LONGer, the width defines the maximum and minimum pulse width, respectively.
- For the ranges WITHin and OUTSide, the width defines the center of a range which is defined by the limits `TRIGger<m>:PARallel:WIDTh:DELTa`.

**Parameters:**

<Width>
Range: 200E-12 to 10000
Increment: 200E-9
*RST: 5E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Width" on page 954

**TRIGger<m>:PARallel:WIDTh:DELTa** <WidthDelta>

Defines a range around the given width value. the setting is relevant if `TRIGger<m>:PARallel:WIDTh:RANGe` is set to WITHin or OUTSide. The width is set with `TRIGger<m>:PARallel:WIDTh:WIDTh`.

**Parameters:**

<WidthDelta>
Range: 0 to 432
Increment: 600E-12
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "±Delta" on page 954

### 24.18.4.4 Timeout trigger

See also:

- `TRIGger<m>:PARallel:TIMeout:EXPRession[:DEFine]` on page 2323

TRIGger<m>:PARallel:TIMeout:RANGe ..... 2326
TRIGger<m>:PARallel:TIMeout:TIME ..... 2326

---

**TRIGger<m>:PARallel:TIMeout:RANGe** <TimeoutMode>

Sets the state condition.

**Parameters:**

<TimeoutMode> HIGH | LOW | EITHer

**HIGH**
The level of a digital channel stays above the threshold, or the logical expression for LOGic trigger source is true.

**LOW**
The level of a digital channel stays below the threshold, or the logical expression for LOGic trigger source is false.

**EITHer**
The signal state remains unchanged.

*RST: HIGH

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 954

---

**TRIGger<m>:PARallel:TIMeout:TIME** <Time>

Defines the time limit for the timeout at which the instrument triggers.

**Parameters:**

<Time>
Range: 100E-12 to 10000
Increment: 100E-9
*RST: 100E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Time" on page 955

### 24.18.4.5 Setup & Hold trigger

See also:

- `TRIGger<m>:PARallel:SETHold:CSOurce[:VALue]` on page 2323

TRIGger<m>:PARallel:SETHold:CSOurce:EDGE....2327
TRIGger<m>:PARallel:SETHold:HTIMe....2327
TRIGger<m>:PARallel:SETHold:STIMe....2327

**TRIGger<m>:PARallel:SETHold:CSOurce:EDGE** <ClockEdge>

Sets the edge of the clock signal. The crossing of the clock edge and the logical threshold defines the time reference point for the setup and hold time measurement.

**Parameters:**

<ClockEdge> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Clock edge" on page 956

**TRIGger<m>:PARallel:SETHold:HTIMe** <HoldTime>

Sets the minimum time *after* the clock edge while data should be stable and not change its state.

See also: "Hold time" on page 956

**Parameters:**

<HoldTime> Range: -99.8E-9 to 100E-9
Increment: 1E-9
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Hold time" on page 956

**TRIGger<m>:PARallel:SETHold:STIMe** <SetupTime>

Sets the minimum time *before* the clock edge while data should be stable and not change its state.

See also: "Setup time" on page 956

**Parameters:**

<SetupTime> Range: -99.8E-9 to 100E-9
Increment: 1E-9
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Setup time" on page 956

### 24.18.4.6 State trigger

See also:

- `TRIGger<m>:PARallel:STATe:CSOurce:VALue` on page 2323
- `TRIGger<m>:PARallel:STATe:EXPRession[:DEFine]` on page 2323

TRIGger<m>:PARallel:STATe:CSOurce:EDGE....2328
TRIGger<m>:PARallel:STATe:BIT<0..15>....2328

---

**TRIGger<m>:PARallel:STATe:CSOurce:EDGE** <Slope>

Sets the edge of the clock signal. The crossing of the clock edge and the logical threshold defines the time at which the logical states and the bus value are analyzed.

**Parameters:**

<Slope> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Clock edge" on page 957

---

**TRIGger<m>:PARallel:STATe:BIT<0..15>** <Bit>

Sets the required state for each digital channel that is used in the bus.

**Parameters:**

<Bit> HIGH | LOW | DONTCARE | DONTcare
Bit value: 1 (HIGH), 0 (LOW), or X (DONTCARE = DONTcare)

**Usage:** Asynchronous command

**Manual operation:** See "Channel states" on page 957

### 24.18.4.7 Pattern trigger

---

**TRIGger<m>:PARallel:PATTern:BIT<0..15>** <Bit>

Sets the required state for each digital channel that is used in the bus.

**Parameters:**

<Bit> HIGH | LOW | DONTCARE | DONTcare
Bit value: 1 (HIGH), 0 (LOW), or X (DONTCARE = DONTcare)

**Usage:** Asynchronous command

**Manual operation:** See "Channel states" on page 958

**TRIGger<m>:PARallel:PATTern:MODE** <Mode>

Sets the mode of the timing condition.

**Parameters:**

<Mode> OFF | TIMeout | WIDTh

**OFF**
No timing condition, only the logical pattern condition is relevant.

**TIMeout**
Defines a minimum time qualification to avoid triggering on unstable or transitional conditions. Use `TRIGger<m>:PARallel:PATTern:TIMeout:MODE` and `TRIGger<m>:PARallel:PATTern:TIMeout[:TIME]` to specify the timeout.

**WIDTh**
Sets a pulse width as timing condition. The pulse starts when the pattern comes true, and the trigger event occurs when the pattern comes false during the specified time limit. Use `TRIGger<m>:PARallel:PATTern:WIDTh:RANGe`, `TRIGger<m>:PARallel:PATTern:WIDTh[:WIDTh]`, and `TRIGger<m>:PARallel:PATTern:WIDTh:DELTa` to specify the width.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Mode: Off, Timeout, Width" on page 959

**TRIGger<m>:PARallel:PATTern:TIMeout:MODE** <TimeoutMode>

Sets the state condition for the timeout qualification if `TRIGger<m>:PARallel:PATTern:MODE` is set to TIMeout. To set the time limit, use `TRIGger<m>:PARallel:PATTern:TIMeout[:TIME]`.

**Parameters:**

<TimeoutMode> HIGH | LOW | EITHer

HIGH: The pattern stays true for the specified time.
LOW: The pattern stays false for the specified time.
EITHER: The pattern remains unchanged for the specified time.

*RST: HIGH

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 959

**TRIGger<m>:PARallel:PATTern:TIMeout[:TIME]** <Time>

Defines the time limit for the timeout at which the instrument triggers.

**Parameters:**

<Time>
Range: 100E-12 to 10000
Increment: 100E-9
*RST: 100E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Time" on page 959

---

**TRIGger<m>:PARallel:PATTern:WIDTh:RANGe** <WidthRangeMode>

Selects how the range of a pulse width is defined if `TRIGger<m>:PARallel:PATTern:MODE` is set to WIDTh.

**Parameters:**

<WidthRangeMode> WITHin | OUTSide | SHORter | LONGer

**WITHin**
Triggers when the pattern comes false inside a given time range. The time limit is defined by `TRIGger<m>:PARallel:PATTern:WIDTh[:WIDTh]` and `TRIGger<m>:PARallel:PATTern:WIDTh:DELTa` (*Width ± Delta*).

**OUTSide**
Triggers when the pattern comes false before or after the given time range. The time limit definition is the same as for WITHin range.

**SHORter | LONGer**
Triggers when the pattern comes false before or after the given width has expired. Width is set with `TRIGger<m>:PARallel:PATTern:WIDTh[:WIDTh]`.

*RST: WITHin

**Usage:** Asynchronous command

**Manual operation:** See "Range" on page 960

---

**TRIGger<m>:PARallel:PATTern:WIDTh[:WIDTh]** <Width>

The effect depend on the setting of the `TRIGger<m>:PARallel:PATTern:WIDTh:RANGe`command.

For the ranges SHOrter and LONGer, the width defines the maximum and minimum time limit, respectively.

For the ranges WITHin and OUTSide, the width defines the center of a range which is defined by the limits "±Delta".

**Parameters:**

<Width>
Range: 100E-12 to 10000
Increment: 100E-9
*RST: 5E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Width" on page 960

---

**TRIGger<m>:PARallel:PATTern:WIDTh:DELTa** <WidthDelta>

Defines a range around the width value set with `TRIGger<m>:PARallel:PATTern:WIDTh[:WIDTh]`.

**Parameters:**

<WidthDelta>
Range: 0 to 432
Increment: 500E-12
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "±Delta" on page 960

### 24.18.4.8 Serial pattern trigger

See also:

- `TRIGger<m>:PARallel:SPATtern:CSOurce[:VALue]` on page 2323
- `TRIGger<m>:PARallel:SPATtern:EXPRession[:DEFine]` on page 2323

---

**TRIGger<m>:PARallel:SPATtern:CSOurce:EDGE** <ClockEdge>

Sets the edge of the clock signal. The bit value is determined at the crossing of the clock edge and the logical threshold.

**Parameters:**

<ClockEdge>
POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Clock edge" on page 961

---

**TRIGger<m>:PARallel:SPATtern:PATTern** <Pattern>

Defines the serial bit string on which to trigger.

**Parameters:**

<Pattern>
Numeric or string pattern, see Chapter 24.4.6, "Bit pattern parameter", on page 1195. The bit value X (don't care) is not allowed.

**Usage:** Asynchronous command

**Manual operation:** See "Pattern" on page 961

### 24.18.5 MSO data

To export data of digital channels and parallel buses to file, use the following commands:

- `EXPort:WAVeform:SOURce` on page 1614
- `EXPort:WAVeform:NAME` on page 1616
- `EXPort:WAVeform:SAVE` on page 1617

The remote data transfer from the instrument to the controlling computer is performed using the following commands:

DIGital<m>:DATA:HEADer? ..... 2332
DIGital<m>:DATA[:VALues]? ..... 2332
BUS<m>:PARallel:DATA:FORMat ..... 2333
BUS<m>:PARallel:DATA:HEADer? ..... 2333
BUS<m>:PARallel:DATA[:VALues]? ..... 2333

#### DIGital<m>:DATA:HEADer?

Returns the header of digital channel data

*Table 24-17: Header data*

| Position | Meaning | Example |
|---|---|---|
| 1 | XStart, acquisition time before trigger, in s | -5E-008 = - 50 ns |
| 2 | XStop, acquisition time after trigger, in s | 5E-008 = 50 ns |
| 3 | Record length of the waveform in Samples | 1000 |
| 4 | Number of values per sample interval. For digital data the result is 1. | 1 |

**Suffix:**
<m> 0..15
Number of the digital channel

**Usage:** Query only

#### DIGital<m>:DATA[:VALues]?

Returns the data of the indicated digital channel for transmission from the instrument to the controlling computer. The data can be used in MATLAB, for example.

**Suffix:**
<m> 0..15
Selects the digital channel.

**Return values:**
<Data> List of data (0 and 1)

**Usage:** Query only

---

**BUS<m>:PARallel:DATA:FORMat** <DataFormat>

Sets the number format for decoded data values on the display for the indicated parallel bus.

It sets also the number representation for remote data transfer with `BUS<m>:PARallel:DATA[:VALues]?`.

**Suffix:**
<m> 1..4

**Parameters:**
<DataFormat> HEX | OCT | BIN | ASCII | ASCii | SIGN | USIG
*RST: HEX

**Manual operation:** See "Data format" on page 942

---

**BUS<m>:PARallel:DATA:HEADer?**

Returns the header data of the indicated bus.

For a detailed description, see `DIGital<m>:DATA:HEADer?`.

**Suffix:**
<m> 1..4
Selects the parallel bus.

**Usage:** Query only

**Firmware/software:** V 2.40

---

**BUS<m>:PARallel:DATA[:VALues]?**

Returns the data of the indicated parallel bus.

Requirements:

- `BUS<m>:PARallel:STATe` is set to `ON`.
- `BUS<m>:PARallel:DISPlay:SHBU` is set to `ON`.
- Number format is set with `BUS<m>:PARallel:DATA:FORMat`.

**Suffix:**
<m> 1..4
Selects the parallel bus.

**Return values:**
<Data> List of values according to the format setting.

**Example:**
```
BUS:PAR:STAT ON
BUS:PAR:DISP:SHBU ON
BUS:PAR:DISP:BTYP COMB
BUS:PAR:DATA:FORMat HEX
BUS:PAR:DATA:VAL?
```

**Usage:** Query only

# 24.19 Waveform generator (option R&S RTP-B6)

## 24.19.1 Waveform generator setup

### 24.19.1.1 General

WGENerator<m>:ACOPy....2334
WGENerator<m>[:ENABle]....2334
WGENerator<m>:SOURce....2334
WGENerator<m>:PRESet....2335

**WGENerator<m>:ACOPy**

Copies all settings from Gen1/Gen2 and applies them to Gen2/Gen1.

**Suffix:**
<m> 1..2

**Usage:** Event

**Manual operation:** See "Copy settings from Gen1/Gen2" on page 968

**WGENerator<m>[:ENABle]** <State>

Enables the waveform generator and outputs the waveform.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF

**Manual operation:** See "Enable" on page 967

**WGENerator<m>:SOURce** <OperationMode>

Sets the operation mode for the the waveform generator.

**Suffix:**
<m> 1..2

**Parameters:**
<OperationMode> FUNCgen | MODulation | SWEep | ARBGenerator
*RST: FUNCgen

**Manual operation:** See "Operation mode" on page 967

---

**WGENerator<m>:PRESet**

Sets the parameters of the waveform generator to their default values.

**Suffix:**
<m> 1..2

**Usage:** Event

**Manual operation:** See "Preset Gen1/Gen2/Patt Gen" on page 968

### 24.19.1.2 Function generator

WGENerator<m>:FUNCtion[:SELect] ..... 2335
WGENerator<m>:FREQuency ..... 2335
WGENerator<m>:PERiod ..... 2336
WGENerator<m>:FUNCtion:PULSe[:WIDTh] ..... 2336
WGENerator<m>:FUNCtion:RAMP[:SYMMetry] ..... 2336
WGENerator<m>:FUNCtion:SQUare:DCYCle ..... 2337

---

**WGENerator<m>:FUNCtion[:SELect]** <FunctionType>

Sets the type of waveform to be generated for the function generator.

**Suffix:**
<m> 1..2

**Parameters:**
<FunctionType> SINusoid | SQUare | RAMP | DC | PULSe | SINC | CARDiac | GAUSs | LORNtz | EXPRise | EXPFall
SINC: Cardial sine
*RST: SINusoid

**Manual operation:** See "Function type" on page 968

---

**WGENerator<m>:FREQuency** <Frequency>

Sets the frequency of the waveform. The available frequency range depends on the selected function type, see Table 15-1.

**Suffix:**
<m> 1..2

**Parameters:**

<Frequency>
Range: 1E-3 to 100E+6
Increment: 1
*RST: 1E+6
Default unit: Hz

**Manual operation:** See "Frequency" on page 972

---

**WGENerator<m>:PERiod** <Period>

Sets the period of the waveform. The available period range depends on the selected function type.

**Suffix:**

<m> 1..2

**Parameters:**

<Period>
Range: 8E-9 to 1000
Increment: 1
*RST: 1E-6
Default unit: s

**Manual operation:** See "Period" on page 973

---

**WGENerator<m>:FUNCtion:PULSe[:WIDTh]** <PulseWidth>

Sets the pulse duration for a pulse waveform.

**Suffix:**

<m> 1..2

**Parameters:**

<PulseWidth>
Range: 16.5E-9 to 1000, actual range depends on the selected frequency
Increment: 1
*RST: 500E-9
Default unit: s

**Manual operation:** See "Pulse width" on page 973

---

**WGENerator<m>:FUNCtion:RAMP[:SYMMetry]** <RampSymmetry>

Sets the symmetry for a ramp waveform.

**Suffix:**

<m> 1..2

**Parameters:**

<RampSymmetry>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Manual operation:** See "Symmetry" on page 973

---

**WGENerator<m>:FUNCtion:SQUare:DCYCle** <SquareDutyCycle>

Sets the duty cycle for the generated square waveform.

**Suffix:**
<m> 1..2

**Parameters:**
<SquareDutyCycle> Range: 0.01 to 99.99 at minimum frequency, actual range depends on the selected frequency
Increment: 1
*RST: 50
Default unit: %

**Manual operation:** See "Duty cycle" on page 973

### 24.19.1.3 Modulation

WGENerator<m>:MODulation:TYPE....2337
WGENerator<m>:MODulation:AM:DCYCle....2338
WGENerator<m>:MODulation:AM:DEPTh....2338
WGENerator<m>:MODulation:AM:FREQuency....2338
WGENerator<m>:MODulation:AM:SYMMetry....2339
WGENerator<m>:MODulation:CARRier:FREQuency....2339
WGENerator<m>:MODulation:CARRier:PERiod....2339
WGENerator<m>:MODulation:AM[:FUNCtion]....2339
WGENerator<m>:MODulation:FM:DCYCle....2340
WGENerator<m>:MODulation:FM:DEViation....2340
WGENerator<m>:MODulation:FM:FREQuency....2340
WGENerator<m>:MODulation:FM:SYMMetry....2341
WGENerator<m>:MODulation:FM[:FUNCtion]....2341
WGENerator<m>:MODulation:FSK:FONE....2341
WGENerator<m>:MODulation:FSK:FTWO....2341
WGENerator<m>:MODulation:FSK[:RATE]....2342
WGENerator<m>:MODulation:PWM:DCYCle....2342
WGENerator<m>:MODulation:PWM:DEPTh....2342
WGENerator<m>:MODulation:PWM:FREQuency....2343
WGENerator<m>:MODulation:PWM:SYMMetry....2343
WGENerator<m>:MODulation:PWM[:FUNCtion]....2343

---

**WGENerator<m>:MODulation:TYPE** <ModulationType>

Sets the modulation type.

**Suffix:**
<m> 1..2

**Parameters:**
<ModulationType> AM | FM | PWM | FSK
*RST: AM

**Manual operation:** See "Modulation type" on page 974

---

**WGENerator<m>:MODulation:AM:DCYCle** <SquareDutyCycle>

Sets the duty cycle of a square waveform for amplitude modulation, if `WGENerator<m>:MODulation:TYPE` is set to `AM`.

**Suffix:**
<m> 1..2

**Parameters:**
<SquareDutyCycle> Range: 10 to 90
Increment: 1
*RST: 50
Default unit: %

**Manual operation:** See "Duty cycle" on page 976

---

**WGENerator<m>:MODulation:AM:DEPTh** <Depth>

Sets the amplitude modulation depth.

**Suffix:**
<m> 1..2

**Parameters:**
<Depth> Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Manual operation:** See "Modulation depth" on page 975

---

**WGENerator<m>:MODulation:AM:FREQuency** <Frequency>

Sets the frequency of the modulation waveform for amplitude modulation, if `WGENerator<m>:MODulation:TYPE` is set to `AM`.

**Suffix:**
<m> 1..2

**Parameters:**
<Frequency> Range: 1E-3 to 1E+6
Increment: 1
*RST: 1000
Default unit: Hz

**Manual operation:** See "Modulation freq" on page 975

**WGENerator<m>:MODulation:AM:SYMMetry** <RampSymmetry>

Sets the symmetry, the percentage of time the ramp modulation waveform is rising, for amplitude modulation, if `WGENerator<m>:MODulation:TYPE` is set to `AM`.

**Suffix:**
<m> 1..2

**Parameters:**
<RampSymmetry>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Manual operation:** See "Symmetry" on page 976

**WGENerator<m>:MODulation:CARRier:FREQuency** <CarrierFreq>

Sets the frequency of the carrier signal for a modulation waveform.

**Suffix:**
<m> 1..2

**Parameters:**
<CarrierFreq>
Range: 1E-3 to 100E+6
Increment: 10
*RST: 1E+6
Default unit: Hz

**Manual operation:** See "Carrier frequency" on page 974

**WGENerator<m>:MODulation:CARRier:PERiod** <CarrierPeriod>

Sets the period of the carrier signal for a modulation waveform.

**Suffix:**
<m> 1..2

**Parameters:**
<CarrierPeriod>
Range: 8E-9 to 1000
Increment: 10
*RST: 1E-6
Default unit: s

**Manual operation:** See "Carrier period" on page 975

**WGENerator<m>:MODulation:AM[:FUNCtion]** <SignalType>

Sets the type of the modulation signal for the amplitude modulation, if `WGENerator<m>:MODulation:TYPE` is set to `AM`.

**Suffix:**
<m> 1..2

**Parameters:**
<SignalType> SINusoid | SQUare | RAMP
*RST: SINusoid

**Manual operation:** See "Modulation signal" on page 975

---

**WGENerator<m>:MODulation:FM:DCYCle** <SquareDutyCycle>

Sets the duty cycle of a square waveform for frequency modulation, if `WGENerator<m>:MODulation:TYPE` is set to `FM`.

**Suffix:**
<m> 1..2

**Parameters:**
<SquareDutyCycle> Range: 10 to 90
Increment: 1
*RST: 50
Default unit: %

**Manual operation:** See "Duty cycle" on page 979

---

**WGENerator<m>:MODulation:FM:DEViation** <Deviation>

Sets the frequency modulation deviation.

**Suffix:**
<m> 1..2

**Parameters:**
<Deviation> Range: 1E-3 to 10E+6
Increment: 1
*RST: 1000
Default unit: Hz

**Manual operation:** See "Freq deviation" on page 978

---

**WGENerator<m>:MODulation:FM:FREQuency** <Frequency>

Sets the frequency of the modulation waveform for frequency modulation, if `WGENerator<m>:MODulation:TYPE` is set to `FM`.

**Suffix:**
<m> 1..2

**Parameters:**
<Frequency> Range: 1E-3 to 1E+6
Increment: 1
*RST: 1000
Default unit: Hz

**Manual operation:** See "Modulation freq" on page 978

**WGENerator<m>:MODulation:FM:SYMMetry** <RampSymmetry>

Sets the symmetry, the percentage of time the ramp modulation waveform is rising, for frequency modulation, if `WGENerator<m>:MODulation:TYPE` is set to `FM`.

**Suffix:**
<m> 1..2

**Parameters:**
<RampSymmetry>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Manual operation:** See "Symmetry" on page 978

**WGENerator<m>:MODulation:FM[:FUNCtion]** <SignalType>

Sets the type of the modulation signal for the frequency modulation, if `WGENerator<m>:MODulation:TYPE` is set to `FM`.

**Suffix:**
<m> 1..2

**Parameters:**
<SignalType> SINusoid | SQUare | RAMP
*RST: SINusoid

**Manual operation:** See "Modulation signal" on page 978

**WGENerator<m>:MODulation:FSK:FONE** <Frequency1>

Sets the frequency of the carrier waveform, if `WGENerator<m>:MODulation:TYPE` is set to `FSK`.

**Suffix:**
<m> 1..2

**Parameters:**
<Frequency1>
Range: 1E-3 to 100E+6
Increment: 1
*RST: 1E+6
Default unit: Hz

**Manual operation:** See "Frequency 1" on page 979

**WGENerator<m>:MODulation:FSK:FTWO** <Frequency2>

Sets the frequency of the modulated waveform, if `WGENerator<m>:MODulation:TYPE` is set to `FSK`.

**Suffix:**
<m> 1..2

**Parameters:**

<Frequency2>
Range: 1E-3 to 100E+6
Increment: 1
*RST: 1000
Default unit: Hz

**Manual operation:** See "Frequency 2" on page 979

---

**WGENerator<m>:MODulation:FSK[:RATE]** <Rate>

Sets the hop rate, the time before a switch from the carrier frequency set with `WGENerator<m>:MODulation:FSK:FONE` and the modulation frequency set with `WGENerator<m>:MODulation:FSK:FTWO`.

**Suffix:**

<m> 1..2

**Parameters:**

<Rate>
Range: 1E-3 to 1E+6
Increment: 1
*RST: 1000
Default unit: Hz

**Manual operation:** See "Hop rate" on page 979

---

**WGENerator<m>:MODulation:PWM:DCYCle** <SquareDutyCycle>

Sets the duty cycle of a square waveform for pulse width modulation, if `WGENerator<m>:MODulation:TYPE` is set to `PWM`.

**Suffix:**

<m> 1..2

**Parameters:**

<SquareDutyCycle>
Range: 10 to 90
Increment: 1
*RST: 50
Default unit: %

**Manual operation:** See "Duty cycle" on page 977

---

**WGENerator<m>:MODulation:PWM:DEPTh** <Depth>

Sets the modulation depth for pulse width modulation.

**Suffix:**

<m> 1..2

**Parameters:**

<Depth>
Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Manual operation:** See "Modulation depth" on page 977

---

**WGENerator<m>:MODulation:PWM:FREQuency** <Frequency>

Sets the frequency of the modulation waveform for pulse width modulation, if `WGENerator<m>:MODulation:TYPE` is set to `PWM`.

**Suffix:**
<m> 1..2

**Parameters:**
<Frequency> Range: 1E-3 to 1E+6
Increment: 1
*RST: 1000
Default unit: Hz

**Manual operation:** See "Modulation freq" on page 977

---

**WGENerator<m>:MODulation:PWM:SYMMetry** <RampSymmetry>

Sets the symmetry, the percentage of time the ramp modulation waveform is rising, for pulse width modulation, if `WGENerator<m>:MODulation:TYPE` is set to `PWM`.

**Suffix:**
<m> 1..2

**Parameters:**
<RampSymmetry> Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Manual operation:** See "Symmetry" on page 977

---

**WGENerator<m>:MODulation:PWM[:FUNCtion]** <SignalType>

Sets the type of the modulation signal for the pulse width modulation, if `WGENerator<m>:MODulation:TYPE` is set to `PWM`.

**Suffix:**
<m> 1..2

**Parameters:**
<SignalType> SINusoid | SQUare | RAMP
*RST: SINusoid

**Manual operation:** See "Modulation signal" on page 976

### 24.19.1.4 Sweep

WGENerator<m>:SWEep:FSTart....2344
WGENerator<m>:SWEep:TIME....2344
WGENerator<m>:SWEep[:FEND]....2344

**WGENerator<m>:SWEep:FSTart** <StartFrequency>

Sets the start frequency of the sweep range.

**Suffix:**
<m> 1..2

**Parameters:**
<StartFrequency> Range: 1E-3 to 100E+6
Increment: 1
*RST: 1000
Default unit: Hz

**Manual operation:** See "Start frequency" on page 980

**WGENerator<m>:SWEep:TIME** <Time>

Sets the duration of the sweep.

**Suffix:**
<m> 1..2

**Parameters:**
<Time> Range: 1E-3 to 500
Increment: 1
*RST: 1E-3
Default unit: s

**Manual operation:** See "Sweep time" on page 980

**WGENerator<m>:SWEep[:FEND]** <StopFrequency>

Sets the stop frequency of the sweep range.

**Suffix:**
<m> 1..2

**Parameters:**
<StopFrequency> Range: 1E-3 to 100E+6
Increment: 1
*RST: 1E+6
Default unit: Hz

**Manual operation:** See "Stop freuqency" on page 980

### 24.19.1.5 ARB

WGENerator<m>:ARBGen:COPY ..... 2345
WGENerator<m>:ARBGen:MULTichannel:IMPort ..... 2345
WGENerator<m>:ARBGen:MULTichannel:NAME ..... 2345
WGENerator<m>:ARBGen:MULTichannel:OPEN ..... 2346
WGENerator<m>:ARBGen:NAME ..... 2346
WGENerator<m>:ARBGen:OPEN ..... 2346
WGENerator<m>:ARBGen:RUNMode ..... 2347
WGENerator<m>:ARBGen:SAMPles? ..... 2347
WGENerator<m>:ARBGen:SELect ..... 2347
WGENerator<m>:ARBGen:SRATe ..... 2348
WGENerator<m>:ARBGen[:SOURce] ..... 2348
WGENerator<m>:ARBGen:RUNSingle ..... 2348

**WGENerator<m>:ARBGen:COPY**

Loads the waveform from the selected signal source (`WGENerator<m>:ARBGen:SELect`).

**Suffix:**
<m> 1..2

**Usage:** Event

**Manual operation:** See "Load" on page 984

**WGENerator<m>:ARBGen:MULTichannel:IMPort** <SavedWfmSrc>

Assigns a waveform from the multichannel file to the arbitrary waveform generator.

**Suffix:**
<m> 1..2

**Parameters:**
<SavedWfmSrc> WFM1 | WFM2 | WFM3 | WFM4
*RST: WFM1

**Example:**
```
WGENerator1:ARBGen:MULTichannel:NAME
ArbMultichannelCurve_2017-02-16_01.bin
WGENerator1:ARBGen:MULTichannel:IMPort WF1
WGENerator1:ARBGen:MULTichannel:OPEN
```

**Manual operation:** See "Select waveform" on page 984

**WGENerator<m>:ARBGen:MULTichannel:NAME** <FilePath>

Defines the path and the multichannel arbitrary waveform file to be imported. If not path is given, the default path
`C:\Users\Public\Public Documents\Rohde-Schwarz\RTx\RefWaveforms`
is used.

**Suffix:**
<m> 1..2

**Parameters:**
<FilePath>

**Example:** See `WGENerator<m>:ARBGen:MULTichannel:IMPort` on page 2345.

**Manual operation:** See "Select waveform" on page 984

---

**WGENerator<m>:ARBGen:MULTichannel:OPEN**

Loads the arbitrary waveform.

**Suffix:**
<m> 1..2

**Example:** See `WGENerator<m>:ARBGen:MULTichannel:IMPort` on page 2345.

**Usage:** Event

**Manual operation:** See "Select waveform" on page 984

---

**WGENerator<m>:ARBGen:NAME** <FilePath>

Sets the file path and the file for an arbitrary waveform, if `WGENerator<m>:ARBGen[:SOURce]` on page 2348 is set to `ARBitrary`.

If not path is given, the default path `C:\Users\Public\Public Documents\Rohde-Schwarz\RTx\RefWaveforms` is used.

This command is only relevant for files with one waveform defined. For multichannel arbitrary waveform files, see `WGENerator<m>:ARBGen:MULTichannel:OPEN`.

**Suffix:**
<m> 1..2

**Parameters:**
<FilePath>

**Manual operation:** See "Load arbitrary waveform" on page 983

---

**WGENerator<m>:ARBGen:OPEN**

Loads the arbitrary waveform, that is selected with the `WGENerator<m>:ARBGen:NAME` command.

This command is only relevant for files with one waveform defined. For multichannel arbitrary waveform files, see `WGENerator<m>:ARBGen:MULTichannel:OPEN`.

**Suffix:**
<m> 1..2

**Usage:** Event

**Manual operation:** See "Load arbitrary waveform" on page 983

---

**WGENerator<m>:ARBGen:RUNMode** <RunMode>

Sets the duration for which the signal of the arbitrary generator will be output after the trigger event.

**Suffix:**
<m> 1..2

**Parameters:**
<RunMode> SINGle | CONTinuous
*RST: CONTinuous

**Manual operation:** See "Run mode" on page 982

---

**WGENerator<m>:ARBGen:SAMPles?**

Displays the number of samples for the arbitrary waveform.

**Suffix:**
<m> 1..2

**Return values:**
<NumSamples> Range: 0 to 128000000
Increment: 10
*RST: 0
Default unit: pts

**Usage:** Query only

**Manual operation:** See "Samples" on page 983

---

**WGENerator<m>:ARBGen:SELect** <ScopeSignSrc>

Selects the oscilloscope source, from which the arbitrary signal is loaded, if `WGENerator<m>:ARBGen[:SOURce]` is set to `SCOPe`.

**Suffix:**
<m> 1..2

**Parameters:**
<ScopeSignSrc> NONE | C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | R1 | R2 | R3 | R4 | DIFF1 | DIFF2 | COMMON1 | COMMON2
*RST: NONE

**Manual operation:** See "Signal source" on page 984

**WGENerator<m>:ARBGen:SRATe** <SampleRate>

Sets the sample rate for the arbitrary waveform.

**Suffix:**
<m> 1..2

**Parameters:**
<SampleRate> Range: 1 to 250E+6
Increment: 10
*RST: 1E+6
Default unit: Sa/s

**Manual operation:** See "Sample rate" on page 983

**WGENerator<m>:ARBGen[:SOURce]** <WaveformSource>

Sets the arbitrary waveform source.

**Suffix:**
<m> 1..2

**Parameters:**
<WaveformSource> ARBitrary | SCOPe

**ARBitrary**
Enables you to load an existing arbitrary file.

**SCOPe**
Enables you to load a scope waveform.

*RST: ARBitrary

**Manual operation:** See "Waveform source" on page 982

**WGENerator<m>:ARBGen:RUNSingle**

Executes a single period of the arbitrary signal generator, for `WGENerator<m>:ARBGen:RUNModeSINGle`.

**Suffix:**
<m> 1..2

**Usage:** Event

**Manual operation:** See "Run single" on page 983

## 24.19.2 Pattern generator setup

PGENerator:BITRate....2349
PGENerator:ENABle....2349
PGENerator:FILE:OPEN....2349
PGENerator:FILE[:NAME]....2349
PGENerator:HLEVel....2349

PGENerator:PRESet....2350
PGENerator:RUNMode....2350
PGENerator:RUNSingle....2350

**PGENerator:BITRate** <BitRate>

Sets the number of transmitted bits per second for the pattern generator.

**Parameters:**

<BitRate>
Range: 1 to 40E+6
Increment: 10
*RST: 1E+6
Default unit: bps

**Manual operation:** See "Bit rate" on page 987

**PGENerator:ENABle** <State>

Enables the pattern generator and outputs the waveform.

**Parameters:**

<State>
ON | OFF

**Manual operation:** See "Enable" on page 967

**PGENerator:FILE:OPEN**

Loads the specified pattern file to the instrument.

**Usage:** Event

**Manual operation:** See "Load pattern file" on page 987

**PGENerator:FILE[:NAME]** <FilePath>

Specifies a file path to open a pattern generator file.

**Parameters:**

<FilePath>
String with the name of the file.

**Manual operation:** See "Load pattern file" on page 987

**PGENerator:HLEVel** <HighLevel>

Sets the high level of the pattern generator signal.

**Parameters:**

<HighLevel>
Range: 1.2 to 5
Increment: 0.1
*RST: 1.2
Default unit: V

**PGENerator:PRESet**

Sets the default pattern generator settings.

**Usage:** Event

**Manual operation:** See "Preset Gen1/Gen2/Patt Gen" on page 968

**PGENerator:RUNMode** <RunMode>

Sets the duration for which the signal of the pattern generator will be output after the trigger event.

**Parameters:**
<RunMode> SINGle | CONTinuous
*RST: CONTinuous

**Manual operation:** See "Run mode" on page 987

**PGENerator:RUNSingle**

Executes a single period of the pattern generator, for `PGENerator:RUNModeSINGle`.

**Usage:** Event

**Manual operation:** See "Run single" on page 987

### 24.19.3 Output setup

WGENerator<m>:OUTPut[:LOAD] 2350
WGENerator<m>:VOLTage[:VPP] 2351
WGENerator<m>:VOLTage:DCLevel 2351
WGENerator<m>:VOLTage:HIGH 2351
WGENerator<m>:VOLTage:INVersion 2351
WGENerator<m>:VOLTage:LOW 2352
WGENerator<m>:VOLTage:OFFSet 2352

**WGENerator<m>:OUTPut[:LOAD]** <Load>

Sets the load of the DUT at its connection.

**Suffix:**
<m> 1..2

**Parameters:**
<Load> FIFTy | HIZ
HIZ: high input impedance
*RST: HIZ

**Manual operation:** See "User Load" on page 989

**WGENerator<m>:VOLTage[:VPP]** <Amplitude>

Sets the amplitude, peak to peak voltage, of the output waveform. This is defined as the voltage difference between the maximum (`WGENerator<m>:VOLTage:HIGH`) and the minimum (`WGENerator<m>:VOLTage:LOW`) voltage levels.

The value is set for the currently selected load (`WGENerator<m>:OUTPut[:LOAD]`). If the load is changed, the value of the amplitude is adapted to this new setting.

**Suffix:**
<m> 1..2

**Parameters:**
<Amplitude>
Range: 0.01 to 12
Increment: 0.1
*RST: 1
Default unit: V

**Manual operation:** See "Amplitude" on page 989

**WGENerator<m>:VOLTage:DCLevel** <DCLevel>

Sets the DC level of the generated DC signal.

**Suffix:**
<m> 1..2

**Parameters:**
<DCLevel>
Range: -5.995 to 5.995
Increment: 0.1
*RST: 0
Default unit: V

**Manual operation:** See "DC Level" on page 989

**WGENerator<m>:VOLTage:HIGH** <High>

Sets the high signal level of the output waveform.

**Suffix:**
<m> 1..2

**Parameters:**
<High>
Range: -5.99 to 6
Increment: 0.1
*RST: 0.5
Default unit: V

**Manual operation:** See "High" on page 989

**WGENerator<m>:VOLTage:INVersion** <Inversion>

Enables inversion, which swaps the two poles of the waveform.

**Suffix:**
<m> 1..2

**Parameters:**
<Inversion> ON | OFF
*RST: OFF

**Manual operation:** See "Invert" on page 989

---

**WGENerator<m>:VOLTage:LOW** <Low>

Sets the low signal level of the output waveform.

**Suffix:**
<m> 1..2

**Parameters:**
<Low> Range: -6 to 5.99
Increment: 0.1
*RST: -0.5
Default unit: V

**Manual operation:** See "Low" on page 989

---

**WGENerator<m>:VOLTage:OFFSet** <Offset>

Sets a voltage offset.

**Suffix:**
<m> 1..2

**Parameters:**
<Offset> Range: -5.995 to 5.995
Increment: 0.1
*RST: 0
Default unit: V

**Manual operation:** See "Offset" on page 989

### 24.19.4 Noise setup

WGENerator<m>:MODulation:NDCLevel....2352
WGENerator<m>:MODulation:NLABsolute?....2353
WGENerator<m>:MODulation:NLPCent....2353
WGENerator<m>:MODulation:NOISe....2353

---

**WGENerator<m>:MODulation:NDCLevel** <LevelDC>

Sets the DC noise level.

**Suffix:**
<m> 1..2

**Parameters:**

<LevelDC> Range: 0 to 12
Increment: 0.1
*RST: 0.01
Default unit: V

**Manual operation:** See "Absolute level" on page 990

---

**WGENerator<m>:MODulation:NLABsolute?**

Queries the level of the noise in volts.

**Suffix:**

<m> 1..2

**Return values:**

<LevelAbs> Range: 0 to 12
Increment: 0.1
*RST: 0
Default unit: V

**Usage:** Query only

**Manual operation:** See "Absolute level" on page 990

---

**WGENerator<m>:MODulation:NLPCent** <LevelPct>

Sets the noise level in percentage of the amplitude.

**Suffix:**

<m> 1..2

**Parameters:**

<LevelPct> Range: 0 to 100
Increment: 1
*RST: 0
Default unit: %

**Manual operation:** See "Relative level" on page 990

---

**WGENerator<m>:MODulation:NOISe** <State>

Enables the adding of noise to the waveform.

**Suffix:**

<m> 1..2

**Parameters:**

<State> ON | OFF
*RST: OFF

**Manual operation:** See "Enable" on page 989

### 24.19.5 DC offset alignment

GENerator:ALIGnment:DC:ABORt....2354
GENerator:ALIGnment:DC:RESult:DATE?....2354
GENerator:ALIGnment:DC:RESult:TIME?....2354
GENerator:ALIGnment:DC:RESult[:STATe]?....2354
GENerator:ALIGnment:DC[:STARt]....2354

**GENerator:ALIGnment:DC:ABORt**

Aborts a DC offset alignment.

**Usage:** Event

**GENerator:ALIGnment:DC:RESult:DATE?**

Queries the date of the last performed DC offset alignment.

**Return values:**
<Date>

**Usage:** Query only

**Manual operation:** See "Date" on page 990

**GENerator:ALIGnment:DC:RESult:TIME?**

Queries the time of the last performed DC offset alignment.

**Return values:**
<Time>

**Usage:** Query only

**Manual operation:** See "Time" on page 991

**GENerator:ALIGnment:DC:RESult[:STATe]?**

Queries the result of a DC offset alignment.

**Return values:**
<State> PASSed | FAILed | NOALigndata
*RST: NOALigndata

**Usage:** Query only

**Manual operation:** See "Overall alignment state" on page 991

**GENerator:ALIGnment:DC[:STARt]**

Starts a DC offset alignment.

**Usage:** Event

**Manual operation:** See "Start Alignement" on page 990

### 24.19.6 Coupling & sync

WGENerator<m>:COUPling:STATe....2355
WGENerator<m>:COUPling:AMPLitude....2355
WGENerator<m>:COUPling:PHASeshift....2355
WGENerator<m>:COUPling[:FREQuency]....2356
WGENerator<m>:COUPling:NOISe....2356
WGENerator<m>:COUPling:AWForm....2356
GENerator:SYNC[:COMBination]....2356

**WGENerator<m>:COUPling:STATe** <CplEnabSt>

Enables the coupling of the generators, with the selected set of parameters: amplitude, frequency and noise.

**Suffix:**
<m> 1..2

**Parameters:**
<CplEnabSt> ON | OFF
*RST: OFF

**Manual operation:** See "Generator enable" on page 991

**WGENerator<m>:COUPling:AMPLitude** <CoupleAmplitude>

Enables the coupling of all amplitude parameters of the generators.

**Suffix:**
<m> 1..2
Specifies the master generator.

**Parameters:**
<CoupleAmplitude> ON | OFF
*RST: OFF

**Manual operation:** See "Frequency parameters" on page 992

**WGENerator<m>:COUPling:PHASeshift** <PhaseShift>

Sets the phase shift between the waveform of Gen1 and Gen2 when the frequency parameters of the two waveforms are coupled.

**Suffix:**
<m> 1..2
Specifies the master generator.

**Parameters:**

<PhaseShift>
Range: -180 to 180
Increment: 1
*RST: 0
Default unit: °

**Manual operation:** See "Phase shift" on page 992

---

**WGENerator<m>:COUPling[:FREQuency]** <CoupleFrequency>

Enables the coupling of all frequency parameters of the generators.

**Suffix:**

<m>
1..2
Specifies the master generator.

**Parameters:**

<CoupleFrequency>
ON | OFF
*RST: OFF

**Manual operation:** See "Amplitude parameters" on page 992

---

**WGENerator<m>:COUPling:NOISe** <CoupleNoise>

Couples all noise parameters of the generators.

**Suffix:**

<m>
1..2

**Parameters:**

<CoupleNoise>
ON | OFF
*RST: OFF

**Manual operation:** See "Noise parameters" on page 992

---

**WGENerator<m>:COUPling:AWForm** <CplArbitraryWfm>

Enables the coupling of the arbitrary waveform of the generators.

**Suffix:**

<m>
1..2

**Parameters:**

<CplArbitraryWfm>
ON | OFF
*RST: OFF

---

**GENerator:SYNC[:COMBination]** <Combination>

Sets which signals generated from the waveform generator are synchronized.

**Parameters:**

<Combination> NONE | GEN12 | G1PG | G2PG | G12PG

GEN12: Gen1 and Gen 2
GEN1PG/GEN2PG: Gen1/Gen 2 and Patt Gen
GEN12PG: Gen1,Gen 2 and Patt Gen

*RST: NONE

**Manual operation:** See "Sync start" on page 993

# 24.20 Pulse source and TDR/TDT analysis (options R&S RTP-B7/K130)

## 24.20.1 Pulse source (option R&S RTP-B7)

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

PSRC[:STATe] ..... 2357
PSRC:OUTPutlow ..... 2358
PSRC:REPRate ..... 2358
PSRC:DUTYcycle ..... 2358
PSRC:CLOCkmode ..... 2358
PSRC:RST ..... 2359

---

**PSRC[:STATe]** <State>

Switches the pulse output on and off.

**Parameters:**

<State> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 998

**PSRC:OUTPutlow** <OutputLowLevel>

Sets the low level of the output pulse.

**Parameters:**

<OutputLowLevel>
Range: -0.2 to -0.05
Increment: 0.01
*RST: -0.05
Default unit: V

**Usage:** Asynchronous command

**Manual operation:** See "Output level" on page 999

**PSRC:REPRate** <RepetitionRate>

Sets the pulse frequency, the repetition rate of the pulse.

**Parameters:**

<RepetitionRate>
Available values depend on the selected clock mode:
Locked: 5/10/20/50/100/200/500 Hz, 1/5/10/25/50/100/250 MHz
Free running: 5/10/20/50/100/200/500 Hz, 1/5/10/25/50 MHz
*RST: 1E+6
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Repetition rate" on page 999

**PSRC:DUTYcycle** <DutyCycle>

Sets the duty cycle of the pulse, which is the ratio of the positive pulse width to the period of the signal.

**Parameters:**

<DutyCycle>
For repetition rates > 5 MHz, the value is fixed at 50 %.
Range: 10 to 90
Increment: 10
*RST: 50
Default unit: %

**Usage:** Asynchronous command

**Manual operation:** See "Duty cycle" on page 999

**PSRC:CLOCkmode** <ClockMode>

Sets the dependency of the pulse clock on the instrument's reference clock.

**Parameters:**

<ClockMode>
LOCKed | FREerunning

**LOCKed**
The pulse source is locked to the reference clock of the instrument.

**FREerunning**
The clock of the pulse source is independent. Deviations of the system do not affect the pulse clock, and deterministic conditions are avoided.

*RST: FREerunning

**Usage:** Asynchronous command

**Manual operation:** See "Clock mode" on page 999

---

**PSRC:RST**

Resets the pulse source to the default state.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Preset" on page 999

## 24.20.2 TDR/TDT analysis

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

### 24.20.2.1 Calibration

TDRT:DEFault ..... 2360
TDRT:CALibration:OPEN:NAME ..... 2360
TDRT:CALibration:OPEN[:EXECute] ..... 2360
TDRT:TYPE ..... 2360
TDRT:CONFig ..... 2360
TDRT:PULSe:OUT ..... 2361
TDRT:PULSe:REFerence ..... 2361
TDRT:PULSe:INVout ..... 2361
TDRT:PULSe:INVReference ..... 2361

TDRT:SLENgth....2362
TDRT:CALibration:SAVE:NAME....2362
TDRT:CALibration:SAVE[:EXECute]....2362

**TDRT:DEFault**

Starts the measurement with default correction values instead of calibrating the setup. Default values are rough values without time reference.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Measure with default values" on page 1005

**TDRT:CALibration:OPEN:NAME** <Path>

Sets the file name, file format and path of the TDR/TDT calibration file to be opened. The file format is always `.xml`.

**Parameters:**
<Path> String with path and file name with extension `.xml`

**Usage:** Asynchronous command

**Manual operation:** See "Open" on page 1005

**TDRT:CALibration:OPEN[:EXECute]**

Opens and loads the TDR/TDT calibration from the file specified with `TDRT:CALibration:OPEN:NAME`.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Open" on page 1005

**TDRT:TYPE** <Type>

Selects the analysis type: TDR (time domain reflectometry), TDT (time domain transmission), or TDR/TDT (both TDR and TDT).

**Parameters:**
<Type> TDR | TDRTdt | TDT
*RST: TDR

**Usage:** Asynchronous command

**Manual operation:** See "Measurement type" on page 1004

**TDRT:CONFig** <Configuration>

Selects the connection: single-ended or differential.

**Parameters:**
<Configuration> SINGleended | DIFFerential
*RST: SINGleended

**Usage:** Asynchronous command

**Manual operation:** See "Configuration" on page 1004

---

**TDRT:PULSe:OUT** <Channel>

Sets the channel to which the "Out" connector of the pulse source is connected via calibration standard or DUT.

**Parameters:**
<Channel> C1W1 | C2W1 | C3W1 | C4W1
*RST: C2W1

**Usage:** Asynchronous command

**Manual operation:** See "Channel" on page 1006

---

**TDRT:PULSe:REFerence** <Channel>

Sets the channel to which the "Ref" connector of the pulse source is connected.

**Parameters:**
<Channel> C1W1 | C2W1 | C3W1 | C4W1
*RST: C1W1

**Usage:** Asynchronous command

**Manual operation:** See "Channel" on page 1006

---

**TDRT:PULSe:INVout** <Channel>

Sets the channel to which the $\overline{\text{Out}}$ connector of the pulse source is connected via calibration standard or DUT. Only required for differential measurements.

**Parameters:**
<Channel> C1W1 | C2W1 | C3W1 | C4W1
*RST: C3W1

**Usage:** Asynchronous command

**Manual operation:** See "Channel" on page 1006

---

**TDRT:PULSe:INVReference** <Channel>

Sets the channel to which the $\overline{\text{Ref}}$ connector of the pulse source is connected. Only required for differential measurements.

**Parameters:**
<Channel> C1W1 | C2W1 | C3W1 | C4W1
*RST: C4W1

**Usage:** Asynchronous command

**Manual operation:** See "Channel" on page 1006

---

**TDRT:SLENgth** <ShortLength>

Sets the electrical length of a user-defined short.

The reset value is the electrical length of the short hat is delivered with the pulse source R&S RTP-B7.

**Parameters:**
<ShortLength> Range: 0 to 2E-9
Increment: 1E-12
*RST: 26.5E-12
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Short type" on page 1008

---

**TDRT:CALibration:SAVE:NAME** <Path>

Sets the file name, file format and path of the TDR/TDT calibration file to be saved. The file format is always `.xml`.

**Parameters:**
<Path> String with path and file name with extension `.xml`

**Usage:** Asynchronous command

**Manual operation:** See "Save calibration" on page 1010

---

**TDRT:CALibration:SAVE[:EXECute]**

Saves the TDR/TDT calibration results to the file that is specified with `TDRT:CALibration:SAVE:NAME`.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Save calibration" on page 1010

### 24.20.2.2 Measurements

TDRT:STATe....2363
TDRT:CALibration:USECaldata....2363
TDRT:BW....2363
TDRT:DOMain....2364

TDRT:PERMittivity....2364
TDRT:SIGNal<m>STATe....2364
TDRT:SIGNal<m>:SCALe....2364
TDRT:SIGNal<m>:OFFSet....2365
TDRT:SIGNal<m>:POSition....2365
TDRT:SIGNal<m>:DATA:HEADer?....2365
TDRT:SIGNal<m>:DATA:VALues?....2366

**TDRT:STATe** <State>

Activates or deactivates the TDR/TDT measurements.

**Parameters:**

<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 1011

**TDRT:CALibration:USECaldata** <UseCalibrationDat>

Disables or enables the frequency response calibration. Without calibration, the measured signals are only rescaled, and you can see the impact of the frequency response calibration on the time domain waveform.

The command is not relevant when measuring with default values.

**Parameters:**

<UseCalibrationDat> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Use frequency response calibration" on page 1011

**TDRT:BW** <Bandwidth>

Sets the bandwidth of the DUT.

**Parameters:**

<Bandwidth> Range: 2E+9 to 16E+9
Increment: 10
*RST: 4E+9
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Bandwidth" on page 1011

**TDRT:DOMain** <OutputDomain>

Sets the domain of the x-axis: time or distance (in m or inch). If you select distance, enter also the permittivity with `TDRT:PERMittivity`.

**Parameters:**

<OutputDomain> TIME | MDIStance | INDistance
MDIStance = distance in m
IDIStance = distance in inch
*RST: TIME

**Usage:** Asynchronous command

**Manual operation:** See "Domain, ε_r" on page 1012

**TDRT:PERMittivity** <Permittivity>

Sets the permittivity "ε_r" of the used DUT.

**Parameters:**

<Permittivity> Range: 1 to 100
Increment: 0.1
*RST: 3.3

**Usage:** Asynchronous command

**Manual operation:** See "Domain, ε_r" on page 1012

**TDRT:SIGNal<m>STATe** <State>

Enables the indicated TDR/TDT signal.

**Suffix:**

<m> 1..3
1 = impedance Z (result of TDR measurement)
2 = reflection coefficient ρ (result of TDR measurement)
3 = step response h (result of TDT measurement)

**Parameters:**

<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 1012

**TDRT:SIGNal<m>:SCALe** <VerticalScale>

Set the vertical scale of the indicated TDR/TDT signal.

**Suffix:**

<m> 1..3
See `TDRT:SIGNal<m>STATe` on page 2364.

**Parameters:**

<VerticalScale>
Range: 1E-12 to 100E+12
Increment: 1E-3
*RST: 0.05
Default unit: /div

**Usage:** Asynchronous command

**Manual operation:** See "Vertical scale" on page 1012

---

**TDRT:SIGNal<m>:OFFSet** <Offset>

Set the offset of the indicated TDR/TDT signal.

**Suffix:**

<m>
1..3
See TDRT:SIGNal<m>STATe on page 2364.

**Parameters:**

<Offset>
Range: 1E-12 to 100E+12
Increment: 0.01
*RST: 0

**Usage:** Asynchronous command

**Manual operation:** See "Offset" on page 1012

---

**TDRT:SIGNal<m>:POSition** <Position>

Set the vertical position of the indicated TDR/TDT signal.

**Suffix:**

<m>
1..3
See TDRT:SIGNal<m>STATe on page 2364.

**Parameters:**

<Position>
Range: 1E-12 to 100E+12
Increment: 0.02
*RST: 0
Default unit: div

**Usage:** Asynchronous command

**Manual operation:** See "Offset" on page 1012

---

**TDRT:SIGNal<m>:DATA:HEADer?**

Returns the header of channel waveform data.

*Table 24-18: Header data*

| Position | Meaning | Example |
|---|---|---|
| 1 | XStart in s | -9.477E-008 = - 94,77 ns |
| 2 | XStop in s | 9.477E-008 = 94,77 ns |

| Position | Meaning | Example |
|---|---|---|
| 3 | Record length of one waveform | 200000 |
| 4 | Number of values per sample interval, is 1 | 1 |

**Suffix:**
<m> 1..3
1 = impedance Z (result of TDR measurement)
2 = reflection coefficient ρ (result of TDR measurement)
3 = pulse response h (result of TDT measurement)

**Usage:** Query only

---

**TDRT:SIGNal<m>:DATA:VALues?** [<Offset>], [<Length>]

Returns the data of the waveform points for transmission from the instrument to the controlling computer. The data can be used in MATLAB, for example.

Without parameters, the complete waveform is retrieved. Using the offset and length parameters, data can be retrieved in smaller portions, which makes the command faster.

To set the export format, use `FORMat[:DATA]`.

**Suffix:**
<m> 1..3
1 = impedance Z (result of TDR measurement)
2 = reflection coefficient ρ (result of TDR measurement)
3 = pulse response h (result of TDT measurement)

**Query parameters:**
[<Offset>] Number of offset waveform points.
Range: 0 to m. Limit: n + m >= record length

**Return values:**
[<Length>] Number of waveform points to be retrieved.
Range: 1 to n. Limit: n + m >= record length

**Usage:** Query only

## 24.21 I/Q software interface (option R&S RTP-K11)

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

- I/Q sampling settings....2367
- I/Q data output....2372

## 24.21.1 I/Q sampling settings

To switch on the input channels, use `CHANnel<m>:STATe`.

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

IQ:STATe....2367
IQ:INSRate....2368
CHANnel<m>:IQ:INPType....2368
CHANnel<m>:IQ:INPMode....2368
IQ:RBWidth....2369
IQ:BWIDth?....2369
IQ:SRATe....2369
IQ:RLENgth....2370
CHANnel<m>:IQ:CFRequency....2370
CHANnel<m>:IQ:SBRF....2370
CHANnel<m>:IQ:SBIF....2371
CHANnel<m>:IQ:WARNing[:STATe]?....2371
CHANnel<m>:IQ:WARNing:MESSage?....2372

---

**IQ:STATe** <IQMode>

Activates the I/Q mode of the instrument.

**Parameters:**

<IQMode> ON | OFF

ON: I/Q mode
OFF: normal oscilloscope mode

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 1016

---

**IQ:INSRate** <InputSampleRate>

Shows the sample rate of capturing the input data in I/Q mode: 20 Gsample/s.

If you use R&S RTP134B or R&S RTP164B instruments, you can change the sample rate to 40 Gsample/s. See also: "Input sample rate" on page 1016.

**Parameters:**
<InputSampleRate> SR20G | SR40G

**Example:** Query the input sample rate:

```
IQ:INSR?
<-- SR20G
```

Only at R&S RTP134B andR&S RTP164B: change the input sample rate:

```
IQ:INSR SR40G
```

**Usage:** Asynchronous command

**Manual operation:** See "Input sample rate" on page 1016

---

**CHANnel<m>:IQ:INPType** <IQInputType>

Sets the type of the input signal.

**Suffix:**
<m> 1..4

**Parameters:**
<IQInputType> REAL | COMPlex

**REAL**
Real RF signal. One real RF signal requires one input channel, thus up to four real signals can be recorded in parallel.

**COMPlex**
Complex I/Q signal in baseband or low IF range. One complex input signal requires two input channels. The In-Phase component must be connected to channel 1 or 3, and the Quadrature component must be connected to channel 2 or 4. Thus up to two complex input signals can be recorded in parallel.

*RST: REAL

**Usage:** Asynchronous command

**Manual operation:** See "Input type, Input mode" on page 1016

---

**CHANnel<m>:IQ:INPMode** <IQInputMode>

Selects the frequency band of a complex input signal.

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<IQInputMode> BASeband | RFIF
Baseband or intermediate frequency (RFIF)
*RST: BASeband

**Usage:** Asynchronous command

**Manual operation:** See "Input type, Input mode" on page 1016

---

**IQ:RBWidth** <RelBw>

Sets the bandwidth factor to define the filter bandwidth.

*Filter BW = Relative BW * Sample rate*

**Parameters:**
<RelBw> Range: 0.04 to 0.8
Increment: 0.01
*RST: 0.8

**Usage:** Asynchronous command

**Manual operation:** See "Rel. BW, Filter BW" on page 1018

---

**IQ:BWIDth?**

Returns the filter bandwidth.

See also: `IQ:RBWidth` on page 2369

**Return values:**
<AbsBw> Range: 40 to 16E+9
*RST: 4E+9
Default unit: Hz

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Rel. BW, Filter BW" on page 1018

---

**IQ:SRATe** <SampleRate>

Sets the required sample rate of the output I/Q data.

**Parameters:**
<SampleRate> Range: 1000 | 400E+6 to 20E+9 | 40E+9 (for input sample rate 20 Gsample/s | 40 Gsample/s)
Increment: 1
*RST: 10E+9
Default unit: Sa/s

**Usage:** Asynchronous command

**Manual operation:** See "Sample rate" on page 1018

---

**IQ:RLENgth** <RecordLength>

Sets the required record length of the output I/Q data. The resulting acquisition time of the I/Q data is:

*Acquisition time = Record length / Sample rate*

**Parameters:**

<RecordLength>
Range: 1 to 40E+6
Increment: 2
*RST: 100000
Default unit: pts

**Usage:** Asynchronous command

**Manual operation:** See "Record length" on page 1018

---

**CHANnel<m>:IQ:CFRequency** <IQCarrierFreqInput>

Sets the carrier frequency of the modulated RF signal or of the complex signal in IF range.

Prerequisites:

- `CHANnel<m>:IQ:INPType` on page 2368 is set to `REAL`
  or:
- `CHANnel<m>:IQ:INPType` on page 2368 is set to `COMPlex` and `CHANnel<m>:IQ:INPMode` on page 2368 is set to `RFIF`

**Suffix:**

<m>
1..4
Selects the input channel.

**Parameters:**

<IQCarrierFreqInput>
Range: 10E-3 to 20E+9
Increment: 100
*RST: 1E+6
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Carrier frequency" on page 1019

---

**CHANnel<m>:IQ:SBRF** <IQSidebandRF>

Defines the frequency position of the RF spectrum in the input signal: normal or inverse. The position is important for correct down-conversion and filtering.

Prerequisite: `CHANnel<m>:IQ:INPType` on page 2368 is set to `REAL`

For details, see "Sideband (real input)" on page 1019

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<IQSidebandRF> INVerse | NORMal
*RST: NORMal

**Usage:** Asynchronous command

**Manual operation:** See "Sideband (real input)" on page 1019

---

**CHANnel<m>:IQ:SBIF** <IQSidebandIF>

Defines the sideband and the frequency position of complex modulated input signal in IF range.

Prerequisites: `CHANnel<m>:IQ:INPType` on page 2368 is set to `COMPlex` and `CHANnel<m>:IQ:INPMode` on page 2368 is set to `RFIF`

For details, see "Sideband (complex IF input)" on page 1019

**Suffix:**
<m> 1..4
Selects the input channel.

**Parameters:**
<IQSidebandIF> NUPPer | NLOWer | IUPPer | ILOWer
NUPPer: Upper sideband in normal position
NLOWer: Lower sideband in normal position
IUPPer: Upper sideband in inverse position
ILOWer: Lower sideband in inverse position
*RST: NUPPer

**Usage:** Asynchronous command

**Manual operation:** See "Sideband (complex IF input)" on page 1019

---

**CHANnel<m>:IQ:WARNing[:STATe]?**

Returns ON if the sampling setup results in an unfavorable combination.

To get the message, use `CHANnel<m>:IQ:WARNing:MESSage?`.

**Suffix:**
<m> 1..4

**Return values:**
<WarningState> ON | OFF
*RST: OFF

**Usage:** Query only

**CHANnel<m>:IQ:WARNing:MESSage?**

Returns the message string if `CHANnel<m>:IQ:WARNing[:STATe]?` is true.

**Suffix:**
<m> 1..4

**Return values:**
<WarningMessage> The string is:
"Undersampling" if the carrier frequency is too fast.
"CF + BW/2 > DevBW" if bandwidth serttings do not fit. CF is the carrier frequency, BW is the filter bandwidth, and DevBW is the maximum instrument bandwidth.

**Usage:** Query only

### 24.21.2 I/Q data output

CHANnel<m>:IQ:DATA[:VALues]?..... 2372
CHANnel<m>:IQ:DATA:HEADer?..... 2373

**CHANnel<m>:IQ:DATA[:VALues]?**

Returns the recorded I/Q data.

**Suffix:**
<m> 1..4
Selects the input channel.
In case of a complex input signal that requires two input channels, the results of sources Ch1 and Ch2 are identical, as well as the results of Ch3 and Ch4.

**Return values:**
<Data> In-phase and and Quadrature floating values in interleaved order.

**Example:**
```
CHAN1:IQ:DATA?
-9.6296054835E-005, -1.5046258568E-006,
0.0001013283545, 1.5832555391E-006,
-0.00014297277085, -2.233616442E-006,
0.000192677413, 3.0105845781E-006,
-0.00020517286612, -3.2058260331E-006,
-0.002648930531, -4.1546467401E-005,
-0.0028401580639, -4.4135249482E-005,
-0.0028636774514, -4.4877564505E-005, ...
```

**Usage:** Query only

**Manual operation:** See "Save IQ data to file" on page 1021

**CHANnel<m>:IQ:DATA:HEADer?**

Returns the header of I/Q data.

***Table 24-19: Header data***

| Position | Meaning | Example |
|---|---|---|
| 1 | XStart, acquisition time before trigger, in s | -5E-008 = - 50 ns |
| 2 | XStop, acquisition time after trigger, in s | 5E-008 = 50 ns |
| 3 | Record length of the waveform in Samples | 1000 |
| 4 | Number of values per sample interval. | 2 (if I and Q values in the data) |

**Suffix:**
<m> 1..4
Selects the input channel.

**Example:** `CHAN1:IQ:DATA:HEAD?`
`-5E-008,5E-008,1000,1`

**Usage:** Query only

**Manual operation:** See "Save IQ data to file" on page 1021

## 24.22 Jitter analysis and clock data recovery

This chaptes decribes the remote commands of jitter option R&S RTP-K12 /K133 and CDR option R&S RTP-K140/K141.

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

- Jitter measurements (option R&S RTP-K12)..... 2374
- Advanced jitter and noise (option R&S RTP-K133/K134)..... 2379
- Clock data recovery (software-based, option R&S RTP-K12)..... 2406
- Eye mask testing..... 2412

### 24.22.1 Jitter measurements (option R&S RTP-K12)

The following table lists the measurement suffixes and the <MeasType> parameter value with a short description.

For a detailed description, see Chapter 18.1.1, "Jitter measurements", on page 1024.

***Table 24-20: Jitter measurements***

| Meas. suffix | <MeasType> parameter value | Meas. type | Description/Result |
|---|---|---|---|
| 37 | CCJitter | Cycle-cycle jitter | Difference between the periods of two adjacent cycles. The measurement is based on the period measurement. |
| 38 | NCJitter | N-cycle jitter | Difference between the time of two adjacent groups of N cycles (periods) each. |
| 39 | CCWidth | Cycle-cycle width | Difference between the pulse width of two adjacent cycles. The measurement is based on the pulse width measurement. |
| 40 | CCDutycycle | Cycle-cycle duty cycle | Difference between the duty cycle of two adjacent cycles. The measurement is based on the duty cycle measurement. |
| 41 | TIE | Time interval error | Time difference between the slope of the input signal and the slope of a reference signal. The reference signal can be a captured clock waveform, or a clock generated by clock data recovery (CDR, software algorithm or hardware generation). |
| 42 | UINTerval | Unit interval | Period of the clock signal. If no clock signal is available, it is recovered by CDR. The period is calculated as the time difference between two consecutive clock edges of the same polarity. |
| 43 | DRATe | Data rate | Frequency of the clock signal. If no clock signal is available, it is recovered by CDR. The measurement is based on the unit interval measurement. |
| 44 | SKWDelay | Skew delay | Delay between the edges of two interdependent waveforms. |
| 45 | SKWPhase | Skew phase | Phase difference between the edges of two waveforms. |
| 1 to 36; 46 | Used for amplitude/time measurements (limit checks). The jitter category uses the same limit checks as amplitude/time. See Chapter 24.12.3, "Amplitude/time measurements", on page 1436. | | |

MEASurement<m>:JITTer:CCSLope....2374
MEASurement<m>:JITTer:PULSe....2375
MEASurement<m>:JITTer:NCYCles....2376
MEASurement<m>:JITTer:CDRMode....2376
MEASurement<m>:JITTer:SOURce<n>:TIESlope....2377
MEASurement<m>:JITTer:DRUNit....2377
MEASurement<m>:JITTer:SKWSlope....2378
MEASurement<m>:JITTer:SKWRelation....2378

**MEASurement<m>:JITTer:CCSLope** <JitterCCSlope>

Selects the slope at which the periods and thus the jitter is measured.

The command is available for the following measurements: cycle-cycle jitter, N-cycle jitter, and cycle-cycle duty cycle (`MEASurement<m>:MAIN` and/or `MEASurement<m>:ADDitional` are set to measurements CCJitter | NCJitter | CCDutycycle).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<JitterCCSlope> FIRSt | POSitive | NEGative | EITHer

**FIRSt**
Measures the period from the first edge that is found, no matter of its direction.

**POSitive**
Measures the period at positive going edges.

**NEGative**
Measures the period at negative going edges.

**EITHer**
Measures the period at both positive and negative going edges. This option is useful, for example, to check the clock stability of a double data rate clock.

*RST: FIRSt

**Usage:** Asynchronous command

**Manual operation:** See "Cycle begin" on page 1028

---

**MEASurement<m>:JITTer:PULSe** <JittPlsPola>

Sets the polarity of pulses for which the pulse width is measured to obtain the cycle-cycle width and the cycle-cycle duty cycle.

The command is available for the following measurements: cycle-cycle width and cycle-cycle duty cycle (`MEASurement<m>:MAIN` and/or `MEASurement<m>:ADDitional` are set to measurements CCWidth | CCDutycycle).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<JittPlsPola> POSitive | NEGative

Pulse width of positive or negative pulses is measured, respectively.

*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Pulse polarity" on page 1028

**MEASurement<m>:JITTer:NCYCles** <JitterNofCycles>

Sets the number of periods (cycles) that are accumulated to measure the N-cycle jitter.

See also: Table 18-1

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<JitterNofCycles> Range: 1 to 2147483647
Increment: 1
*RST: 2

**Usage:** Asynchronous command

**Firmware/software:** FW 2.60. The algorithm of N-cycle measurement has been changed and `MEASurement:JITTer:COFFset` has been removed.

**Manual operation:** See "Number of cycles" on page 1028

**MEASurement<m>:JITTer:CDRMode** <JitterCDRMode>

Defines the origin of the clock signal - wether a real clock signal or a clock generated using one of the CDR methods.

The command is available for the following measurements: time-interval error, unit interval and data rate (`MEASurement<m>:MAIN` and/or `MEASurement<m>:ADDitional` are set to measurements TIE | UINTerval | DRATe).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<JitterCDRMode> SIGNal | SW1 | SW2 | HW

**SIGNal**
The clock is a real clock signal.

**SW1 | SW2**
The clock is generated by a software algorithm. You can define two software algorithms for CDR. To configure the CDR, use the `CDR:SOFTware` commands.

**HW**
The clock is generated from the source signal by the integrated CDR circuitry. The HW CDR must be active to acquire the jitter data (trigger type = CDR or serial pattern with CDR)

*RST: SW1

**Usage:** Asynchronous command

**Manual operation:** See "Clock" on page 1030

---

**MEASurement<m>:JITTer:SOURce<n>:TIESlope** <JitterTIESlope>

Sets the clock and data edges that are used for measurements. Clock or data is defined by the SOURce suffix <n>.

The command is available for the following measurements: time-interval error, unit interval and data rate (`MEASurement<m>:MAIN` and/or `MEASurement<m>:ADDitional` are set to measurements TIE | UINTerval | DRATe.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

<n> 1..2
1 = data slope (only relevant for time interval error measurements with explicit clock signal)
2 = clock slope

**Parameters:**

<JitterTIESlope> POSitive | NEGative | EITHer

**POSitive**
The positive clock slope can be used, for example, for single data rate (SDR) signals with bit start at the positive clock edge.

**NEGative**
The negative clock slope can be used, for example, for SDR signals with bit start at the negative clock edge.

**EITHer**
For clock edges, this option can be used for double data rate (DDR) signals.
For data edges, it is the most common setting.

*RST: EITHer

**Usage:** Asynchronous command

**Manual operation:** See "Slope" on page 1030

---

**MEASurement<m>:JITTer:DRUNit** <JittDatRateUnt>

Sets the unit for data rate measurements.

**Suffix:**

<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**

<JittDatRateUnt> BIT_S | Hz

*RST: BIT_S

**Usage:** Asynchronous command

**Manual operation:** See "Unit" on page 1031

---

**MEASurement<m>:JITTer:SKWSlope** <JitterSkewSlope>

Sets the edge of the first waveform from which the skew delay or phase is measured: positive, negative or both.

The command is available for the following measurement: skew delay and skew phase (`MEASurement<m>:MAIN` and/or `MEASurement<m>:ADDitional` are set to measurements SKWDelay | SKWPhase).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<JitterSkewSlope> POSitive | NEGative | EITHer
*RST: POSitive

**Usage:** Asynchronous command

**Manual operation:** See "Edge polarity" on page 1031

---

**MEASurement<m>:JITTer:SKWRelation** <JittskwRelt>

Sets the edge of the second waveform relative to the first waveform.

The command is available for the following measurements: skew delay and skew phase (`MEASurement<m>:MAIN` and/or `MEASurement<m>:ADDitional` are set to measurements SKWDelay | SKWPhase).

**Suffix:**
<m> 1..10
See "Selection of the measurement group: MEASurement<m>" on page 1426.

**Parameters:**
<JittskwRelt> MATChing | INVerse

**MATChing**
Measures from positive to positive edge or from negative to negative edge.

**INVerse**
Measures from positive to negative edge or from negative to positive edge.

*RST: MATChing

**Usage:** Asynchronous command

**Manual operation:** See "Relative polarity" on page 1031

## 24.22.2 Advanced jitter and noise (option R&S RTP-K133/K134)

This chapter contains all remote commands to set up the advanced jitter and advanced noise analysis.

**Selection of the advanced jitter measurement: ADVJitter<m>**

With R&S RTP, you can configure up to two simultaneous advanced jitter measurements. In manual operation, they are represented by subtabs "JA1" and "JA2" in the "Advanced Jitter" dialog box. For remote operation, they are indicated by the suffix `ADVJitter<m>`.

| Remote control: ADVJitter suffix <m> | Manual operation: "JA" subtab |
|---|---|
| 1 to 2 correspond to | "JA1" to "JA2" |


- Signal....2379
- Results....2387
- Advanced settings....2403


### 24.22.2.1 Signal


ADVJitter<m>:SIGNal:TYPE....2380
ADVJitter<m>:SIGNal:SOURce....2380
ADVJitter<m>:SIGNal:QUICkmeas....2380
ADVJitter<m>:SIGNal:BITRate....2380
ADVJitter<m>:SIGNal:CDRMode....2381
ADVJitter<m>:SIGNal:REFSource....2381
ADVJitter<m>:SIGNal:CLKSource....2381
ADVJitter<m>:SIGNal:EDGE....2381
ADVJitter<m>:SIGNal:CLEStimation....2382
ADVJitter<m>:SIGNal:CLFFactor....2382
ADVJitter<m>:SIGNal:CFEStimation....2382
ADVJitter<m>:SIGNal:CLFRequency....2382
ADVJitter<m>:SIGNal:CLOFfset....2383
ADVJitter<m>:SIGNal:DATRate?....2383
ADVJitter<m>:SIGNal:SAMTime....2383
ADVJitter<m>:SIGNal:GATE:COUPling....2383
ADVJitter<m>:SIGNal:GATE:CURSor....2384
ADVJitter<m>:SIGNal:GATE:RANGe:ABSolute:STARt....2384
ADVJitter<m>:SIGNal:GATE:RANGe:ABSolute:STOP....2384
ADVJitter<m>:SIGNal:GATE:RANGe:MODE....2385
ADVJitter<m>:SIGNal:GATE:RANGe:RELative:STARt....2385
ADVJitter<m>:SIGNal:GATE:RANGe:RELative:STOP....2385
ADVJitter<m>:SIGNal:GATE:ZDIagram....2385
ADVJitter<m>:SIGNal:GATE:STATe....2386
ADVJitter<m>:SIGNal:SERStandard....2386

**ADVJitter<m>:SIGNal:TYPE** <SignalType>

Select, if the analyzed signal is a data or clock signal.

**Suffix:**
<m> 1..2

**Parameters:**
<SignalType> CLOCk | DATNrz
*RST: DATNrz

**Manual operation:** See "Signal Type" on page 1038

**ADVJitter<m>:SIGNal:SOURce** <SourceType>

Selects the source for the advanced jitter analysis.

**Suffix:**
<m> 1..2

**Parameters:**
<SourceType> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | DIFF1 | DIFF2 | COMMON1 | COMMON2

**Manual operation:** See "Analyzed source" on page 1036

**ADVJitter<m>:SIGNal:QUICkmeas**

Starts a quick advanced jitter analysis.

**Suffix:**
<m> 1..2

**Usage:** Event

**Manual operation:** See "Quick start analysis" on page 1038

**ADVJitter<m>:SIGNal:BITRate** <NominalBitRate>

Sets the frequency of the feed forward CDR. It corresponds to the data rate of the data stream from which the clock is to be recovered.

**Suffix:**
<m> 1..2

**Parameters:**
<NominalBitRate> Range: 1000 to 10E+9
Increment: 10
*RST: 1E+9
Default unit: bps

**Manual operation:** See "Nominal bit rate" on page 1040

**ADVJitter<m>:SIGNal:CDRMode** <MethodType>

Selects the reference source for the CDR signal, if `ADVJitter<m>:SIGNal:REFSource` is set to `CDR`.

**Suffix:**
<m> 1..2

**Parameters:**
<MethodType> SIGNal | SW1 | SW2 | HW | REF
*RST: SW1

**Manual operation:** See "Reference source" on page 1040

**ADVJitter<m>:SIGNal:REFSource** <SourceType>

Selects if an explicit clock or a CDR signal is used as a reference signal.

**Suffix:**
<m> 1..2

**Parameters:**
<SourceType> CLOCk | CDR
*RST: CDR

**Manual operation:** See "Clock / CDR" on page 1040

**ADVJitter<m>:SIGNal:CLKSource** <SourceType>

Selects the reference source for the clock signal, if `ADVJitter<m>:SIGNal:REFSource` is set to `CLOCk`.

**Suffix:**
<m> 1..2

**Parameters:**
<SourceType> C1W1 | C1W2 | C1W3 | C2W1 | C2W2 | C2W3 | C3W1 | C3W2 | C3W3 | C4W1 | C4W2 | C4W3 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | DIFF1 | DIFF2 | COMMON1 | COMMON2

**Manual operation:** See "Reference source" on page 1040

**ADVJitter<m>:SIGNal:EDGE** <EdgeType>

The clock edge sets the point in time when the state of the data signal is checked.

**Suffix:**
<m> 1..2

**Parameters:**
<EdgeType> POSitive | NEGative | EITHer
*RST: EITHer

**Manual operation:** See "Clock edge" on page 1041

---

**ADVJitter<m>:SIGNal:CLEStimation**

Executes an estimation of the signal for the clock frequency.

**Suffix:**
<m> 1..2

**Usage:** Event
Asynchronous command

**Manual operation:** See "Estimate clock frequency" on page 1041

---

**ADVJitter<m>:SIGNal:CLFFactor** <Factor>

Sets a clock multiplier factor.

**Suffix:**
<m> 1..2

**Parameters:**
<Factor> Range: 1 to 100
Increment: 1
*RST: 1

**Manual operation:** See "Clock multiplier" on page 1041

---

**ADVJitter<m>:SIGNal:CFEStimation**

Executes an estimation of the signal for the clock offset.

**Suffix:**
<m> 1..2

**Usage:** Event
Asynchronous command

**Manual operation:** See "Estimate clock offset" on page 1042

---

**ADVJitter<m>:SIGNal:CLFRequency** <Frequency>

Sets the clock frequency for the explicit clock signal.

**Suffix:**
<m> 1..2

**Parameters:**
<Frequency> Range: 1000 to 10E+9
Increment: 10
*RST: 1E+9
Default unit: Hz

**Manual operation:** See "Clock frequency" on page 1041

**ADVJitter<m>:SIGNal:CLOFfset** <Offset>

Sets a clock offset for the explicit clock signal.

**Suffix:**
<m> 1..2

**Parameters:**
<Offset> Range: -1000 to 1000
Increment: 0.01
*RST: 0
Default unit: UI

**Manual operation:** See "Clock offset" on page 1042

**ADVJitter<m>:SIGNal:DATRate?**

Displays the calculated data rate of the explicit clock signal.

**Suffix:**
<m> 1..2

**Return values:**
<DatarateValue> Range: 1000 to 10E+9
Increment: 10
*RST: 1E+9
Default unit: bps

**Usage:** Query only

**Manual operation:** See "Data rate" on page 930

**ADVJitter<m>:SIGNal:SAMTime** <SamplingTime>

Sets a sampling time for the explicit clock signal, an offset for the clock edge in relation to the bit start. The clock edge sets the beginning of the unit interval. The sampling time is a number between 0 and 1. Value 0 sets the clock edge to the beginning of the bit period; value 0.5 sets the clock edge to the middle of the bit period.

**Suffix:**
<m> 1..2

**Parameters:**
<SamplingTime> Range: 0 to 1
Increment: 0.01
*RST: 0.5
Default unit: UI

**Manual operation:** See "Sampling time" on page 1042

**ADVJitter<m>:SIGNal:GATE:COUPling** <ModeType>

Sets if the gate is coupled to the zoom or cursor.

**Suffix:**
<m> 1..2

**Parameters:**
<ModeType> NONE | CURSor | ZOOM
*RST: NONE

**Manual operation:** See "Coupling mode" on page 1039

---

**ADVJitter<m>:SIGNal:GATE:CURSor** <CursorType>

Sets the cursor the gate is coupled to, if `ADVJitter<m>:SIGNal:GATE:COUPling` is set to `CURSor`.

**Suffix:**
<m> 1..2

**Parameters:**
<CursorType> CURSOR1 | CURSor1 | CURSOR2 | CURSor2 | CURSOR3 | CURSor3 | CURSOR4 | CURSor4
*RST: CURSOR1

**Manual operation:** See "Cursor / Zoom" on page 1039

---

**ADVJitter<m>:SIGNal:GATE:RANGe:ABSolute:STARt** <StartValue>

Sets the start time of the gate area in percent, if `ADVJitter<m>:SIGNal:GATE:RANGe:MODE` is set to `ABS`.

**Suffix:**
<m> 1..2

**Parameters:**
<StartValue> Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0

**Manual operation:** See "Start / Stop" on page 1040

---

**ADVJitter<m>:SIGNal:GATE:RANGe:ABSolute:STOP** <StopValue>

Sets the stop time of the gate area in percent, if `ADVJitter<m>:SIGNal:GATE:RANGe:MODE` is set to `ABS`.

**Suffix:**
<m> 1..2

**Parameters:**
<StopValue> Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Manual operation:** See "Start / Stop" on page 1040

**ADVJitter<m>:SIGNal:GATE:RANGe:MODE** <ModeType>

Selects if the gate area is defined in relative values (percent) or absolute.

**Suffix:**
<m> 1..2

**Parameters:**
<ModeType> ABS | REL
*RST: ABS

**Manual operation:** See "Mode" on page 1039

**ADVJitter<m>:SIGNal:GATE:RANGe:RELative:STARt** <StartValue>

Sets the relative start of the gate area in percent, if `ADVJitter<m>:SIGNal:GATE:RANGe:MODE` is set to `REL`.

**Suffix:**
<m> 1..2

**Parameters:**
<StartValue> Range: 0 to 100
Increment: 0.1
*RST: 0
Default unit: %

**Manual operation:** See "Relative start / stop" on page 1040

**ADVJitter<m>:SIGNal:GATE:RANGe:RELative:STOP** <StopValue>

Sets the relative stop of the gate area in percent, if `ADVJitter<m>:SIGNal:GATE:RANGe:MODE` is set to `REL`.

**Suffix:**
<m> 1..2

**Parameters:**
<StopValue> Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Manual operation:** See "Relative start / stop" on page 1040

**ADVJitter<m>:SIGNal:GATE:ZDIagram** <ZoomType>

Sets the zoom the gate is coupled to, if `ADVJitter<m>:SIGNal:GATE:COUPling` is set to `ZOOM`.

**Suffix:**
<m> 1..2

**Parameters:**
<ZoomType>

**Manual operation:** See "Cursor / Zoom" on page 1039

---

**ADVJitter<m>:SIGNal:GATE:STATe** <State>

Enables the gating setup for the advanced jitter analysis.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "Use gate" on page 1039

---

**ADVJitter<m>:SIGNal:SERStandard** <SerStandardType>

Sets the serial standard. Set `CUSTOM` to adjust the bit rate and the bandwidth manually.

The serial standards are available only if the instrument is able to support its maximum nominal bit rate.

**Suffix:**
<m> 1..2

**Parameters:**
<SerStandardType> CUSTOM | USB20 | USB31GEN1 | USB31GEN2 | DISPPORT11 | DISPPORT12 | FIBCHAN1 | FIBCHAN2 | FIBCHAN4 | FIBCHAN8 | FIBCHAN16 | HDMI12 | HDMI14 | HDMI20 | PCIGEN1 | PCIGEN2 | PCIGEN3 | PCIGEN4 | SATAGEN1500 | SATAGEN11667 | SATAGEN1 | SATAGEN2500 | SATAGEN21667 | SATAGEN2 | SATAGEN3500 | SATAGEN31667 | SATAGEN3 | SASGEN1 | SASGEN2 | SASGEN3 | TB10G | XAUI | CUSTOM | USB20 | USB31GEN1 | DISPPORT11 | DISPPORT12 | FIBCHAN1 | FIBCHAN2 | FIBCHAN4 | HDMI12 | HDMI14 | HDMI20 | PCIGEN1 | PCIGEN2 | SATAGEN1500 | SATAGEN11667 | SATAGEN1 | SATAGEN2500 | SATAGEN21667 | SATAGEN2 | SATAGEN3500 | SATAGEN31667 | SATAGEN3 | SASGEN1 | SASGEN2 | XAUI | | USB31GEN2 | FIBCHAN8 | FIBCHAN16 | PCIGEN3 | PCIGEN4 | SASGEN3 | TB10G
*RST: CUSTOM

**Usage:** Asynchronous command

**Manual operation:** See "Serial standard" on page 1036

#### 24.22.2.2 Results

For remote operation, the component is indicated by the suffix `COMPonents<n>`.

***Table 24-21: Advanced jitter components***

| Suffix <n> | Component | Description/Result |
|---|---|---|
| **Basic components** | | |
| 1 | Symbol rate | Number of data symbols per second of the analyzed signal. It is equivalent to the baud rate.<br>For binary signals, the signal rate is equal to the bit rate. |
| 2 | Symbol duration | The length of time to transfer one signal symbol. Equals the inverse value of the symbol rate. |
| 3 | Event Count | Number of analyzed data transitions. |
| **Jitter components (option R&S RTP-K133)** | | |
| 4 | TJ@BER | The eye closure at the user-defined target BER. |
| 5 | RJ+(O)BUJ | Consists of any influence that cannot be predicted with model parameters. Usually, this jitter component follows a Gaussian probability distribution. Other Bounded Uncorrelated Jitter (OBUJ) is included here. OBUJ refers to any jitter that is bounded, uncorrelated time interval error (TIE) distribution, for example from unobserved data signals. |
| 6 | DJ (δ-δ) | Industry standard approximation to the deterministic jitter measurement, calculated by fitting a model of two overlapping, equally large Gaussian distributions to the observed DJ histogram. The calculated value is the time difference between the two distributions center points. |
| 7 | TJ | Measures the total Time Interval Error (TIE) of the input signal.<br>It is defined as the time between each transition of the input signal and its corresponding reference clock transition. |
| 8 | DJ | The reproducible portion of the overall TIE of the input signal.<br>It consists of the Data Dependent Jitter (DDJ) and Periodic Jitter (PJ) components. |
| 9 | DDJ | Consists of Inter-symbol Interference (ISI) and Duty Cycle Distortion (DCD).<br>It depends on the symbol sequence. ISI is an effect commonly caused by limited signal bandwidth in the transmitter and transmission line, which forces influence from past symbols onto future symbols. DCD is a mismatch between rising and falling signal transitions due to reference level mismatch or other imbalances between transitions. |
| 10 | DCD | Measure of imbalance between rising and falling data transitions. It is defined as the time difference between the average rising and average falling edge. |
| 11 | PJ | Caused by crosstalk from periodic signal sources to the transmitter or transmission line.<br>It represents the harmonic portion of the overall time interval error (TIE). |
| 12 | ISI | The time between the earliest and the latest signal transition of a single transition type, i.e.rising and falling edges. The measurement result is the largest value across all transition types. |

| Suffix <n> | Component | Description/Result |
|---|---|---|
| 13 | DDJ +PJ | Composite impact of all unknown but deterministic portions of the overall time interval error of the input signal. It consists of any disturbance that comes from a deterministic source but cannot be reproduced from the input signal. Often it is caused by crosstalk from other data signals. |
| 14 | (O)BUJ | Composite impact of all unknown but deterministic portions of the overall time interval error of the input signal. It consists of any disturbance that comes from a deterministic source but cannot be reproduced from the input signal. Often it is caused by crosstalk from other data signals. |
| 15 | (O)BUJ (δ-δ) | Other bounded uncorrelated Dual-Dirac jitter (O)BUJ (δ- δ) is an industry standard approximation to the other bounded uncorrelated jitter measurement. It is calculated by fitting a model of two overlapping, equally large Gaussian distributions to the observed RJ+(O)BUJ histogram. The calculated value is the amplitude difference between the two distributions center points. |
| 16 | RJ | Consists of any influence that cannot be predicted with model parameters. Usually, this jitter component follows a Gaussian probability distribution. The algorithm computes the standard deviation and the peak-peak value of the TIE caused by RJ within the analyzed waveform. |
| **Noise components (option R&S RTP-K134)** | | |
| 17 | EH@BER | Returns the vertical eye opening at the user-defined BER. |
| 18 | RN+(O)BUN | Random noise plus (other) bounded uncorrelated noise or (other) bounded uncorrelated noise is the combined effect of both noise types.<br>If periodic components are activated RN + OBUN are displayed, otherwise RN + BUN.<br>RN + (O)BUN is typically used, if track or spectrum plots are desired. Both plots are not available for RN only or (O)BUN only components as the time dependencies of these noise types are unknown. RN + (O)BUN spectrum is a useful approximation of the RN spectrum for negligible small (O)BUN. |
| 20 | TN | Measures the total level error of the input signal. It is defined as the amplitude deviation between the amplitude of the input signal at the decision time point and its corresponding reference level. |
| 21 | DN | The reproducible portion of the overall level error of the input signal. It consists of the data-dependent noise, the periodic noise components and the other bounded uncorrelated jitter components. |
| 22 | DDN | Consists of inter-symbol interface noise and level distortion. It depends on the symbol sequence. ISIN is an effect whereby past symbols have an influence on future symbols.<br>LD is a mismatch between rising and falling signal transitions due to reference level mismatch or other imbalances between transitions. |
| 23 | LD | A measure of imbalance between different symbol levels. It is defined as the difference between the average level errors of the symbols one and zero. |
| 24 | PN | Periodic noise is typically caused by crosstalk from periodic signal sources to the transmitter or transmission line. It represents the harmonic portion of the overall level error. |
| 25 | ISIN | The level difference between the highest and the lowest level error of a symbol, i.e. one or zero.The largest value across all symbols is the measurement result. |

| Suffix <n> | Component | Description/Result |
|---|---|---|
| 26 | DDN + PN | Data-dependent noise plus periodic noise is the combined effect of both jitter types.<br>Note that DDN+PN equals deterministic noise minus other bounded uncorrelated noise.<br>DDN+PN spectrum is a useful approximation of the deterministic noise spectrum fot negligible small other bounded uncorrelated noise. |
| 27 | (O)BUN | Composite impact of all unknown but deterministic portions of the overall time interval error of the input signal. It consists of any disturbance that comes from a deterministic source but cannot be reproduced from the input signal. Often it is caused by crosstalk from other data signals. |
| 28 | (O)BUN (δ-δ | Other bounded uncorrelated Dual-Dirac noise (O)BUN (δ- δ) is an industry standard approximation to the other bounded uncorrelated jitter measurement. It is calculated by fitting a model of two overlapping, equally large Gaussian distributions to the observed RN+(O)BUN histogram. The calculated value is the amplitude difference between the two distributions center points. |
| 29 | RN | Consists of any influence that cannot be predicted with model parameters. Usually, this noise component follows a Gaussian probability distribution. The algorithm computes the standard deviation and the peak-peak value of the level error (LE) caused by RN within the analyzed waveform. |

## General

ADVJitter<m>:RESult:ENABle....2389
ADVJitter<m>:DCOMposition:COMPonents<n>:ENABle....2390
ADVJitter<m>:RESult:STATistics....2390
ADVJitter<m>:RESult:STEPresponse....2390
ADVJitter<m>:RESult:COMPonents<n>:HISTogram....2390
ADVJitter<m>:RESult:COMPonents<n>:SPECtrum....2391
ADVJitter<m>:RESult:COMPonents<n>:TRACk....2391
ADVJitter<m>:RESult:UNIT....2391
ADVJitter<m>:RESult:COMPonents<n>:STATistics:MAXPeakpeak?....2392
ADVJitter<m>:RESult:COMPonents<n>:STATistics:MAXStddev?....2392
ADVJitter<m>:RESult:COMPonents<n>:STATistics[:MINStddev]?....2392
ADVJitter<m>:RESult:COMPonents<n>:STATistics:MINPeakpeak?....2392
ADVJitter<m>:RESult:COMPonents<n>:STATistics:PEAKpeak?....2392
ADVJitter<m>:RESult:COMPonents<n>:STATistics:WFMCount?....2392
ADVJitter<m>:RESult:COMPonents<n>:STATistics:STDDev?....2392
ADVJitter<m>:RESult:PERiodic<n>:COUNt?....2392
ADVJitter<m>:RESult:PERiodic<n>:DIRection?....2392
ADVJitter<m>:RESult:PERiodic<n>:VALue?....2393
ADVJitter<m>:RESult:PERiodic<n>[:FREQuency]?....2393

**ADVJitter<m>:RESult:ENABle** <State>

Enables the advanced jitter analysis. It displays all enabled diagrams and displays a table with the results for the component measurements.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "Enable" on page 1045

---

**ADVJitter<m>:DCOMposition:COMPonents<n>:ENABle** <State>

Enables the measurement for the specified jitter component.

**Suffix:**
<m> 1..2
<n> 1..29
See Table 24-21

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "Basic components" on page 1048

---

**ADVJitter<m>:RESult:STATistics** <State>

Enables statistics measurements for the advanced jitter analysis.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "Statistics" on page 1046

---

**ADVJitter<m>:RESult:STEPresponse** <State>

Enables a step response diagram for the advanced jitter measurements..

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "Step response" on page 1046

---

**ADVJitter<m>:RESult:COMPonents<n>:HISTogram** <State>

Enables a histogram for the selected measurement component.

**Suffix:**
<m> 1..2
<n> 1..29
See Table 24-21

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "Component specific results" on page 1046

---

**ADVJitter<m>:RESult:COMPonents<n>:SPECtrum** <State>

Enables a spectrum diagram for the selected measurement component.

**Suffix:**
<m> 1..2
<n> 1..29
See Table 24-21

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "Component specific results" on page 1046

---

**ADVJitter<m>:RESult:COMPonents<n>:TRACk** <State>

Enables a track for the advanced jitter measurements.

**Suffix:**
<m> 1..2
<n> 1..29
See Table 24-21

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "Component specific results" on page 1046

---

**ADVJitter<m>:RESult:UNIT** <UnitType>

Selects between the absolute and relative unit for the displayed results.

**Suffix:**
<m> 1..2

**Parameters:**
<UnitType> ABS | REL
*RST: ABS

**Manual operation:** See "Unit" on page 1047

---

**ADVJitter<m>:RESult:COMPonents<n>:STATistics:MAXPeakpeak?** <PeakValue>
**ADVJitter<m>:RESult:COMPonents<n>:STATistics:MAXStddev?** <StdDevValue>
**ADVJitter<m>:RESult:COMPonents<n>:STATistics[:MINStddev]?** <StdDevValue>
**ADVJitter<m>:RESult:COMPonents<n>:STATistics:MINPeakpeak?** <PeakValue>
**ADVJitter<m>:RESult:COMPonents<n>:STATistics:PEAKpeak?**
**ADVJitter<m>:RESult:COMPonents<n>:STATistics:WFMCount?** <CountValue>
**ADVJitter<m>:RESult:COMPonents<n>:STATistics:STDDev?**

Return the statistic results of the specified measurement.

- `MAXPeakpeak`: absolute maximum value of the component.
- `MAXStddev`: standard deviation of the maximum value.
- `MINPeakpeak`: absolute minimum value of the component.
- `MINStddev`: standard deviation of the minimum value.
- `PEAKpeak`: peak to peak value of the long-term measurement results.
- `WFMCount`: number of waveforms (acquisitions) the measurement is based on.
- `STDDev`: standard deviation of the long-term measurement results.

**Suffix:**
<m> 1..2

<n> 1..29
See Table 24-21

**Return values:**
<StdDev> Range: -100E+24 to 100E+24
Increment: 100E-12
*RST: 0

**Usage:** Query only

---

**ADVJitter<m>:RESult:PERiodic<n>:COUNt?**

Returns the number of periodic components.

**Suffix:**
<m> 1..2

<n> *
Not relevant

**Return values:**
<Count>

**Usage:** Query only

---

**ADVJitter<m>:RESult:PERiodic<n>:DIRection?**

Returns the direction of the selected periodic component.

**Suffix:**
<m> 1..2
<n> *
Number of periodic components.

**Return values:**
<CompDirection> UNDefined | VERTical | HOEXtrinsic | HOINtrinsic
`HOEXtrinsic`: horizontal extrinsic
`HOINtrinsic`: horizontal intrinsic
*RST: UNDefined

**Usage:** Query only

---

**ADVJitter<m>:RESult:PERiodic<n>:VALue?**

Returns the value of the selected periodic component.

**Suffix:**
<m> 1..2
<n> *
Number of periodic components.

**Return values:**
<CompValue> Range: -1000 to 1000
Increment: 0
*RST: 0

**Usage:** Query only

---

**ADVJitter<m>:RESult:PERiodic<n>[:FREQuency]?**

Returns the frequency of the selected periodic component.

**Suffix:**
<m> 1..2
<n> *
Number of periodic components.

**Return values:**
<FreqValue> Range: 0 to 1E+15
Increment: 0
*RST: 0
Default unit: Hz

**Usage:** Query only

### Measurements

ADVJitter<m>:DCOMposition:CLEBasicmeas ..... 2394
ADVJitter<m>:DCOMposition:SELBasicmeas ..... 2394
ADVJitter<m>:DCOMposition:CLEJitcomp ..... 2394

ADVJitter<m>:DCOMposition:SELJitcomp....2394
ADVJitter<m>:DCOMposition:CLENoisecomp....2395
ADVJitter<m>:DCOMposition:SELNoisecomp....2395
ADVJitter<m>:RESult:TRANsitions....2395
ADVJitter<m>:RESult:LEVels....2395

**ADVJitter<m>:DCOMposition:CLEBasicmeas**

Clears all selected basic jitter components.

**Suffix:**
<m> 1..2

**Usage:** Event
Asynchronous command

**Manual operation:** See "Select all /Clear" on page 1049

**ADVJitter<m>:DCOMposition:SELBasicmeas**

Selects all basic jitter components.

**Suffix:**
<m> 1..2

**Usage:** Event
Asynchronous command

**Manual operation:** See "Select all /Clear" on page 1049

**ADVJitter<m>:DCOMposition:CLEJitcomp**

Clears all selected jitter components.

**Suffix:**
<m> 1..2

**Usage:** Event
Asynchronous command

**Manual operation:** See "Select all /Clear" on page 1053

**ADVJitter<m>:DCOMposition:SELJitcomp**

Selects all jitter components.

**Suffix:**
<m> 1..2

**Usage:** Event
Asynchronous command

**Manual operation:** See "Select all /Clear" on page 1053

**ADVJitter<m>:DCOMposition:CLENoisecomp**

Clears all selected noise components.

**Suffix:**
<m> 1..2

**Usage:** Event
Asynchronous command

**Manual operation:** See "Select all /Clear" on page 1057

**ADVJitter<m>:DCOMposition:SELNoisecomp**

Selects all noise components.

**Suffix:**
<m> 1..2

**Usage:** Event
Asynchronous command

**Manual operation:** See "Select all /Clear" on page 1057

**ADVJitter<m>:RESult:TRANsitions** <TransitionType>

Sets the measurements for which transitions are displayed in the diagram.

**Suffix:**
<m> 1..2

**Parameters:**
<TransitionType> RISing | FALLing | BOTH
*RST: BOTH

**Manual operation:** See "Transitions" on page 1053

**ADVJitter<m>:RESult:LEVels** <LevelType>

Sets which levels are considered for the display of the diagrams, rising, falling or both. This setting is mostly relevant for histograms.

**Suffix:**
<m> 1..2

**Parameters:**
<LevelType> HIGH | LOW | BOTH
*RST: BOTH

**Manual operation:** See "Levels" on page 1058

### Bathtub

ADVJitter<m>:DCOMposition:TARBer....2396
ADVJitter<m>:RESult:BATHtub....2396
ADVJitter<m>:RESult:TYPBathtub....2396
ADVJitter<m>:RESult:PJ....2397
ADVJitter<m>:RESult:OBUJ....2397
ADVJitter<m>:RESult:NBAThtub....2397
ADVJitter<m>:RESult:PN....2397
ADVJitter<m>:RESult:OBUN....2398

**ADVJitter<m>:DCOMposition:TARBer** <TargetBtErrRate>

Sets the target bit error rate for total jitter at BER components.

**Suffix:**
<m> 1..2

**Parameters:**
<TargetBtErrRate> Range: 1 to 32
Increment: 1
*RST: 12

**Manual operation:** See "Target BER 10" on page 1058

**ADVJitter<m>:RESult:BATHtub** <State>

Enables a bathtub plot for the advanced jitter measurements.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "DDJ bathtub" on page 1058

**ADVJitter<m>:RESult:TYPBathtub** <BathtubType>

Selects if you want to display the bathtub results versus total jitter (TJ) or data-dependent jitter (DDJ) and random jitter (RJ) components.

**Suffix:**
<m> 1..2

**Parameters:**
<BathtubType> DDJ | DDJP | DDJB | DDJO | TJ | DDJRj
`TJ`: Total jitter
`DDJRJ`: Data-dependent jitter and random jitter.
*RST: TJ

**ADVJitter<m>:RESult:PJ** <State>

If enabled, the periodic jitter components are considered for the jitter bathtub.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: ON

**Manual operation:** See "PJ / (O)BUJ components" on page 1059

**ADVJitter<m>:RESult:OBUJ** <State>

If enabled, the other bounded uncorrelated jitter components are considered for the jitter bathtub.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: ON

**Manual operation:** See "PJ / (O)BUJ components" on page 1059

**ADVJitter<m>:RESult:NBAThtub** <State>

Enables a noise bathtub plot, a graph of BER in a log scale versus data-dependent jitter (DDN).

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "DDN bathtub" on page 1059

**ADVJitter<m>:RESult:PN** <State>

If enabled, the periodic noise components are considered for the noise bathtub.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: ON

**Manual operation:** See "PN / (O)BUN components" on page 1060

**ADVJitter<m>:RESult:OBUN** <State>

If enabled, the other bounded uncorrelated noise components are considered for the noise bathtub.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: ON

**Manual operation:** See "PN / (O)BUN components" on page 1060

### Eye

ADVJitter<m>:RESult:PHORizontal ..... 2398
ADVJitter<m>:RESult:PVERtical ..... 2398
ADVJitter<m>:RESult:SYNTheticeye ..... 2398
ADVJitter<m>:RESult:TYPSynteye ..... 2399

**ADVJitter<m>:RESult:PHORizontal** <State>

If enabled, adds horizontal periodic components to the data-dependent synthetic eye.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "Add periodic components to DD synthetic eye" on page 1061

**ADVJitter<m>:RESult:PVERtical** <State>

If enabled, adds vertical periodic components to the data-dependent synthetic eye.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "Add periodic components to DD synthetic eye" on page 1061

**ADVJitter<m>:RESult:SYNTheticeye** <State>

Selects if you want to display synthetic eye results only for the deterministic jitter (DJ) components or the data-dependent jitter (DDJ) components.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "Synthetic eye" on page 1061

---

**ADVJitter<m>:RESult:TYPSynteye** <SyntheticEyeTyp>

Selects if you want to display synthetic eye results only for the deterministic jitter (DDP) components or the data-dependent jitter (DD) components.

**Suffix:**
<m> 1..2

**Parameters:**
<SyntheticEyeTyp> DD | DDPH | DDPV | DDP
*RST: DDP

### Diagram

ADVJitter<m>:RESult:VERTical:HBINs....2399
ADVJitter<m>:RESult:VERTical:CONTinuous....2400
ADVJitter<m>:RESult:VERTical:BMTYpe....2400
ADVJitter<m>:RESult:VERTical:BHContinuous....2400
ADVJitter<m>:RESult:VERTical:JSContinuous....2400
ADVJitter<m>:RESult:VERTical:NHContinuous....2401
ADVJitter<m>:RESult:VERTical:NSContinuous....2401
ADVJitter<m>:RESult:COMPonents<n>:SCALe....2401
ADVJitter<m>:RESult:COMPonents<n>:OFFSet....2402
ADVJitter<m>:DCOMposition:SPJTable....2402
ADVJitter<m>:DCOMposition:SPJLabels....2403
ADVJitter<m>:DCOMposition:SPNLabels....2403
ADVJitter<m>:DCOMposition:JINTerleaved....2403

---

**ADVJitter<m>:RESult:VERTical:HBINs** <MaxBinNumber>

Sets the number of bins, the number of data intervals, of the histogram.

**Suffix:**
<m> 1..2

**Parameters:**
<MaxBinNumber> Range: 2 to 1000
Increment: 10
*RST: 200

**Manual operation:** See "Max. numb of bins" on page 1061

**ADVJitter<m>:RESult:VERTical:CONTinuous** <State>

If enabled, the measurement scale is set automatically.

If disabled you can manually set a `ADVJitter<m>:RESult:COMPonents<n>:SCALe` and a `ADVJitter<m>:RESult:COMPonents<n>:OFFSet`.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: ON

**Manual operation:** See "Continuous auto scale" on page 1062

**ADVJitter<m>:RESult:VERTical:BMTYpe** <MeasType>

Selects the basic measurement type to apply the vertical scale to.

**Suffix:**
<m> 1..2

**Parameters:**
<MeasType> SBRate | SBDuration
`SBRate`: Symbol rate
`SBDuration`: Symbol duration
*RST: SBRate

**Manual operation:** See "Measurement type" on page 1062

**ADVJitter<m>:RESult:VERTical:BHContinuous** <State>

If enabled, the measurement scale of the basic jitter measurement histogram and track is set automatically.

If disabled you can manually set a `ADVJitter<m>:RESult:COMPonents<n>:SCALe` and a `ADVJitter<m>:RESult:COMPonents<n>:OFFSet`.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: ON

**Manual operation:** See "Continuous auto scale" on page 1062

**ADVJitter<m>:RESult:VERTical:JSContinuous** <State>

If enabled, the measurement scale of the jitter spectrum is set automatically.

If disabled you can manually set a `ADVJitter<m>:RESult:COMPonents<n>:SCALe` and a `ADVJitter<m>:RESult:COMPonents<n>:OFFSet`.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: ON

**Manual operation:** See "Continuous auto scale" on page 1062

---

**ADVJitter<m>:RESult:VERTical:NHContinuous** <State>

If enabled, the measurement scale of the noise histogram and track is set automatically.

If disabled you can manually set a `ADVJitter<m>:RESult:COMPonents<n>:SCALe` and a `ADVJitter<m>:RESult:COMPonents<n>:OFFSet`.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: ON

**Manual operation:** See "Continuous auto scale" on page 1062

---

**ADVJitter<m>:RESult:VERTical:NSContinuous** <State>

If enabled, the measurement scale for the noise spectrum is set automatically.

If disabled you can manually set a `ADVJitter<m>:RESult:COMPonents<n>:SCALe` and a `ADVJitter<m>:RESult:COMPonents<n>:OFFSet`.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: ON

**Manual operation:** See "Continuous auto scale" on page 1062

---

**ADVJitter<m>:RESult:COMPonents<n>:SCALe** <Scale>

Sets the scale for the selected component.

You can set separate scale values for:

- `n=1` (symbol rate)
- `n= 2` (symbol duration)

- `n= 4...16`: sets only one offset value that is relevant for all advanced jitter components.
- `n= 17...29`: sets only one offset value that is relevant for all advanced noise components.

**Suffix:**
<m> 1..2

<n> 1..29
See Table 24-21

**Parameters:**
<Scale> Range: 1E-15 to 1E+15
Increment: 10E-6
*RST: 0.5
Default unit: s/div

**Manual operation:** See "Meas scale" on page 1062

---

**ADVJitter<m>:RESult:COMPonents<n>:OFFSet** <Offset>

Sets the offset for the selected component.

You can set separate offset values for:

- `n=1` (symbol rate)
- `n= 2` (symbol duration)
- `n= 4...16`: sets only one offset value that is relevant for all advanced jitter components.
- `n= 17...29`: sets only one offset value that is relevant for all advanced noise components.

**Suffix:**
<m> 1..2

<n> 1..29
See Table 24-21

**Parameters:**
<Offset> Range: -1E+15 to 1E+15
Increment: 1E-6
*RST: 0
Default unit: s

**Manual operation:** See "Meas offset" on page 1062

---

**ADVJitter<m>:DCOMposition:SPJTable** <State>

Enables the display of the periodic jitter components table.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: ON

**Manual operation:** See "PJ periodic component table" on page 1063

---

**ADVJitter<m>:DCOMposition:SPJLabels** <State>

Enables the display of the periodic jitter components labels.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "PJ periodic component labels" on page 1063

---

**ADVJitter<m>:DCOMposition:SPNLabels** <State>

Enables the display of the periodic noise components labels.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "PJ periodic component labels" on page 1063

---

**ADVJitter<m>:DCOMposition:JINTerleaved** <State>

Displays interleaved histogram values.

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "Interleaved" on page 1061

### 24.22.2.3 Advanced settings

ADVJitter<m>:DCOMposition:MRESplength? ..... 2404
ADVJitter<m>:DCOMposition:RLEStimate ..... 2404
ADVJitter<m>:DCOMposition:POSStep ..... 2404
ADVJitter<m>:DCOMposition:RESPlength ..... 2404
ADVJitter<m>:DCOMposition:ENP ..... 2405

ADVJitter<m>:DCOMposition:NHORizontal ..... 2405
ADVJitter<m>:DCOMposition:NVERtical ..... 2405
ADVJitter<m>:DCOMposition:ENOBu ..... 2406

**ADVJitter<m>:DCOMposition:MRESplength?**

Queries the estimated maximum step length response.

**Suffix:**
<m> 1..2

**Return values:**
<MaxLength> Range: 1 to 200
Increment: 1
*RST: 20
Default unit: UI

**Usage:** Query only

**Manual operation:** See "Length" on page 1064

**ADVJitter<m>:DCOMposition:RLEStimate**

Runs an estimate of the step response for the advanced jitter analysis measurements.

**Suffix:**
<m> 1..2

**Usage:** Event
Asynchronous command

**Manual operation:** See "Length" on page 1064

**ADVJitter<m>:DCOMposition:POSStep** <Position>

Sets the steps position.

**Suffix:**
<m> 1..2

**Parameters:**
<Position> Range: 0 to 100
Increment: 1
*RST: 50
Default unit: %

**Manual operation:** See "Step position" on page 1064

**ADVJitter<m>:DCOMposition:RESPlength** <Length>

Set the step response length in unit intervals.

**Suffix:**
<m> 1..2

**Parameters:**

<Length>

Range: 1 to 200
Increment: 1
*RST: 20
Default unit: UI

**Manual operation:** See "Length" on page 1064

---

**ADVJitter<m>:DCOMposition:ENP** <State>

Enables the display of the periodic jitter components table.

**Suffix:**

<m> 1..2

**Parameters:**

<State> ON | OFF

*RST: ON

**Manual operation:** See "Enable" on page 1065

---

**ADVJitter<m>:DCOMposition:NHORizontal** <MaxNumber>

Sets the maximum number of periodic components in the horizontal direction that are considered for the measurement. If the actual number of periodic components is higher than the set value, they are considered for the random jitter component.

**Suffix:**

<m> 1..2

**Parameters:**

<MaxNumber>

Range: 0 to 10
Increment: 1
*RST: 5

**Manual operation:** See "Horizonta (h) direction" on page 1065

---

**ADVJitter<m>:DCOMposition:NVERtical** <MaxNumber>

Sets the maximum number of periodic components in the vertical direction that are considered for the measurement. If the actual number of periodic components is higher than the set value, they are considered for the random jitter component.

**Suffix:**

<m> 1..2

**Parameters:**

<MaxNumber>

Range: 0 to 10
Increment: 1
*RST: 5

**Manual operation:** See "Vertical (v) direction" on page 1065

**ADVJitter<m>:DCOMposition:ENOBu** <State>

**Suffix:**
<m> 1..2

**Parameters:**
<State> ON | OFF
*RST: OFF

**Manual operation:** See "(Other) bounded uncorrelated" on page 1065

### 24.22.3 Clock data recovery (software-based, option R&S RTP-K12)

CDR:SOFTware<m>:ALGorithm....2406
CDR:SOFTware<m>:SERStandard....2407
CDR:SOFTware<m>:ESBRate:SOURce....2407
CDR:SOFTware<m>:ESBRate:BREStimate....2408
CDR:SOFTware<m>:ESBRate:CONTinuous....2408
CDR:SOFTware<m>:BITRate....2408
CDR:SOFTware<m>:ESLope....2409
CDR:SOFTware<m>:SYNC....2409
CDR:SOFTware<m>:RESults....2410
CDR:SOFTware<m>:PLL:ORDer....2410
CDR:SOFTware<m>:PLL:BWIDth....2410
CDR:SOFTware<m>:PLL:RELBwidth....2411
CDR:SOFTware<m>:PLL:DAMPing....2411
CDR:SOFTware<m>:REVLockpll....2411
CDR:SOFTware<m>:SAMTime....2412

**CDR:SOFTware<m>:ALGorithm** <Algorithm>

Sets the software algorithm that is used for clock data recovery.

**Suffix:**
<m> 1..2
Number of the software CDR setup

**Parameters:**
<Algorithm> CFRequency | PLL | FF | PLLStart

**CFRequency**
Constant frequency. CDR uses the nominal bit rate to generate the clock signal. The method assumes that the frequency of the signal is constant during the complete acquisition.

**PLL**
Phase-locked loop control system. It can follow slow deviations in the frequency of the data stream.

**FF**
Feed forward algorithm, a mathematical optimization method. Feed forward is available if none of the listed serial standards is used (`CDR:SOFTware<m>:SERStandard` is set to `CUSTOM`). It is faster than PLL, i.e., the settling time is shorter. Feed forward is also more precise than PLL. To set up the algorithm, set the bandwidth and the relative bandwith.

*RST: PLLStart

**Usage:** Asynchronous command

**Manual operation:** See "Algorithm" on page 1068

---

**CDR:SOFTware<m>:SERStandard** <SerialStandard>

Sets the serial standard. The nominal bit rate and bandwidth settings are set according to the selected standard, no estimation is required. The serial standards are available only if the instrument is able to support its maximum nominal bit rate.

Set `CUSTOM` to adjust the bit rate and the bandwidth manually.

**Suffix:**
<m> 1..2
Number of the software CDR setup

**Parameters:**
<SerialStandard> CUSTOM | USB20 | USB31GEN1 | USB31GEN2 | DISPPORT11 | DISPPORT12 | FIBCHAN1 | FIBCHAN2 | FIBCHAN4 | FIBCHAN8 | FIBCHAN16 | HDMI12 | HDMI14 | HDMI20 | PCIGEN1 | PCIGEN2 | PCIGEN3 | PCIGEN4 | SATAGEN1500 | SATAGEN11667 | SATAGEN1 | SATAGEN2500 | SATAGEN21667 | SATAGEN2 | SATAGEN3500 | SATAGEN31667 | SATAGEN3 | SASGEN1 | SASGEN2 | SASGEN3 | TB10G | XAUI | CUSTOM | USB20 | USB31GEN1 | DISPPORT11 | DISPPORT12 | FIBCHAN1 | FIBCHAN2 | FIBCHAN4 | HDMI12 | HDMI14 | HDMI20 | PCIGEN1 | PCIGEN2 | SATAGEN1500 | SATAGEN11667 | SATAGEN1 | SATAGEN2500 | SATAGEN21667 | SATAGEN2 | SATAGEN3500 | SATAGEN31667 | SATAGEN3 | SASGEN1 | SASGEN2 | XAUI | | USB31GEN2 | FIBCHAN8 | FIBCHAN16 | PCIGEN3 | PCIGEN4 | SASGEN3 | TB10G

*RST: CUSTOM

**Usage:** Asynchronous command

**Manual operation:** See "Serial standard" on page 1068

---

**CDR:SOFTware<m>:ESBRate:SOURce** <BitrateSource>

Selects the source of the data stream, which is analized to recover the clock.

**Suffix:**
<m> 1..2
Number of the software CDR setup

**Parameters:**
<BitrateSource> C1W1 | C2W1 | C3W1 | C4W1 | R1 | R2 | R3 | R4

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 1068

---

**CDR:SOFTware<m>:ESBRate:BREStimate**

Estimates the bit rate once, for the current acquisition.

**Suffix:**
<m> 1..2
Number of the software CDR setup

**Usage:** Event
Asynchronous command

**Manual operation:** See "Bit rate estimation" on page 1069

---

**CDR:SOFTware<m>:ESBRate:CONTinuous** <BitrateEstimation>

Enables ongoing correction of the bit rate.

**Suffix:**
<m> 1..2
Number of the software CDR setup

**Parameters:**
<BitrateEstimation> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Bit rate estimation" on page 1069

---

**CDR:SOFTware<m>:BITRate** <Bitrate>

Sets the quiescent frequency of the PLL. It corresponds to the data rate of the data stream from which the clock is to be recovered.

**Suffix:**
<m> 1..2
Number of the software CDR setup

**Parameters:**

<Bitrate>
Range: 100 to max {5 Gbps, <available instrument bandwidth}
Increment: 10
*RST: 1E+9
Default unit: bps

**Usage:** Asynchronous command

**Manual operation:** See "Nominal bit rate" on page 1069

---

**CDR:SOFTware<m>:ESLope** <Edge>

Selects the edges of the data stream that are used for the clock recovery.

- "Either": Both positive and negative edges are used
- "Positive / Negative": Only one edge direction is used. Use one of these settings if the other edge might deliver unreliable results.

**Suffix:**

<m>
1..2
Number of the software CDR setup

**Parameters:**

<Edge>
POSitive | NEGative | EITHer
*RST: EITHer

**Usage:** Asynchronous command

**Manual operation:** See "Data edges" on page 1068

---

**CDR:SOFTware<m>:SYNC** <InitialSync>

Defines the phase reference for the first clock edge.

**Suffix:**

<m>
1..2
Number of the software CDR setup

**Parameters:**

<InitialSync>
SAMPle | DATaedge

**SAMPle**
The first clock edge matches the first sample of the waveform at the left border of the display.

**DATaedge**
The first clock edge matches the first edge of the data signal.

*RST: DATaedge

**Usage:** Asynchronous command

**Manual operation:** See "Initial phase" on page 1070

**CDR:SOFTware<m>:RESults** <Results>

The PLL requires some time to synchronize to the phase of the data stream. You can select when the CDR algorithm returns clock edges.

**Suffix:**
<m> 1..2
Number of the software CDR setup

**Parameters:**
<Results> ALL | AISYnc

**ALL**
All clock edges are used.

**AISYnc**
The clock edges of the synchronization time are discarded; results are gathered after initial synchronization of the CDR. Thus, meaningful TIE measurement results can be obtained.

*RST: ALL

**Usage:** Asynchronous command

**Manual operation:** See "Selected results" on page 1070

**CDR:SOFTware<m>:PLL:ORDer** <PLLOrder>

Sets the order of the PLL: first or second order. PLL of higher order can compensate for more complex jitter behavior.

**Suffix:**
<m> 1..2
Number of the software CDR setup

**Parameters:**
<PLLOrder> FIRSt | SECond

*RST: SECond

**Usage:** Asynchronous command

**Manual operation:** See "Algorithm settings" on page 1069

**CDR:SOFTware<m>:PLL:BWIDth** <PLLBandwidth>

Sets the PLL bandwidth. It defines the part of the spectrum that the PLL can follow during synchronization. The PLL bandwidth is usually defined by the transmission standard.

**Suffix:**
<m> 1..2
Number of the software CDR setup

**Parameters:**

| | | |
|---|---|---|
| <PLLBandwidth> | Range: | Nominal BITRate * Range of RELBwidth (dependent range) |
| | Increment: | 10 |
| | *RST: | 599.88E+3 |
| | Default unit: | Hz |

**Usage:** Asynchronous command

**Manual operation:** See "Algorithm settings" on page 1069

---

**CDR:SOFTware<m>:PLL:RELBwidth** <PLLRelBw>

Sets the relative bandwidth, that is the ratio of the nominal bit rate to the PLL bandwidth.

**Suffix:**

<m> 1..2
Number of the software CDR setup

**Parameters:**

| | | |
|---|---|---|
| <PLLRelBw> | Range: | 10 to 5000 |
| | Increment: | 1 |
| | *RST: | 1667 |

**Usage:** Asynchronous command

**Manual operation:** See "Algorithm settings" on page 1069

---

**CDR:SOFTware<m>:PLL:DAMPing** <DampingFactor>

Sets the damping factor, which is only relevant for second order PLL.

**Suffix:**

<m> 1..2
Number of the software CDR setup

**Parameters:**

| | | |
|---|---|---|
| <DampingFactor> | Range: | 0.5 to 1 |
| | Increment: | 0.01 |
| | *RST: | 0.7 |

**Usage:** Asynchronous command

**Manual operation:** See "Algorithm settings" on page 1069

---

**CDR:SOFTware<m>:REVLockpll** <ReverseLockPll>

Enables the reverese lock PLL algorithm. The signal is locked on the first edge.

**Suffix:**

<m> 1..2

**Parameters:**

<ReverseLockPll> ON | OFF

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Reverse lock" on page 1070

---

**CDR:SOFTware<m>:SAMTime** <UntIntvlOffs>

Sets a sampling time for the clock signal, an offset for the clock edge in relation to the bit start. The clock edge sets the beginning of the unit interval. The sampling time is a number between 0 and 1. Value 0 sets the clock edge to the beginning of the bit period; value 0.5 sets the clock edge to the middle of the bit period.

**Suffix:**

<m> 1..2

**Parameters:**

<UntIntvlOffs>

Range: 0 to 1
Increment: 0.01
*RST: 0.5
Default unit: UI

**Usage:** Asynchronous command

**Manual operation:** See "Sampling time" on page 1069

### 24.22.4 Eye mask testing

The eye mask testing is available if the option R&S RTP-K12 is activated on the instrument.

The commands for eye mask testing are described in Chapter 24.14.6, "Eye mask testing", on page 1560.

## 24.23 Eye diagram analysis

The suffix EYE<m> selects the eye diagram to which the command applies. You can configure and analyze up to four eye diagrams in parallel.

There are two ways to generate eye diagrams on the R&S RTP:

- Standard eye diagrams, which are a superposition of repetitively sampled digital data.
- To use advanced eye diagrams, option R&S RTP-K136/K137 or R&S RTP-K91/K93 is required.

### 24.23.1 Basic setup

EYE<m>[:STATe] ..... 2413
EYE<m>:SOURce ..... 2413
EYE<m>:HORizontal[:UNIT] ..... 2414
EYE<m>:HORizontal:ABSolute:RANGe ..... 2414
EYE<m>:HORizontal:ABSolute:POSition ..... 2414
EYE<m>:HORizontal:UINTerval:RANGe ..... 2415
EYE<m>:HORizontal:UINTerval:POSition ..... 2415
EYE<m>:TIMReference:CLOCk:EDGE ..... 2415
EYE<m>:TIMReference:CLOCk:FACTor ..... 2415
EYE<m>:TIMReference:CLOCk:FREQuency ..... 2416
EYE<m>:TIMReference:CLOCk:FREStimation ..... 2416
EYE<m>:TIMReference:CLOCk:LEVel ..... 2416
EYE<m>:TIMReference:CLOCk:OFEStimation ..... 2417
EYE<m>:TIMReference:CLOCk:OFFSet ..... 2417
EYE<m>:TIMReference:CLOCk:SAMTime ..... 2417
EYE<m>:TIMReference:CLOCk:SOURce ..... 2418
EYE<m>:TIMReference:PROTocol:SELect ..... 2418
EYE<m>:TIMReference[:SOURce] ..... 2418
EYE<m>:MEASurement:MAXPrecision ..... 2419

**EYE<m>[:STATe]** <State>

Activates or deactivates the eye diagram.

**Suffix:**
<m> 1..4

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 926

**EYE<m>:SOURce** <MainSource>

Selects the waveform from which the eye diagram is generated (data source). For analog channels, peak detect, envelope, and average waveforms are not supported.

**Suffix:**
<m> 1..4

**Parameters:**
<MainSource> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 926

**EYE<m>:HORizontal[:UNIT]** <TimeUnits>

Sets the method to define the horizontal scale of the eye diagram.

**Suffix:**
<m> 1..4

**Parameters:**
<TimeUnits> ATIMe | UINTerval

**ATIMe**
Absolute time

**UINTerval**
Unit intervals. You can use unit interval, if the nominal frequency of the clock signal is known.

*RST: ATIMe

**Usage:** Asynchronous command

**Manual operation:** See "Time units" on page 928

**EYE<m>:HORizontal:ABSolute:RANGe** <HorizontalRange>

Sets the time range that is covered by the eye diagram.

**Suffix:**
<m> 1..4

**Parameters:**
<HorizontalRange> Range: 100E-12 to 100
Increment: 1E-9
*RST: 200E-9
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Range, Position" on page 928

**EYE<m>:HORizontal:ABSolute:POSition** <HorizPosition>

Sets the place of the zero point in the diagram, in seconds. The zero point is the alignment point on which the slice timestamps are superimposed.

**Suffix:**
<m> 1..4

**Parameters:**
<HorizPosition> Range: -100 to 100
Increment: 1E-9
*RST: 0
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Range, Position" on page 928

**EYE<m>:HORizontal:UINTerval:RANGe** <RangeUI>

Sets the number of unit intervals that are visible in the eye diagram.

**Suffix:**
<m> 1..4

**Parameters:**
<RangeUI> Range: 100E-12 to 100E+9
Increment: 0.01
*RST: 2
Default unit: UI

**Usage:** Asynchronous command

**Manual operation:** See "Range UI, Position UI" on page 928

**EYE<m>:HORizontal:UINTerval:POSition** <PositionUI>

Sets the zero point in the diagram, in unit intervals. The zero point is the alignment point on which the segment timestamps are superimposed.

**Suffix:**
<m> 1..4

**Parameters:**
<PositionUI> Range: -100E+9 to 100E+9
Increment: 0.01
*RST: 0
Default unit: UI

**Usage:** Asynchronous command

**Manual operation:** See "Range UI, Position UI" on page 928

**EYE<m>:TIMReference:CLOCk:EDGE** <EdgeType>

The clock edge sets the point in time when the state of the data signal is checked.

**Suffix:**
<m> 1..4

**Parameters:**
<EdgeType> POSitive | NEGative | EITHer
*RST: EITHer

**Usage:** Asynchronous command

**Manual operation:** See "Clock edge" on page 930

**EYE<m>:TIMReference:CLOCk:FACTor** <Factor>

Sets a clock multiplier factor.

Available if `EYE<m>:TIMReference[:SOURce]` is set to `CLOCk`.

**Suffix:**
<m> 1..4

**Parameters:**
<Factor> Range: 1 to 100
Increment: 1
*RST: 1

**Usage:** Asynchronous command

**Manual operation:** See "Clock multiplier" on page 930

---

**EYE<m>:TIMReference:CLOCk:FREQuency** <Frequency>

Sets the clock frequency for the explicit clock signal.

Available if `EYE<m>:TIMReference[:SOURce]` is set to `CLOCk`.

**Suffix:**
<m> 1..4

**Parameters:**
<Frequency> Range: 1000 to 10E+9
Increment: 10
*RST: 1E+9
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Clock frequency" on page 929

---

**EYE<m>:TIMReference:CLOCk:FREStimation**

Executes an estimation of the signal for the clock frequency.

Available if `EYE<m>:TIMReference[:SOURce]` is set to `CLOCk`.

**Suffix:**
<m> 1..4

**Usage:** Event

**Manual operation:** See "Estimate clock frequency" on page 929

---

**EYE<m>:TIMReference:CLOCk:LEVel** <Level>

Sets the vertical reference level of the timing reference source.

Available if `EYE<m>:TIMReference[:SOURce]` is set to `CLOCk`.

**Suffix:**
<m> 1..4

**Parameters:**
<Level> UPPer | MIDDle | LOWer
*RST: MIDDle

**Usage:** Asynchronous command

**Manual operation:** See "Vertical reference level" on page 929

---

**EYE<m>:TIMReference:CLOCk:OFEStimation**

Executes an estimation of the signal for the clock offset.

Available if `EYE<m>:TIMReference[:SOURce]` is set to `CLOCk`.

**Suffix:**
<m> 1..4

**Usage:** Event

**Manual operation:** See "Estimate clock offset" on page 930

---

**EYE<m>:TIMReference:CLOCk:OFFSet** <Offset>

Sets a clock offset for the explicit clock signal.

Available if `EYE<m>:TIMReference[:SOURce]` is set to `CLOCk`.

**Suffix:**
<m> 1..4

**Parameters:**
<Offset>
Range: -1000 to 1000
Increment: 0.01
*RST: 0
Default unit: UI

**Usage:** Asynchronous command

**Manual operation:** See "Clock offset" on page 930

---

**EYE<m>:TIMReference:CLOCk:SAMTime** <SamplingTime>

Sets a sampling time for the explicit clock signal, an offset for the clock edge in relation to the bit start. The clock edge sets the beginning of the unit interval. The sampling time is a number between 0 and 1. Value 0 sets the clock edge to the beginning of the bit period; value 0.5 sets the clock edge to the middle of the bit period.

Available if `EYE<m>:TIMReference[:SOURce]` is set to `CLOCk`.

**Suffix:**
<m> 1..4

**Parameters:**
<SamplingTime>
Range: 0 to 1
Increment: 0.01
*RST: 0.5
Default unit: UI

**Usage:** Asynchronous command

**Manual operation:** See "Sampling time" on page 930

---

**EYE<m>:TIMReference:CLOCk:SOURce** <SourceType>

Selects the source of the clock channel, if `EYE<m>:TIMReference[:SOURce]` is set to `CLOCk`.

**Suffix:**
<m> 1..4

**Parameters:**
<SourceType> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15

**Usage:** Asynchronous command

**Manual operation:** See "Clock" on page 926

---

**EYE<m>:TIMReference:PROTocol:SELect** <SerialBusIndex>

Selects the active bus for the eye analysis, if `EYE<m>:TIMReference[:SOURce]` is set to `PROTocol`.

**Suffix:**
<m> 1..4

**Parameters:**
<SerialBusIndex> SBUS1 | SBUS2 | SBUS3 | SBUS4
*RST: SBUS1

**Usage:** Asynchronous command

**Manual operation:** See "Protocol" on page 927

---

**EYE<m>:TIMReference[:SOURce]** <ReferenceSource>

Selects the source of the reference signal used for the eye diagram.

**Suffix:**
<m> 1..4

**Parameters:**
<ReferenceSource> CLOCk | CDR | PROTocol
*RST: CLOCk

**Usage:** Asynchronous command

**Manual operation:** See "Clock / CDR / Protocol" on page 926

**EYE<m>:MEASurement:MAXPrecision** <EnabClassifier>

If this function is enabled, the rising and falling transitions for the eye measurmens "Eye rise time", "Eye fall time" and "Duty cycle distortion" are separated.

**Suffix:**
<m> 1..4

**Parameters:**
<EnabClassifier> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Maximum precision" on page 937

### 24.23.2 Display

**EYE<m>:MSLices** <MaxSliceCount>

Sets the number of waveform slices for a single acquisition.

**Suffix:**
<m> 1..4

**Parameters:**
<MaxSliceCount> Range: 1 to 2500000
Increment: 1
*RST: 2000

**Usage:** Asynchronous command

**Manual operation:** See "Slices per acquisition" on page 932

**EYE<m>:DISPlay:MODE** <EyestripeMode>

Defines when the eye stripe is shown: always, never, or during mask tests.

**Suffix:**
<m> 1..4

**Parameters:**
<EyestripeMode> ALWays | MTONly | NEVer
MTONly: Mask test only
*RST: MTONly

**Usage:** Asynchronous command

**Manual operation:** See "Eye stripe display" on page 932

**EYE<m>:DISPlay:PERSistence:INFinite** <State>

If persistence is enabled (`EYE<m>:DISPlay:PERSistence:STATe`), each new data point in the diagram area remains on the screen infinitely until this command is set to `OFF`.

**Suffix:**
<m> 1..4

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Infinite persistence" on page 932

**EYE<m>:DISPlay:PERSistence:RESet**

Resets the display, removing persistent values.

**Suffix:**
<m> 1..4

**Usage:** Event
Asynchronous command

**Manual operation:** See "Reset" on page 932

**EYE<m>:DISPlay:PERSistence:STATe** <State>

If enabled, each new data point in the diagram area remains on the screen for the duration defined using `EYE<m>:DISPlay:PERSistence:TIME`, or as long as `EYE<m>:DISPlay:PERSistence:INFinite` is enabled.

If disabled, the signal value is only displayed as long as it actually occurs.

**Suffix:**
<m> 1..4

**Parameters:**
<State> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 932

**EYE<m>:DISPlay:PERSistence:TIME** <Time>

If persistence is enabled (`EYE<m>:DISPlay:PERSistence:STATe`), each new data point in the diagram area remains on the screen for the duration defined here.

**Suffix:**
<m> 1..4

**Parameters:**
<Time> Range: 0.05 to 50
Increment: 0.05
*RST: 0.05
Default unit: s

**Usage:** Asynchronous command

**Manual operation:** See "Time" on page 932

### 24.23.3 Qualify: gate settings

EYE<m>:QUALify:GATE[:STATe] 2421
EYE<m>:QUALify:GATE:COUPling 2421
EYE<m>:QUALify:GATE:MODE 2422
EYE<m>:QUALify:GATE:ABSolute:STARt 2422
EYE<m>:QUALify:GATE:ABSolute:STOP 2422
EYE<m>:QUALify:GATE:RELative:STARt 2422
EYE<m>:QUALify:GATE:RELative:STOP 2422
EYE<m>:QUALify:GATE:CURSor 2423
EYE<m>:QUALify:GATE:ZDIagram 2423

**EYE<m>:QUALify:GATE[:STATe]** <State>

Enables the gate.

**Suffix:**
<m> 1..4

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Gate" on page 933

**EYE<m>:QUALify:GATE:COUPling** <CouplingMode>

Sets the gate coupling. To use cursor lines or zoom areas as gate limits, the cursor or zoom must be defined before.

**Suffix:**
<m> 1..4

**Parameters:**
<CouplingMode> NONE | CURSor | ZOOM
*RST: NONE

**Usage:** Asynchronous command

**Manual operation:** See "Gate" on page 933

---

**EYE<m>:QUALify:GATE:MODE** <Mode>

Defines whether the gate limit is defined in absolute or relative values when no gate coupling is set.

**Suffix:**
<m> 1..4

**Parameters:**
<Mode> ABS | REL
*RST: ABS

**Usage:** Asynchronous command

**Manual operation:** See "Gate" on page 933

---

**EYE<m>:QUALify:GATE:ABSolute:STARt** <Start>
**EYE<m>:QUALify:GATE:ABSolute:STOP** <Stop>

Set the absolute time values for gate start and gate end.

**Suffix:**
<m> 1..4

**Parameters:**
<Start>, <Stop> Range: -100E+24 to 100E+24
Increment: 0.01
*RST: 0.01

**Example:** Gate definition for eye diagram 2 in absolute time values:

```
EYE2:QUALify:GATE:COUPling NONE
EYE2:QUALify:GATE:MODE ABS
EYE2:QUALify:GATE:ABSolute:STARt -0.0002
EYE2:QUALify:GATE:ABSolute:STOP 0.0002
```

**Usage:** Asynchronous command

**Manual operation:** See "Gate" on page 933

---

**EYE<m>:QUALify:GATE:RELative:STARt** <RelativeStart>
**EYE<m>:QUALify:GATE:RELative:STOP** <RelativeStop>

Set the relative values for gate start and gate end.

**Suffix:**
<m> 1..4

**Parameters:**
<RelativeStart>, <RelativeStop> Range: 0 to 100
Increment: 0.1
*RST: 100
Default unit: %

**Example:** Gate definition for eye diagram 2 in relative values:

```
EYE2:QUALify:GATE:COUPling NONE
EYE2:QUALify:GATE:MODE REL
EYE2:QUALify:GATE:RELative:STARt 10
EYE2:QUALify:GATE:RElative:STOP 90
```

**Usage:** Asynchronous command

**Manual operation:** See "Gate" on page 933

---

**EYE<m>:QUALify:GATE:CURSor** <Cursorset>

Selects the cursor set to be used for eye gating. The gate area is defined by the cursor lines.

**Suffix:**
<m> 1..4

**Parameters:**
<Cursorset> CURSOR1 | CURSor1 | CURSOR2 | CURSor2 | CURSOR3 | CURSor3 | CURSOR4 | CURSor4

CURSOR1 = CURSor1, CURSOR2 = CURSor2, CURSOR3 = CURSor3, CURSOR4 = CURSor4

*RST: CURSOR1

**Example:** Gate definition for eye diagram 2 with cursor coupling:

```
EYE2:QUALify:GATE:COUPling CURSor
EYE2:QUALify:GATE:CURSor CURSor1
```

**Usage:** Asynchronous command

**Manual operation:** See "Gate" on page 933

---

**EYE<m>:QUALify:GATE:ZDIagram** <ZoomDiagram>

Selects the zoom diagram to be used for eye gating. The zoom area is is used as gate.

**Suffix:**
<m> 1..4

**Parameters:**
<ZoomDiagram> String with the name of the zoom diagram

**Example:** Gate definition for eye diagram 2 with zoom coupling:

```
EYE2:QUALify:GATE:COUPling ZOOM
EYE2:QUALify:GATE:ZDIagram "Zoom1"
```

**Usage:** Asynchronous command

**Manual operation:** See "Gate" on page 933

### 24.23.4 Qualify: signal settings

EYE<m>:QUALify:SIGNal[:STATe] ..... 2424
EYE<m>:QUALify:SIGNal:SOURce ..... 2424
EYE<m>:QUALify:SIGNal:CONDition ..... 2424
EYE<m>:QUALify:SIGNal:LSELect ..... 2425

**EYE<m>:QUALify:SIGNal[:STATe]** <State>

Enables the waveform-based qualification for the eye generation.

**Suffix:**
<m> 1..4

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "State" on page 934

**EYE<m>:QUALify:SIGNal:SOURce** <Source>

Selects the waveform to be compared.

**Suffix:**
<m> 1..4

**Parameters:**
<Source> C1W1 | C2W1 | C3W1 | C4W1 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | R1 | R2 | R3 | R4

**Usage:** Asynchronous command

**Manual operation:** See "Source" on page 934

**EYE<m>:QUALify:SIGNal:CONDition** <Condition>

Selects the comparison condition for the the waveform level.

**Suffix:**
<m> 1..4

**Parameters:**
<Condition> LTHan | GTHan | INRange | OORange
*RST: GTHan

**Usage:** Asynchronous command

**Manual operation:** See "Condition" on page 934

**EYE<m>:QUALify:SIGNal:LSELect** <LevelSelection>

Selects the threshold. You can use the lower, middle, or upper reference level. These levels are the reference levels that are defined for measurements.

**Suffix:**
<m> 1..4

**Parameters:**
<LevelSelection> UPPer | MIDDle | LOWer
*RST: MIDDle

**Example:** The slices of the channel 1 signal only contribute to the eye generation if the voltage level of the signal on channel 2 is higher than the upper reference level.

```
EYE:SOURce C1W1
EYE:QUALify:SIGNal:SOURce C2W1
EYE:QUALify:SIGNal:CONDition GTHan
EYE:QUALify:SIGNal:LSELect UPPer
EYE:QUALify:SIGNal ON
```

**Usage:** Asynchronous command

**Manual operation:** See "Vertical reference level" on page 934

### 24.23.5 Filter

EYE<m>:FILTer:STATe ..... 2425
EYE<m>:FILTer:BPATtern:MODE ..... 2426
EYE<m>:FILTer:BPATtern:PREFix<1..8> ..... 2426
EYE<m>:FILTer:BPATtern:CENTer ..... 2426
EYE<m>:FILTer:BPATtern:SUFFix<1..5> ..... 2426
EYE<m>:FILTer:BPATtern:SLEFt ..... 2426
EYE<m>:FILTer:BPATtern:SRIGht ..... 2427
EYE<m>:FILTer:BPATtern:RESet ..... 2427
EYE<m>:FILTer:DDR:FTYPe ..... 2427
EYE<m>:FILTer:DDR:LENerror ..... 2427

**EYE<m>:FILTer:STATe** <State>

Applies the filter settings to the data signal.

**Suffix:**
<m> 1..4

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 935

**EYE<m>:FILTer:BPATtern:MODE** <Mode>

Selects which bits contribute to the eye generation.

**Suffix:**
<m> 1..4

**Parameters:**
<Mode> ABITs | LTRansition | CLEVel | BPATtern

**ABITs**
All bits contribute to the eye diagram, same as `EYE<m>:FILTer:STATe` is OFF.

**LTRansition**
Level transition, only bits after a signal transition contribute (0 to 1, or 1 to 0).

**CLEVel**
Constant level, only bits without a preceding signal transition contribute ( 0 to 0, or 1 to 1).

**BPATtern**
Only bits contribute that match the defined pattern before and after the center eye.

*RST: ABITs

**Usage:** Asynchronous command

**Manual operation:** See "Mode" on page 935

**EYE<m>:FILTer:BPATtern:PREFix<1..8>** <Bit>
**EYE<m>:FILTer:BPATtern:CENTer** <BitValue>
**EYE<m>:FILTer:BPATtern:SUFFix<1..5>** <Bit>

Set the values of the indicated prefix, center, and suffix bits.

**Suffix:**
<m> 1..4

**Parameters:**
<Bit> ONE | ZERO | DC

DC: don't care

**Manual operation:** See "Pattern" on page 935

**EYE<m>:FILTer:BPATtern:SLEFt**

Shifts the pattern one bit to the left.

**Suffix:**
<m> 1..4

**Usage:** Event
Asynchronous command

**Manual operation:** See "Pattern control" on page 936

**EYE<m>:FILTer:BPATtern:SRIGht**

Shifts the pattern one bit to the right.

**Suffix:**
<m> 1..4

**Usage:** Event
Asynchronous command

**Manual operation:** See "Pattern control" on page 936

**EYE<m>:FILTer:BPATtern:RESet**

Resets the bit sequence filter to the default (DC for all bits).

**Suffix:**
<m> 1..4

**Usage:** Event
Asynchronous command

**Manual operation:** See "Pattern control" on page 936

**EYE<m>:FILTer:DDR:FTYPe** <CommTyp>

Sets which frame types contribute to the eye diagram.

**Suffix:**
<m> 1..4

**Parameters:**
<CommTyp> ANY | ANYC | READ | WRITe | RDNC | WRNC
ANYC: any consecutive
RDNC: read non-consecutive
WRNC: write non-consecutive
*RST: ANY

**Usage:** Asynchronous command

**Manual operation:** See "Frame Type" on page 936

**EYE<m>:FILTer:DDR:LENerror** <CommTyp>

If enabled, the length errors contribute to the eye diagram.

**Suffix:**
<m> 1..4

**Parameters:**
<CommTyp> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Length Error" on page 936

### 24.23.6 Results of eye

EYE<m>:RESult:SLICe<n>:COUNt?....2428
EYE<m>:RESult:SLICe<n>:LPOSition?....2428
EYE<m>:RESult:SLICe<n>[:POSition]?....2428
EYE<m>:RESult:TACQs?....2429
EYE<m>:RESult:TSLices?....2429

**EYE<m>:RESult:SLICe<n>:COUNt?**

Returns the number of filtered bits.

**Suffix:**
<m> 1..4

<n> *
Not relevant

**Return values:**
<Count>

**Usage:** Query only

**EYE<m>:RESult:SLICe<n>:LPOSition?**

Returns the position of the last slice in the acquistion.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<LastPosition>

**Usage:** Query only

**EYE<m>:RESult:SLICe<n>[:POSition]?**

Returns the position of the filtered bit in the acquisition.

**Suffix:**
<m> 1..4

<n> *

**Return values:**
<Position> Range: -100 to 100
Increment: 1E-3
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**EYE<m>:RESult:TACQs?**

Returns the total number of acquisitions accumulated into the eye.

**Suffix:**
<m> 1..4

**Return values:**
<Acquisitions> Range: 0 to 1000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**EYE<m>:RESult:TSLices?**

Returns the total number of slices accumulated into the eye.

**Suffix:**
<m> 1..4

**Return values:**
<Slices> Range: 0 to 1000000000
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

## 24.23.7 Results of mask tests on eye diagrams

MTESt:RESult:COUNt:SLICes?....2429
MTESt:RESult:COUNt:FSLices?....2430
MTESt:RESult:VIOLation<m>:COUNt?....2430
MTESt:RESult:VIOLation<m>[:SLICe]?....2430
MTESt:RESult:VIOLation<m>:STARt?....2431
MTESt:RESult:VIOLation<m>:STOP?....2431
EYE<m>:ZCOupling....2431
EYE<m>:ZDIagram....2431

---

**MTESt:RESult:COUNt:SLICes?** <MaskTestName>

Returns the number of slices that contributed to the eye diagram.

**Query parameters:**
<MaskTestName> String parameter

**Return values:**
<SlicesCompleted> Range: 0 to 100E+24
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**MTESt:RESult:COUNt:FSLices?** <MaskTestName>

Returns the number of slices that violated the mask.

**Query parameters:**
<MaskTestName> String parameter

**Return values:**
<SliceHits> Range: 0 to 100E+24
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**MTESt:RESult:VIOLation<m>:COUNt?** <MaskTestName>

**Suffix:**
<m> *
Violation index, corresponds to the number in the "Violation" column of mask test result.

**Query parameters:**
<MaskTestName>

**Return values:**
<Count>

**Usage:** Query only

---

**MTESt:RESult:VIOLation<m>[:SLICe]?** <MaskTestName>

**Suffix:**
<m> *
Violation index, corresponds to the number in the "Violation" column of mask test result.

**Query parameters:**
<MaskTestName>

**Return values:**
<Slice> Range: 0 to 4294967295
Increment: 1
*RST: 0

**Usage:** Query only
Asynchronous command

---

**MTESt:RESult:VIOLation<m>:STARt?** <MaskTestName>
**MTESt:RESult:VIOLation<m>:STOP?** <MaskTestName>

Return the start and stop time of the slice with the indicated violation.

**Suffix:**
<m> *
Violation index, corresponds to the number in the "Violation" column of mask test result.

**Query parameters:**
<MaskTestName>

**Return values:**
<Time> Range: -100 to 100
Increment: 1E-3
*RST: 0
Default unit: s

**Usage:** Query only
Asynchronous command

---

**EYE<m>:ZCOupling** <ZoomCoupling>

If enabled, the mask eye area is set to the limits of a zoom area.

The zoom diagram is selected using `EYE<m>:ZDIagram`.

**Suffix:**
<m> 1..4

**Parameters:**
<ZoomCoupling> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

---

**EYE<m>:ZDIagram** <ZoomDiagram>

Selects the zoom to which the eye mask area is set if `EYE<m>:ZCOupling` is enabled.

**Suffix:**
<m> 1..4

**Parameters:**
<ZoomDiagram> String with the name of the diagram on which the zoom is based

**Usage:** Asynchronous command

## 24.24 Deembedding (option R&S RTP-K121/-K122)

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

- Measurement setup ..... 2432
- Save and load the deembedding setup ..... 2435
- Components ..... 2436
- Responses ..... 2443

### 24.24.1 Measurement setup

DEEMbedding<m>[:STATe] ..... 2432
DEEMbedding<m>:ADD ..... 2432
DEEMbedding<m>:REMove ..... 2433
DEEMbedding<m>:CCOunt? ..... 2433
DEEMbedding<m>:REALtime ..... 2434
DEEMbedding<m>:BANDwidth ..... 2434
DEEMbedding<m>:CMPDelay ..... 2435
DEEMbedding<m>:PRESet ..... 2435

**DEEMbedding<m>[:STATe]** <State>

Activates the deembedding - the correction of parasitic effects of the measurement setup on the measured signal.

**Suffix:**

<m> 1..4
Selects the channel that is configured for deembedding.

**Parameters:**

<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 1089

**DEEMbedding<m>:ADD** <Index>, [<Type>]

Adds a component to the measurement setup.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Setting parameters:**
<Index> Defines the place of the component in the measurement path. The DUT is always the first component (index = 1), the probe or input is the last component.

<Type> PROBe | FIXTure | CABLe | DUT | ADAPter | INPut | OTHer | RTZA10 | RTZA16 | RTZA17 | PP | PC | EXTFrontend

**RTZA10 | RTZA16 | RTZA17**
SMA adapters and cable offered by Rohde & Schwarz

**PP | PC**
Proven probe, proven cable

**EXTFrontend**
External frontend, requires option R&S RTP-K553.

**Example:** See Chapter 24.3.6.1, "Adding cables to the measurement setup", on page 1186.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Meas setup definition" on page 1090

---

**DEEMbedding<m>:REMove** <Index>

Deletes the component with the indicated index from the measurement setup.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Setting parameters:**
<Index> Defines the place of the component in the measurement path. The DUT is always the first component (index = 1), the probe or input is the last component.

**Usage:** Setting only
Asynchronous command

**Manual operation:** See "Meas setup definition" on page 1090

---

**DEEMbedding<m>:CCOunt?**

Returns the total number of components in the measurement setup, including disabled components.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Return values:**
<Value> Integer value, number of components

**Usage:** Query only

**Manual operation:** See "Meas setup definition" on page 1090

---

**DEEMbedding<m>:REALtime** <IsRealTime>

Requires realtime extension option R&S RTP-K122.

If ON, the deembedding calculation is done by the hardware in real time. You can trigger on the corrected signal, and the acquisition rate remains unchanged.

If OFF, the deembedding calculation is done by software. This method is slower, therefore, triggering on the corrected signal is not possible.

**Suffix:**
<m> 1..4

**Parameters:**
<IsRealTime> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Realtime deembedding" on page 1089

---

**DEEMbedding<m>:BANDwidth** <Bandwidth>

Sets the maximum bandwidth until which the signal is corrected. This maximum value is the minimum bandwidth value of probe, probe tip and oscilloscope bandwidth, and can not be higher than the highest frequency in a used S-parameter file.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Parameters:**
<Bandwidth> Bandwidth limit for correction.
Range: 1E+9 to 16E+9
Increment: 10
*RST: 4E+9
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Effective bandwidth" on page 1089

**DEEMbedding<m>:CMPDelay** <DelayCompens>

Enables the compensation for the group delay of the complete measurement setup. If enabled, the calibration time reference point is the start point of the measurement setup. If disabled, the calibration time reference point is the oscilloscope's channel input.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Parameters:**
<DelayCompens> ON | OFF
*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "Delay compensation" on page 1089

**DEEMbedding<m>:PRESet**

Sets the complete deembedding configuration to the default values. Use this command always before setting up the deembedding, because *RST does not affect the deembedding settings.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Example:** See Chapter 24.3.6.1, "Adding cables to the measurement setup", on page 1186.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Preset Deembedding" on page 1090

## 24.24.2 Save and load the deembedding setup

DEEMbedding<m>:NAME....2435
DEEMbedding<m>:SAVE....2436
DEEMbedding<m>:OPEN....2436
DEEMbedding<m>:DELete....2436

**DEEMbedding<m>:NAME** <Path>

Sets the file name, file format and path of the deembedding setup file. The file format is always `.xml`.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Parameters:**
<Path> String with path and file name with extension `.xml`

**Usage:** Asynchronous command

**Manual operation:** See "Save / recall deembedding settings" on page 1090

---

**DEEMbedding<m>:SAVE**

Saves the deembedding configuration to the file specified with `DEEMbedding<m>:NAME`.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Usage:** Event

**Manual operation:** See "Save / recall deembedding settings" on page 1090

---

**DEEMbedding<m>:OPEN**

Opens and loads the deembedding configuration from the file specified with `DEEMbedding<m>:NAME`.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Usage:** Event

**Manual operation:** See "Save / recall deembedding settings" on page 1090

---

**DEEMbedding<m>:DELete**

Deletes the deembedding configuration file specified with `DEEMbedding<m>:NAME`.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Usage:** Event

**Manual operation:** See "Save / recall deembedding settings" on page 1090

### 24.24.3 Components

- General settings....2437
- DUT settings....2438
- R&S RT-ZM probe settings....2440
- Cable, adapter, fixture and custom settings....2441

#### 24.24.3.1 General settings

**DEEMbedding<m>:COMPonent<n>[:STATe]** <IsEnabled>

Enables or disables the indicated component for deembedding.

The first and the last component are always enabled and cannot be disabled.

**Suffix:**

<m> 1..4
Selects the channel that is configured for deembedding.

<n> 2..n-1
Index of the component in the measurement setup. The DUT has index 1, the probe has index n.

**Parameters:**

<IsEnabled> ON | OFF

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 1091

**DEEMbedding<m>:COMPonent<n>:TYPE** <Type>

Sets the type of the indicated component. The first component is always DUT, the last component is PROBe or INPut.

**Suffix:**

<m> 1..4
Selects the channel that is configured for deembedding.

<n> 1..n
Index of the component in the measurement setup. The DUT has index 1, the probe has index n.

**Parameters:**

<Type> PROBe | FIXTure | CABLe | DUT | ADAPter | INPut | OTHer | RTZA10 | RTZA16 | RTZA17 | PP | PC | EXTFrontend

**RTZA10 | RTZA16 | RTZA17**
SMA adapters and cable offered by Rohde & Schwarz

**PP | PC**
Proven probe, proven cable

**EXTFrontend**
External frontend, requires option R&S RTP-K553.

*RST: CABLe

**Usage:** Asynchronous command

**Manual operation:** See "Component" on page 1091

**DEEMbedding<m>:COMPonent<n>:NAME** <Name>

Sets a name for the indicated component that helps you identifying it.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

<n> 1..n
Index of the component in the measurement setup. The DUT has index 1, the probe has index n.

**Parameters:**
<Name> String parameter, name of the component

**Usage:** Asynchronous command

**Manual operation:** See "Name" on page 1091

**DEEMbedding<m>:COMPonent<n>:PCOunt?**

Returns the number of all ports of the selected component (input ports and output ports).

The number of ports depends on the attached probe. If a modular probe R&S RT-ZM or differential probe R&S RT-ZD is used, the number of ports is 4. Otherwise, 2 ports are available.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

<n> 1..n
Index of the component in the measurement setup. The DUT has index 1, the probe has index n.

**Return values:**
<TotalPorts> ONE | TWO | THRee | FOUR
*RST: TWO

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Total ports" on page 1091

### 24.24.3.2 DUT settings

DEEMbedding<m>:COMPonent<n>:MODE ..... 2438
DEEMbedding<m>:COMPonent<n>:LOAD:REMProbeload ..... 2439
DEEMbedding<m>:COMPonent<n>:LOAD:IMPedance? ..... 2439

**DEEMbedding<m>:COMPonent<n>:MODE** <Mode>

Sets the probing mode.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

<n> 1, only for available DUT

**Parameters:**
<Mode> PROBing | TERMinating

**PROBing**
For measurements with a high-ohmic probe and measure on an existing line in parallel to the load.

**TERMinating**
For measurements if the measuring equipment is the load of the line.

*RST: PROBing

**Usage:** Asynchronous command

**Manual operation:** See "Probing, Terminating" on page 161

---

**DEEMbedding<m>:COMPonent<n>:LOAD:REMProbeload** <Value>

Removes the loading of the probe if `DEEMbedding<m>:COMPonent<n>:MODE` is set to `PROBing`.

If the probe loading is removed, you see the signal that would be at the measurement point if the probe's input impedance is ideal (infinite impedance).

Without removing the loading, you see the real signal at the measurement point, including the probe loading.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

<n> 1, only for available DUT

**Parameters:**
<Value> OFF | ON

**Usage:** Asynchronous command

**Manual operation:** See "Remove probe load" on page 161

---

**DEEMbedding<m>:COMPonent<n>:LOAD:IMPedance?** <Value>

Shows the source impedance of the DUT. If an R&S RT-ZM probe is connected, the value depends on the selected probe mode: common mode, differential, or single-ended measurement.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

<n> 1, only for available DUT

**Return values:**
<Value> Numeric value

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Source impedance" on page 161

### 24.24.3.3 R&S RT-ZM probe settings

This chapter lists the commands that correspond to the "Deembedding" > "RT-ZMxx" dialog box. More commands are provided in Chapter 24.8.7.6, "Modular probes, deembedding", on page 1271.

PROBe<m>:DEEMbedding:TIPModule[:SELect] ..... 2440
PROBe<m>:DEEMbedding:TIPModule:ZMA<n>:SUBModule ..... 2440
PROBe<m>:DEEMbedding:BANDwidth ..... 2441

---

**PROBe<m>:DEEMbedding:TIPModule[:SELect]** <ProbeTip>

Selects the tip module that is used for measurement.

**Suffix:**
<m> 1..4
Selects the channel to which the probe is connected.

**Parameters:**
<ProbeTip> NONE | ZMA10 | ZMA12 | ZMA14 | ZMA15 | ZMA30 | ZMA40 | ZMA50 | CUST

*RST: ZMA10

**Usage:** Asynchronous command

**Manual operation:** See "Select probe tip module, Pin spacing" on page 160

---

**PROBe<m>:DEEMbedding:TIPModule:ZMA<n>:SUBModule** <Value>

Defines additional settings for some tip modules:

For the browser module R&S RT-ZMA30, measure the space between the pins, and select the appropriate value.

If R&S RT-ZMA40 is selected, choose the used submodule: semi-rigid cables or none.

**Suffix:**
<m> 1..4
Selects the channel to which the probe is connected.

<n> 30 (R&S RT-ZMA30) or 40 (R&S RT-ZMA40)

**Parameters:**
<Value> D16 | D25 | D45 | D75 | NONE | SRCS

**For R&S RT-ZMA30:**
D16 = 1.6 mm

D25 = 2.5 mm
D45 = 4.5 mm
D75 = 7.5 mm
**For R&S RT-ZMA40:**
NONE = no submodule
SRCS = semi-rigid cables

**Usage:** Asynchronous command

**Manual operation:** See "Select probe tip module, Pin spacing" on page 160

---

**PROBe<m>:DEEMbedding:BANDwidth** <ProbeBandwidth>

Sets the maximum bandwidth until which the signal is corrected. This maximum value is the minimum bandwidth value of probe, probe tip and oscilloscope bandwidth, and can not be higher than the highest frequency in a used S-parameter file. Consider that most tip modules support the full bandwidth of the probe amplifier, but some tip modules have limited bandwidth.

**Suffix:**
<m> 1..4
Selects the channel to which the probe is connected.

**Parameters:**
<ProbeBandwidth>
Range: 10000 to 16E+9
Increment: 10
*RST: 4E+9
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Effective bandwidth" on page 161

### 24.24.3.4 Cable, adapter, fixture and custom settings

DEEMbedding<m>:COMPonent<n>:IDENtical ..... 2441
DEEMbedding<m>:COMPonent<n>:IPONe ..... 2442
DEEMbedding<m>:COMPonent<n>:IPTWo ..... 2442
DEEMbedding<m>:COMPonent<n>:OPONe ..... 2442
DEEMbedding<m>:COMPonent<n>:OPTWo ..... 2442
DEEMbedding<m>:COMPonent<n>:SPONe ..... 2442
DEEMbedding<m>:COMPonent<n>:SPTWo ..... 2442

---

**DEEMbedding<m>:COMPonent<n>:IDENtical** <MatchedPaths>

The setting is relevant if the measurement setup is a 4-port, and the component is a 2-port. Usually, these components are cables and adapters, which are described by *.s2p files. In case of a 4-port setup, 2 cables or 2 adapters are used. If they are identical, for example, when a matched-pair cable is used, they can be described by the same file. In this case, set the command to ON.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

<n> 1..*
Selects the component. The DUT has suffix 1.

**Parameters:**
<MatchedPaths> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Identical Paths" on page 1096

---

**DEEMbedding<m>:COMPonent<n>:IPONe** <InputPort1>
**DEEMbedding<m>:COMPonent<n>:IPTWo** <InputPort2>
**DEEMbedding<m>:COMPonent<n>:OPONe** <OutputPort1>
**DEEMbedding<m>:COMPonent<n>:OPTWo** <OutputPort2>

Assign the ports to the input and output of the component according to the data in the Touchstone file. This information is not directly written in the file. You need background information how the S-parameters were determined, i.e. which port was input and which port was output.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

<n> 1..*
Selects the component. The DUT has suffix 1.

**Parameters:**
<InputPort1>
<InputPort2>
<OutputPort1>
<OutputPort2>
ONE | TWO | THRee | FOUR
Port number that is assigned to the input or output.
*RST: ONE for input 1, TWO for output 1, THRee for input 2, FOUR for output 4

**Usage:** Asynchronous command

**Manual operation:** See "Input 1, Input 2, Output 1, Output 2" on page 1096

---

**DEEMbedding<m>:COMPonent<n>:SPONe** <SParamFilePath>
**DEEMbedding<m>:COMPonent<n>:SPTWo** <SParamFilePath>

Set the file name, file format and path of the S-parameter file, and load the S-parameters.

See also: "S-Parameters, Filetype" on page 1096.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

<n> 1..*
Selects the component. The DUT has suffix 1.

**Parameters:**
<SParamFilePath> String with path and file name with extension `s2p` (2-port component) or `s4p` (4-port component).

**Example:** See Chapter 24.3.6.1, "Adding cables to the measurement setup", on page 1186.

**Usage:** Asynchronous command

**Manual operation:** See "S-Parameters, Filetype" on page 1096

### 24.24.4 Responses

**DEEMbedding<m>:CADC?**

Maximum attenuation or gain of the measurement setup before deembedding at DC. This value is the expected basic attenuation or gain, which is corrected by deembedding.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Return values:**
ValueAtDC Range: -1E+9 to 1E+9
Increment: 0.01
*RST: 0
Default unit: dB

**Usage:** Query only

**Manual operation:** See "$Att_{DC}$, $Gain_{DC}$" on page 1098

**DEEMbedding<m>:MATTenuation?**

Maximum attenuation of the measurement setup before deembedding, with reference to a frequency response normalized at DC to 0 dB. The deembedding filter must amplify the amplitude response by the "$Att_{max}$" value, and thus also increases the noise by this value. The value also helps to decide whether deembedding is useful for the defined effective bandwidth.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Return values:**
<MaxFilterAtt> Range: -1E+9 to 1E+9
Increment: 0.01
*RST: 0
Default unit: dB

**Usage:** Query only

**Manual operation:** See "$Att_{max}$" on page 1098

---

**DEEMbedding<m>:MGAin?**

Maximum gain of the measurement setup before deembedding, with reference to a frequency response normalized at DC to 0 dB. The deembedding filter must attenuate the amplitude response by this value. The value helps also to avoid an overload of the ADC.

**Suffix:**
<m> 1..4
Selects the channel that is configured for deembedding.

**Return values:**
<MaxFilterGn> Range: -1E+9 to 1E+9
Increment: 0.01
*RST: 0
Default unit: dB

**Usage:** Query only

**Manual operation:** See "$Gain_{max}$" on page 1098

## 24.25 External frontend control (option R&S RTP-K553)

Suffix <m> indicates the index of the external frontend if multiple frontends are used.


- Setup and connection ..... 2444
- Frontend settings and parameters ..... 2447
- Accessories ..... 2460


### 24.25.1 Setup and connection


EFRontend<m>:CONNection:CONFig ..... 2445
EFRontend<m>:CONNection:CSTate? ..... 2445
EFRontend<m>:CONNection:ERRor? ..... 2445
EFRontend<m>:CONNection:STATe ..... 2446
EFRontend:COUNt ..... 2446
EFRontend:CSELect<m> ..... 2446
EFRontend<m>:IDN? ..... 2446
EFRontend:PRESet ..... 2447
EFRontend:SETup ..... 2447
EFRontend<m>:TYPE:LIST? ..... 2447

**EFRontend<m>:CONNection:CONFig** <Type>, <IPAddress or name>, [<Identifier>], [<Symbolic>]
**EFRontend<m>:CONNection:CONFig?** <Type>, <IPAddress or name>, [<Identifier>]

Defines the connection of the indicated frontend: type of the frontend and the IP address or host name. Optionally, the identifier string can be used.

**Suffix:**
<m> 1...4, index of the external frontend

**Parameters:**
<Symbolic> ???

**Parameters for setting and query:**
<Type> String with the type of the connected frontend. For a list of available type strings, use `EFRontend<m>:TYPE:LIST?`.

<IPAddress or name> String with the IP address or host name of the connected frontend.

<Identifier> IDN string of the connected frontend

**Usage:** Asynchronous command

**Manual operation:** See "Type" on page 1110

**EFRontend<m>:CONNection:CSTate?**

Queries the current connection state.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<State> ON | OFF
*RST: OFF

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Enable" on page 1109

**EFRontend<m>:CONNection:ERRor?**

Queries the errors that occur in the external frontend and in the communication between frontend and oscilloscope.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<State> Error string

**Usage:** Query only
Asynchronous command

**EFRontend<m>:CONNection:STATe** <State>

Switches the connection of the indicated frontend on or off.

**Suffix:**
<m> 1...4, index of the external frontend

**Parameters:**
<State> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Enable" on page 1109

**EFRontend:COUNt** <Number>

Sets the number of connected frontends.

**Parameters:**
<Number> Possible values: 1, 2, 3 or 4

**Manual operation:** See "Number of Frontends" on page 1108

**EFRontend:CSELect<m>** <Channel>

Selects the input channel to which the indicated external frontend is connected.

**Suffix:**
<m> 1..4
Position of the frontend, counted from left to right.

**Parameters:**
<Channel> CHAN1 | CHANnel1 | CHAN2 | CHANnel2 | CHAN3 | CHANnel3 | CHAN4 | CHANnel4
Input channel: CHAN1 = CHANnel1, CHAN2 = CHANnel2, CHAN3 = CHANnel3, CHAN4 = CHANnel4

**Manual operation:** See "Measurement Setup" on page 1108

**EFRontend<m>:IDN?**

Queries the identifier of the indicated frontend.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<IDNString> Identifier string

**Usage:** Query only

**Manual operation:** See "Identity" on page 1110

**EFRontend:PRESet**

Resets the complete configuration of the external frontends to the default values. External frontends have their own preset, the general preset does not affect the frontend settings.

**Usage:** Event
Asynchronous command

**Manual operation:** See "Preset settings" on page 1108

**EFRontend:SETup** <SetupType>

Selects the type of the test setup.

**Parameters:**
<SetupType> COULo | COURef | ADVanced
COULo: Coupled LO (local oscillators)
COURef: Coupled reference input
ADVanced: user-defined setup
*RST: COULo

**Manual operation:** See "Measurement Setup" on page 1108

**EFRontend<m>:TYPE:LIST?**

Queries the supported external frontend types.

To set the frontend type, use `EFRontend<m>:CONNection:CONFig`.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<TypeList> String with a list of supported frontend types.

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Type" on page 1110

### 24.25.2 Frontend settings and parameters

The frontend settings and the most important parameters are available in the GUI. All other parameters can be queried remotely if information is needed.

EFRontend<m>:AMPLmode....2448
EFRontend<m>:ATTenuation:AUTO....2449
EFRontend<m>:ATTenuation[:VALue]....2449
EFRontend<m>:ATTenuation:STEP:LIST?....2449
EFRontend<m>:FREQuency:BAND:COUNt?....2450
EFRontend<m>:FREQuency:BAND<n>:LIST?....2450

EFRontend<m>:FREQuency:BAND<n>:LOWer? 2450
EFRontend<m>:FREQuency:BAND<n>:UPPer? 2451
EFRontend<m>:FREQuency:BCONfig:AUTO 2451
EFRontend<m>:FREQuency:BCONfig:COUNt? 2451
EFRontend<m>:FREQuency:BCONfig:IFRange? 2451
EFRontend<m>:FREQuency:BCONfig:LIST? 2452
EFRontend<m>:FREQuency:BCONfig:RFRange? 2452
EFRontend<m>:FREQuency:BCONfig:SELect 2452
EFRontend<m>:FREQuency:IFBWidth:LIMit? 2453
EFRontend<m>:FREQuency:IFRequency:VALue? 2453
EFRontend<m>:FREQuency:LOSCillator:FREQuency? 2453
EFRontend<m>:FREQuency:LOSCillator:INPut:FREQuency? 2454
EFRontend<m>:FREQuency:LOSCillator:MODE 2454
EFRontend<m>:FREQuency:LOSCillator:OUTPut:FREQuency? 2454
EFRontend<m>:FREQuency:LOSCillator:OUTPut:STATe 2455
EFRontend<m>:FREQuency:IFRequency:MAX? 2455
EFRontend<m>:FREQuency:IFRequency:MIN? 2456
EFRontend<m>:FREQuency:IFRequency:SIDeband? 2456
EFRontend<m>:FREQuency:REFerence[:VALue] 2456
EFRontend<m>:FREQuency:REFerence:LIST? 2457
EFRontend<m>:FREQuency:RFRequency:MAX? 2457
EFRontend<m>:FREQuency:RFRequency:MIN? 2457
EFRontend<m>:FREQuency:RFRequency:VALue 2458
EFRontend<m>:GAIN? 2458
EFRontend<m>:IFRequency:ATTenuation? 2458
EFRontend<m>:RFConnector 2459
EFRontend<m>:RFINput:LEVel 2459
EFRontend<m>:UPDate 2459

**EFRontend<m>:AMPLmode** <Mode>

Defines the vertical scale of the signal.

**Suffix:**

<m> 1...4, index of the external frontend

**Parameters:**

<Mode> RF | IF

Scaling mode

**RF**

The vertical scale is calculated out of the specified RF Level and the external frontend gain so that the signal uses the complete height of the diagram. See also: `EFRontend<m>:GAIN?` and `EFRontend<m>:RFINput:LEVel`.

**IF**

Vertical scale = 50 mV/div

*RST: RF

**Manual operation:** See "Amplitude mode" on page 1113

**EFRontend<m>:ATTenuation:AUTO** <State>

Defines the RF attenuation, which is meant to protect the mixer of the external frontend from high RF input levels.

**Suffix:**
<m> 1...4, index of the external frontend

**Parameters:**
<State> ON | OFF

**ON**
The instrument provides a good compromise between intermodulation and noise.

**OFF**
Set `EFRontend<m>:ATTenuation[:VALue]` to fine-tune the effect of intermodulation and noise.

*RST: ON

**Usage:** Asynchronous command

**Manual operation:** See "RF attenuation auto mode, RF attenuation" on page 1113

**EFRontend<m>:ATTenuation[:VALue]** <RFAttenuation>

Sets the RF attenuation if `EFRontend<m>:ATTenuation:AUTO` is OFF.

**Suffix:**
<m> 1...4, index of the external frontend

**Parameters:**
<RFAttenuation> Get a list of possible attenuation step values with `EFRontend<m>:ATTenuation:STEP:LIST?`.

Range: -100 to 100
*RST: -10
Default unit: dB

**Usage:** Asynchronous command

**Manual operation:** See "RF attenuation auto mode, RF attenuation" on page 1113

**EFRontend<m>:ATTenuation:STEP:LIST?**

Queries the possible attenuation step values.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<BandList> String with a list of allowed attenuation step values.

**Example:**
```
EFR:ATT:STEP:LIST?
<--
"0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,
18,19,20,21,22,23,24,25,26,27,28,29,30,31"
```

**Usage:** Query only
Asynchronous command

**Manual operation:** See "RF attenuation auto mode, RF attenuation" on page 1113

---

**EFRontend<m>:FREQuency:BAND:COUNt?**

Queries the number of frequency bands provided by the selected frontend.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<NoBands> Integer, number of frequency bands

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:BAND<n>:LIST?**

Queries the frequency range that is supported by the selected frontend frequency band.

**Suffix:**
<m> 1...4, index of the external frontend

<n> 1..*
Band for multi-band frontends
Use `EFRontend<m>:FREQuency:BAND:COUNt?` to determine the number of available bands.

**Return values:**
<BandList> String with start and stop frequency of the selected band.

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:BAND<n>:LOWer?**

Queries the lower frequency limit of the band addressed by suffix <n>.

**Suffix:**
<m> 1...4, index of the external frontend

<n> 1..*
Band for multi-band frontends
Use `EFRontend<m>:FREQuency:BAND:COUNt?` to determine the number of available bands.

**Return values:**
<Frequency> Start frequency of the specified band

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:BAND<n>:UPPer?**

Queries the upper frequency limit of the band addressed by suffix <n>.

**Suffix:**
<m> 1...4, index of the external frontend

<n> 1..*
Band for multi-band frontends
Use `EFRontend<m>:FREQuency:BAND:COUNt?` to determine the number of available bands.

**Return values:**
<Frequency> End frequency of the specified band

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:BCONfig:AUTO** <Config>

Sets the RF band selection auto mode. Currently only auto mode and the lower IF band are supported.

**Suffix:**
<m> 1...4, index of the external frontend

**Parameters:**
<Config> ON

**Usage:** Asynchronous command

---

**EFRontend<m>:FREQuency:BCONfig:COUNt?**

Queries the number of available band configurations.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<State> Range: 1

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:BCONfig:IFRange?** <BandConfigID>

Queries the IF range of the indicated band.

**Suffix:**
<m> 1...4, index of the external frontend

**Query parameters:**
<BandConfigID> Frequency band configuration = 1.

**Return values:**
<FrequencyRange> String with the frequency range.

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:BCONfig:LIST?**

Queries the names of available band configurations. Currently only the lower IF band is supported.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<BandConfigList> String with the band name

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:BCONfig:RFRange?** <BandConfigID>

Queries the RF range of the indicated band.

**Suffix:**
<m> 1...4, index of the external frontend

**Query parameters:**
<BandConfigID> Frequency band configuration = 1.

**Return values:**
<FrequencyRange> String with the frequency range.

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:BCONfig:SELect** <Config>

Selects the band configuration.

**Suffix:**
<m> 1...4, index of the external frontend

**Parameters:**
<Config> String with the name of the band configuration, e.g. "IF Low".

**Usage:** Asynchronous command

**EFRontend<m>:FREQuency:IFBWidth:LIMit?**

Queries the maximum analysis bandwidth of the connected external frontend.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<IFValue> Range: 0 to 1E+12
Increment: 1E+6
*RST: 1E+9
Default unit: Hz

**Usage:** Query only
Asynchronous command

**Manual operation:** See "RF Bandwidth" on page 1112

**EFRontend<m>:FREQuency:IFRequency:VALue?**

Queries the intermediate frequency to which the signal is downconverted by the frontend.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<IFValue> Range: -1E+12 to 1E+12
Increment: 1E+6
*RST: 1E+9
Default unit: Hz

**Usage:** Query only
Asynchronous command

**Manual operation:** See "IF Output" on page 1112

**EFRontend<m>:FREQuency:LOSCillator:FREQuency?**

Queries the frequency of the local oscillator of the connected frontend.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<Value> Range: 0 to 1E+12
Increment: 1E+6
*RST: 10E+6
Default unit: Hz

**Usage:** Query only
Asynchronous command

**EFRontend<m>:FREQuency:LOSCillator:INPut:FREQuency?**

Queries the required frequency on LO input if the LO input mode is external. See also: `EFRontend<m>:FREQuency:LOSCillator:MODE` on page 2454

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<Value> Returns the same value as `EFRontend<m>:FREQuency:LOSCillator:OUTPut:FREQuency?`.

Range: 0 to 1E+12
Increment: 1E+6
*RST: 10E+6
Default unit: Hz

**Usage:** Query only
Asynchronous command

**Manual operation:** See "LO In Frequency (cable)" on page 1112

**EFRontend<m>:FREQuency:LOSCillator:MODE** <Value>

For measurement setups "Coupled LO" and "Coupled Ref", the command queries the input mode of the local oscillator. The mode is defined by the selected setup definition.

In "Advanced" measurement setup, set the LO input mode according to the connection of the local oscillator in your setup.

See also:`EFRontend:SETup` on page 2447.

**Suffix:**
<m> 1...4, index of the external frontend

**Parameters:**
<Value> INTernal | EXTernal

**INTernal**
The LO input of the selected frontend is NOT connected to the LO output of another frontend. The reference frequency is received via "Ref In" from the oscilloscope.

**EXTernal**
The LO input of the selected frontend is connected to the LO output of another frontend.

*RST: INTernal

**Usage:** Asynchronous command

**Manual operation:** See "LO input, LO output" on page 1113

**EFRontend<m>:FREQuency:LOSCillator:OUTPut:FREQuency?**

Queries the current LO output frequency.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<Value> Returns the same value as `EFRontend<m>:FREQuency:LOSCillator:INPut:FREQuency?`.

Range: 0 to 1E+12
Increment: 1E+6
*RST: 10E+6
Default unit: Hz

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:LOSCillator:OUTPut:STATe** <State>

For measurement setups "Coupled LO" and "Coupled Ref", the command queries the connection state of the LO output. The state is defined by the selected setup definition.

In "Advanced" measurement setup, set the LO output state according to the connection of the local oscillator in your setup.

See also:`EFRontend:SETup` on page 2447.

**Suffix:**
<m> 1...4, index of the external frontend

**Parameters:**
<State> ON | OFF

**ON**
The LO output of the selected frontend is connected to the LO input of another frontend.

**OFF**
The LO output is not connected.

*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "LO input, LO output" on page 1113

---

**EFRontend<m>:FREQuency:IFRequency:MAX?**

Queries the maximum IF frequency.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<IFMax> Range: -1E+12 to 1E+12
Increment: 1E+6
*RST: 1E+9
Default unit: Hz

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:IFRequency:MIN?**

Queries the minimum IF frequency.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<IFMin> Range: -1E+12 to 1E+12
Increment: 1E+6
*RST: 1E+9
Default unit: Hz

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:IFRequency:SIDeband?**

Queries the frequency band configuration, which is always IFLow.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<Sideband> IFLow

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:REFerence[:VALue]** <Frequency>

Sets the reference frequency that is used for frequency conversion on the external frontend.

You can query supported values with `EFRontend<m>:FREQuency:REFerence:LIST?`.

**Suffix:**
<m> 1...4, index of the external frontend

**Parameters:**
<Frequency> Range: 10E+6 (coupled LO and coupled reference setup), 10E+6 | 640E+6 | 1E+9 (advanced setup)
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "Reference frequency" on page 1112

**EFRontend<m>:FREQuency:REFerence:LIST?**

Queries the possible reference frequencies, which are supported by the connected external frontend.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<TypeList> String with frequency values

**Example:**
```
EFR:SET COULo
EFR:FREQ:REF:LIST?
<-- 10000000

EFR:SET COURef
EFR:FREQ:REF:LIST?
<-- 10000000

EFR:SET ADVanced
EFR:FREQ:REF:LIST?
<-- 10000000,640000000,10000000000
```

**Usage:** Query only
Asynchronous command

**EFRontend<m>:FREQuency:RFRequency:MAX?**

Queries the maximum RF frequency.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**
<RFMax> Range: 0 to 1E+12
Increment: 1E+6
*RST: 1E+9
Default unit: Hz

**Usage:** Query only
Asynchronous command

**EFRontend<m>:FREQuency:RFRequency:MIN?**

Queries the minimum RF frequency.

**Suffix:**
<m> 1...4, index of the external frontend

**Return values:**

<RFMin> Range: 0 to 1E+12
Increment: 1E+6
*RST: 1E+9
Default unit: Hz

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:FREQuency:RFRequency:VALue** <RFValue>

Sets the RF frequency of the signal that the external frontend receives from the DUT. The value is the center frequency that determines the LO frequency generation.

**Suffix:**

<m> 1...4, index of the external frontend

**Parameters:**

<RFValue> Range: 24E+9 to 44E+9
Increment: 1E+6
*RST: 24E+9
Default unit: Hz

**Usage:** Asynchronous command

**Manual operation:** See "RF Input" on page 1112

---

**EFRontend<m>:GAIN?**

Returns the gain if the connected external frontend.

**Suffix:**

<m> 1...4, index of the external frontend

**Return values:**

<Gain> Range: -100 to 100
Increment: 1
*RST: 0
Default unit: dB

**Usage:** Query only

**Manual operation:** See "External Frontend gain" on page 1113

---

**EFRontend<m>:IFRequency:ATTenuation?**

Queries the IF attenuation.

**Suffix:**

<m> 1...4, index of the external frontend

**Return values:**

<IFValue> Range: -100 to 100
Increment: 1
*RST: 0
Default unit: dB

**Usage:** Query only
Asynchronous command

---

**EFRontend<m>:RFConnector** <RFConnector>

Sets the RF connector.

**Suffix:**

<m> 1...4, index of the external frontend

**Parameters:**

<RFConnector> RF1

---

**EFRontend<m>:RFINput:LEVel** <RFLevel>

Sets the reference level of the signal at the RF input of the external frontend. If this level is entered correctly, the external frontend is configured such that the level at the input of the R&S RTP is optimal.

**Suffix:**

<m> 1...4, index of the external frontend

**Parameters:**

<RFLevel> Range: -100 to 100
Increment: 1
*RST: -10
Default unit: dBm

**Usage:** Asynchronous command

**Manual operation:** See "RF Level" on page 1112

---

**EFRontend<m>:UPDate**

Updates the firmware on the connected external frontend. This process can take some time.

**Suffix:**

<m> 1...4, index of the external frontend

**Usage:** Event
Asynchronous command

### 24.25.3 Accessories

EFRontend<m>:ALIGn:FILE....2460
EFRontend<m>:ALIGn:STATe....2460

**EFRontend<m>:ALIGn:FILE** <FilePath>

Sets the path to the S-parameter file that contains the correction values for the cable that connects the frontend to the oscilloscope.

**Suffix:**
<m> 1...4, index of the external frontend

**Parameters:**
<FilePath> String with path and name of the s2p file.

**Usage:** Asynchronous command

**Manual operation:** See "S-Parameters" on page 1114

**EFRontend<m>:ALIGn:STATe** <CableAlignment>

Activates the correction of the IF signal due to IF cable loss from the frontend to the oscilloscope. Activating loads the S-parameter file that is defined with `EFRontend<m>:ALIGn:FILE` on page 2460.

**Suffix:**
<m> 1...4, index of the external frontend

**Parameters:**
<CableAlignment> ON | OFF
*RST: OFF

**Usage:** Asynchronous command

**Manual operation:** See "Cable Deembedding: Enable" on page 1114

## 24.26 Maintenance

Some of the commands in the following chapter are asynchronous. An overlapping or asynchronous command does not automatically finish executing before the next command starts executing. If overlapping commands must be executed in a defined order, e.g. to avoid wrong measurement results, they must be serviced sequentially.

To prevent an overlapping execution of commands, one of the commands `*OPC`, `*OPC?` or `*WAI` can be used after the command or a command set.

For more information, see:

- www.rohde-schwarz.com/rc-via-scpi, chapter "Command Sequence and Synchronization"
- Chapter 24.3, "Programming examples", on page 1172

DIAGnostic:SERVice:WFAModel?....2461
DIAGnostic:SERVice:WFASeries?....2461
DIAGnostic:SERVice:WFAType?....2461
CALibration:DATE?....2461
CALibration:TIME?....2462
CALibration:RESult?....2462
DIAGnostic:SERVice:STST:EXECute....2462
DIAGnostic:SERVice:STST:STATe?....2462
DIAGnostic:SERVice:PWD....2463

**DIAGnostic:SERVice:WFAModel?**

Returns the model name of the oscilloscope.

**Return values:**
<WFAModel> Product type

**Usage:** Query only
Asynchronous command

**DIAGnostic:SERVice:WFASeries?**

Returns the model series of the oscilloscope.

**Return values:**
<WFASeries> RTO | RTE | RTP
*RST: RTO

**Usage:** Query only
Asynchronous command

**DIAGnostic:SERVice:WFAType?**

Returns the instrument family of the oscilloscope.

**Return values:**
<WFAType> RTO2000 | RTO6 | RTP | RTPFL
RTP: part numbers 1320.5007
RTPFl: part numbers 1803.7000
*RST: RTO2000

**Usage:** Query only
Asynchronous command

**CALibration:DATE?**

Returns the date of the last selfalignment.

**Return values:**
<Date>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Date / Time / Overall alignment state" on page 104

---

**CALibration:TIME?**

Returns the time of the last selfalignment.

**Return values:**
<Time>

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Date / Time / Overall alignment state" on page 104

---

**CALibration:RESult?**

Returns the result of the last selfalignment and the current alignment status. In remote mode, `*CAL?` provides more detailed information.

**Return values:**
<ResultState> PASSed | FAILed | NOALigndata
*RST: FAILed

**Usage:** Query only
Asynchronous command

**Manual operation:** See "Date / Time / Overall alignment state" on page 104

---

**DIAGnostic:SERVice:STST:EXECute**

Starts the selftest.

**Usage:** Event
Asynchronous command

---

**DIAGnostic:SERVice:STST:STATe?**

Returns the summary result of the selftest.

**Return values:**
<State> PSSD | FAILed | UNDefined
*RST: UNDefined

**Usage:** Query only
Asynchronous command

**Manual operation:** See "State" on page 106

**DIAGnostic:SERVice:PWD** <Password>

Sets the password to enter the service mode.

**Setting parameters:**
<Password> Password string

**Usage:** Setting only
Asynchronous command

## 24.27 Status reporting

This chapter describes the remote commands that are used to read the status registers.

For information on structure, hierarchy, and contents of the status registers, see Chapter 23.5.4, "Remote control - status reporting system", on page 1154.

- STATus:OPERation register....2463
- STATus:QUEStionable registers....2464

### 24.27.1 STATus:OPERation register

`STATus:OPERation` commands provide information on the activity of the instrument.

See also: "STATus:OPERation register" on page 1155

STATus:OPERation:CONDition?....2463
STATus:OPERation[:EVENt]?....2463
STATus:OPERation:ENABle....2464

**STATus:OPERation:CONDition?**
**STATus:OPERation[:EVENt]?**

The CONDition command returns information on actions the instrument is currently executing. The contents of the register is retained.

The EVENt command returns information on actions the instrument has executed since the last reading. Reading the EVENt register deletes its contents.

Bits:

- 0 = ALIGnment
- 2 = AUToset
- 3= WTRIgger (wait for trigger)
- 4= MEASuring

**Usage:** Query only

**STATus:OPERation:ENABle** <Enable>

Controls the ENABle part of the STATus:OPERation register. The ENABle defines which events in the EVENt part of the status register are forwarded to the OPERation summary bit (bit 7) of the status byte. The status byte can be used to create a service request.

**Parameters:**

<Enable> Range: 1 to 65535
Increment: 1

**Example:** `STATus:OPERation:ENABle 5`
The ALIGnment event (bit 0) and AUToset event (bit 2) are forwarded to the OPERation summary bit of the status byte.

### 24.27.2 STATus:QUEStionable registers

The commands of the `STATus:QUEStionable` subsystem control the status reporting structures of the STATus:QUEStionable registers.

See also: "STATus:QUEStionable register" on page 1156

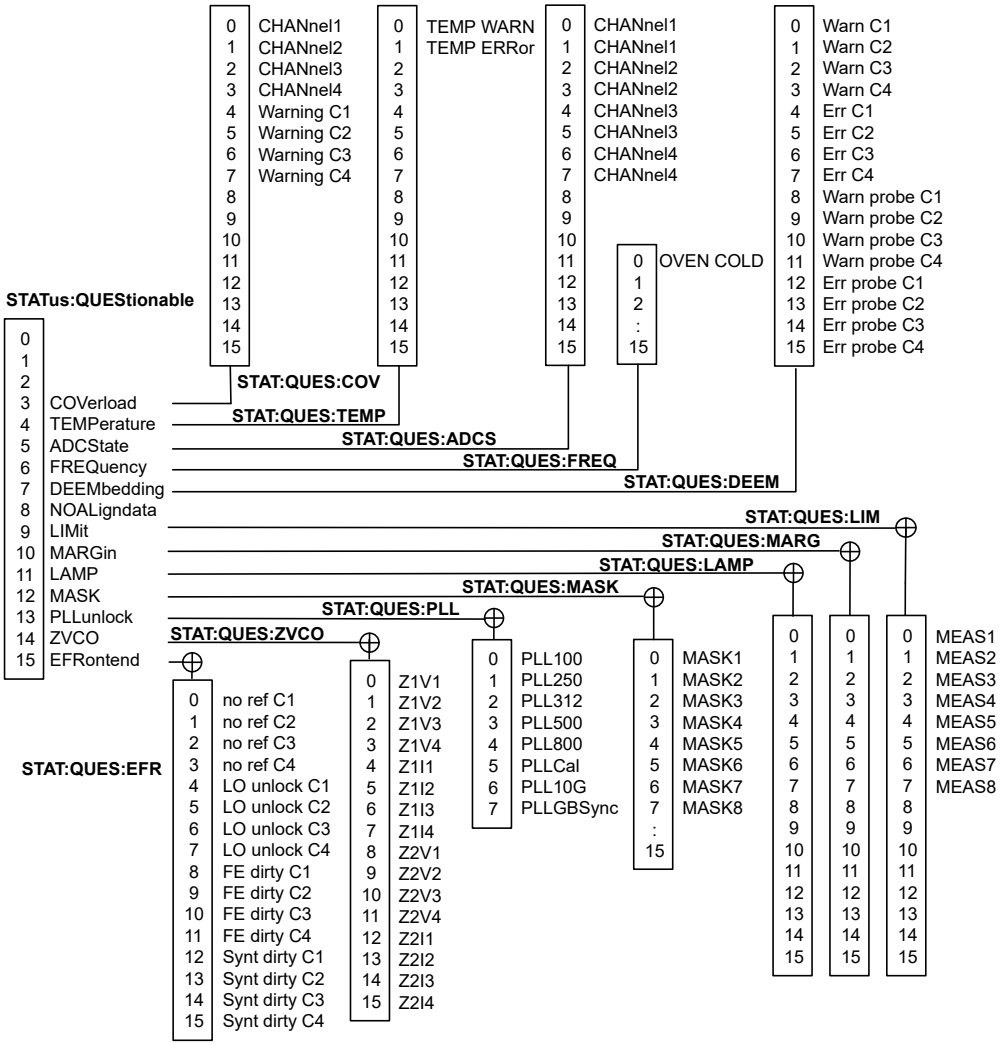



*Figure 24-1: Overview of the STATus:QUEStionable register*

The following commands are available:

STATus:QUEStionable:COVerload:CONDition? 2466
STATus:QUEStionable:TEMPerature:CONDition? 2466
STATus:QUEStionable:ADCState:CONDition? 2467
STATus:QUEStionable:LIMit:CONDition? 2467
STATus:QUEStionable:MARGin:CONDition? 2467
STATus:QUEStionable:FREQuency:CONDition? 2467
STATus:QUEStionable:PLL:CONDition? 2467
STATus:QUEStionable:DEEMbedding:CONDition? 2467
STATus:QUEStionable:LAMPlitude:CONDition? 2467
STATus:QUEStionable:ZVCoverload:CONDition? 2467
STATus:QUEStionable:EFRontend:CONDition? 2467
STATus:QUEStionable:MASK:CONDition? 2467
STATus:QUEStionable:COVerload:ENABle 2467
STATus:QUEStionable:TEMPerature:ENABle 2467
STATus:QUEStionable:ADCState:ENABle 2467

STATus:QUEStionable:LIMit:ENABle....2467
STATus:QUEStionable:MARGin:ENABle....2467
STATus:QUEStionable:FREQuency:ENABle....2467
STATus:QUEStionable:PLL:ENABle....2467
STATus:QUEStionable:DEEMbedding:ENABle....2467
STATus:QUEStionable:LAMPlitude:ENABle....2467
STATus:QUEStionable:ZVCoverload:ENABle....2467
STATus:QUEStionable:EFRontend:ENABle....2467
STATus:QUEStionable:MASK:ENABle....2467
STATus:QUEStionable:COVerload[:EVENt]?....2467
STATus:QUEStionable:TEMPerature[:EVENt]?....2467
STATus:QUEStionable:ADCState[:EVENt]?....2467
STATus:QUEStionable:LIMit[:EVENt]?....2467
STATus:QUEStionable:MARGin[:EVENt]?....2467
STATus:QUEStionable:FREQuency:EVENt?....2467
STATus:QUEStionable:PLL[:EVENt]?....2467
STATus:QUEStionable:DEEMbedding[:EVENt]?....2467
STATus:QUEStionable:LAMPlitude[:EVENt]?....2467
STATus:QUEStionable:ZVCoverload[:EVENt]?....2468
STATus:QUEStionable:EFRontend[:EVENt]?....2468
STATus:QUEStionable:MASK[:EVENt]?....2468
STATus:QUEStionable:COVerload:NTRansition....2468
STATus:QUEStionable:TEMPerature:NTRansition....2468
STATus:QUEStionable:ADCState:NTRansition....2468
STATus:QUEStionable:LIMit:NTRansition....2468
STATus:QUEStionable:MARGin:NTRansition....2468
STATus:QUEStionable:FREQuency:NTRansition....2468
STATus:QUEStionable:PLL:NTRansition....2468
STATus:QUEStionable:DEEMbedding:NTRansition....2468
STATus:QUEStionable:LAMPlitude:NTRansition....2468
STATus:QUEStionable:ZVCoverload:NTRansition....2468
STATus:QUEStionable:EFRontend:NTRansition....2468
STATus:QUEStionable:MASK:NTRansition....2468
STATus:QUEStionable:COVerload:PTRansition....2468
STATus:QUEStionable:TEMPerature:PTRansition....2468
STATus:QUEStionable:ADCState:PTRansition....2468
STATus:QUEStionable:LIMit:PTRansition....2468
STATus:QUEStionable:MARGin:PTRansition....2468
STATus:QUEStionable:FREQuency:PTRansition....2468
STATus:QUEStionable:PLL:PTRansition....2468
STATus:QUEStionable:DEEMbedding:PTRansition....2468
STATus:QUEStionable:LAMPlitude:PTRansition....2468
STATus:QUEStionable:ZVCoverload:PTRansition....2468
STATus:QUEStionable:EFRontend:PTRansition....2468
STATus:QUEStionable:MASK:PTRansition....2468
MTESt:SBITnumber?....2469

**STATus:QUEStionable:COVerload:CONDition?**
**STATus:QUEStionable:TEMPerature:CONDition?**

**STATus:QUEStionable:ADCState:CONDition?**
**STATus:QUEStionable:LIMit:CONDition?**
**STATus:QUEStionable:MARGin:CONDition?**
**STATus:QUEStionable:FREQuency:CONDition?**
**STATus:QUEStionable:PLL:CONDition?**
**STATus:QUEStionable:DEEMbedding:CONDition?**
**STATus:QUEStionable:LAMPlitude:CONDition?**
**STATus:QUEStionable:ZVCoverload:CONDition?**
**STATus:QUEStionable:EFRontend:CONDition?**
**STATus:QUEStionable:MASK:CONDition?**

Returns the contents of the CONDition part of the status register to check for questionable instrument or measurement states. Reading the CONDition registers does not delete the contents.

**Usage:** Query only

---

**STATus:QUEStionable:COVerload:ENABle** <Value>
**STATus:QUEStionable:TEMPerature:ENABle** <Value>
**STATus:QUEStionable:ADCState:ENABle** <Value>
**STATus:QUEStionable:LIMit:ENABle** <Value>
**STATus:QUEStionable:MARGin:ENABle** <Value>
**STATus:QUEStionable:FREQuency:ENABle** <Value>
**STATus:QUEStionable:PLL:ENABle** <Value>
**STATus:QUEStionable:DEEMbedding:ENABle** <Value>
**STATus:QUEStionable:LAMPlitude:ENABle** <Value>
**STATus:QUEStionable:ZVCoverload:ENABle** <Value>
**STATus:QUEStionable:EFRontend:ENABle** <Value>
**STATus:QUEStionable:MASK:ENABle** <Value>

Sets the ENABle part that allows true conditions in the EVENt part to be reported in the summary bit. If a bit is set to 1 in the enable part and its associated event bit transitions to true, a positive transition occurs in the summary bit and is reported to the next higher level.

**Parameters:**
<Value> Bit mask in decimal representation

**Example:** `STATus:QUEStionable:MASK:ENABle 24`
Set bits no. 3 and 4 of the STATus:QUEStionable:MASK:ENABle register part: $24 = 8 + 16 = 2^3 + 2^4$

---

**STATus:QUEStionable:COVerload[:EVENt]?**
**STATus:QUEStionable:TEMPerature[:EVENt]?**
**STATus:QUEStionable:ADCState[:EVENt]?**
**STATus:QUEStionable:LIMit[:EVENt]?**
**STATus:QUEStionable:MARGin[:EVENt]?**
**STATus:QUEStionable:FREQuency:EVENt?**
**STATus:QUEStionable:PLL[:EVENt]?**
**STATus:QUEStionable:DEEMbedding[:EVENt]?**
**STATus:QUEStionable:LAMPlitude[:EVENt]?**

**STATus:QUEStionable:ZVCoverload[:EVENt]?**
**STATus:QUEStionable:EFRontend[:EVENt]?**
**STATus:QUEStionable:MASK[:EVENt]?**

Returns the contents of the EVENt part of the status register to check whether an event has occurred since the last reading. Reading an EVENt register deletes its contents.

**Usage:** Query only

---

**STATus:QUEStionable:COVerload:NTRansition** <Value>
**STATus:QUEStionable:TEMPerature:NTRansition** <Value>
**STATus:QUEStionable:ADCState:NTRansition** <Value>
**STATus:QUEStionable:LIMit:NTRansition** <Value>
**STATus:QUEStionable:MARGin:NTRansition** <Value>
**STATus:QUEStionable:FREQuency:NTRansition** <Value>
**STATus:QUEStionable:PLL:NTRansition** <Value>
**STATus:QUEStionable:DEEMbedding:NTRansition** <Value>
**STATus:QUEStionable:LAMPlitude:NTRansition** <Value>
**STATus:QUEStionable:ZVCoverload:NTRansition** <Value>
**STATus:QUEStionable:EFRontend:NTRansition** <Value>
**STATus:QUEStionable:MASK:NTRansition** <Value>

Sets the negative transition filter. If a bit is set, a 1 to 0 transition in the corresponding bit of the condition register causes a 1 to be written in the corresponding bit of the event register.

**Parameters:**
<Value> Bit mask in decimal representation

**Example:** `STATus:QUEStionable:MASK:NTRansition 24`
Set bits no. 3 and 4 of the STATus:QUEStionable:MASK:NTRansition register part: $24 = 8 + 16 = 2^3 + 2^4$

---

**STATus:QUEStionable:COVerload:PTRansition** <Value>
**STATus:QUEStionable:TEMPerature:PTRansition** <Value>
**STATus:QUEStionable:ADCState:PTRansition** <Value>
**STATus:QUEStionable:LIMit:PTRansition** <Value>
**STATus:QUEStionable:MARGin:PTRansition** <Value>
**STATus:QUEStionable:FREQuency:PTRansition** <Value>
**STATus:QUEStionable:PLL:PTRansition** <Value>
**STATus:QUEStionable:DEEMbedding:PTRansition** <Value>
**STATus:QUEStionable:LAMPlitude:PTRansition** <Value>
**STATus:QUEStionable:ZVCoverload:PTRansition** <Value>
**STATus:QUEStionable:EFRontend:PTRansition** <Value>
**STATus:QUEStionable:MASK:PTRansition** <Value>

Sets the positive transition filter. If a bit is set, a 0 to 1 transition in the corresponding bit of the condition register causes a 1 to be written in the corresponding bit of the event register.

**Parameters:**

<Value> Bit mask in decimal representation

**Example:** `STATus:QUEStionable:MASK:PTRansition 24`
Set bits no. 3 and 4 of the STATus:QUEStionable:MASK:PTRansition register part: *24 = 8 + 16 = $2^3 + 2^4$*

---

**MTESt:SBITnumber?** <MaskTestName>

Returns the number of the status bit that belongs to the indicated mask test in the `STATus:QUEStionable:MASK` register. The status bits are assigned in the order of mask creation.

**Query parameters:**

<MaskTestName> String with the name of the mask test

**Return values:**

<StatusBitNumber> Bit number, see "STATus:QUEStionable:MASK register" on page 1161.

Range: 0 to 7
Increment: 1
*RST: 0

**Example:**

```
:MTESt:SBITnumber? 'MT1'
<--0
:MTESt:SBITnumber? 'MT2'
<--1
:MTESt:SBITnumber? 'MT3'
<--2
```

**Usage:** Query only
Asynchronous command

## 24.28 Remote trace

The commands in this section configure tracing of the remote control interface and of events. They also configure the display of the SCPI remote trace.

Before you start tracing, configure all settings as desired. Modifying settings while tracing is active may result in loss of already traced data. Useful exception: Selecting a new target file while tracing is allowed. For start mode `EXPLicit` a restart of the instrument resets the settings to the documented default values.

If you want to start tracing already during startup of the instrument, configure all settings (including start mode `AUTO`). Then restart your instrument. Tracing will be started automatically during the restart, using the already configured settings.

If you use an XML file as trace file, ensure that tracing is stopped properly. If tracing is aborted instead of stopped, for example by shutting down the instrument for stop mode `EXPLicit`, the XML file will be invalid, because some tags are not closed.

When the maximum file size is reached (except for stop mode `BUFFerfull`) or if tracing is started with an already existing trace file, a backup of the trace file is created and the file itself is reset and overwritten. When the file is full for the second time or when tracing is started the next time, the first backup file is lost because it is overwritten by the next backup. In order to prevent loss of data, set a sufficient file size, select an appropriate stop mode and archive/copy completed trace files if you want to keep them.

### 24.28.1 Standard commands

TRACe:REMote:MODE:FILE:NAME ..... 2470
TRACe:REMote:MODE:FILE:FORMat ..... 2470
TRACe:REMote:MODE:FILE:SIZE ..... 2470
TRACe:REMote:MODE:FILE:STARtmode ..... 2470
TRACe:REMote:MODE:FILE:STOPmode ..... 2471
TRACe:REMote:MODE:FILE:ENABle ..... 2471
TRACe:REMote:MODE:FILE:FILTer ..... 2471

**TRACe:REMote:MODE:FILE:NAME** <FilePath>

Sets the directory and file name where the remote trace file is stored.

**Parameters:**
<FilePath> String parameter

**TRACe:REMote:MODE:FILE:FORMat** <Format>

Sets the file format of the remote trace file.

**Parameters:**
<Format> ASCii | XML

**TRACe:REMote:MODE:FILE:SIZE** <FileSize>

Sets the maximum size of the remote trace file.

**Parameters:**
<FileSize> Integer value

**TRACe:REMote:MODE:FILE:STARtmode** <StartMode>

Defines how the remote trace is started.

**Parameters:**
<StartMode> AUTO | EXPLicit

**AUTO**
Starts the remote trace immediately.

**EXPLicit**
Starts remote trace with `TRACe:REMote:MODE:FILE:ENABle ON`

---

**TRACe:REMote:MODE:FILE:STOPmode** <StopMode>

Defines when the remote trace is stopped.

**Parameters:**

<StopMode> AUTO | EXPLicit | ERRor | BUFFerfull

**AUTO**
Ends the remote trace on device shutdown.

**EXPLicit**
Ends remote trace with `TRACe:REMote:MODE:FILE:ENABle OFF`

**ERRor**
Ends remote trace when a SCPI error occurs.

**BUFFerfull**
Ends remote trace when the maximum file size is reached.

---

**TRACe:REMote:MODE:FILE:ENABle** <Enable>

Enables and disables the remote trace to file.

**Parameters:**

<Enable> ON | OFF

---

**TRACe:REMote:MODE:FILE:FILTer** <Input>, <Output>, <Error>, <Trigger>, <DeviceClear>, <StatusRegister>, <Connection>, <RemoteLocalEvents>, <Locking>

Defines the content of the remote trace file.

**Parameters:**

<Input> ON | OFF
Input data

<Output> ON | OFF
Output data

<Error> ON | OFF
New SCPI error queue entries

<Trigger> ON | OFF
Trigger events

<DeviceClear> ON | OFF
Device clear events

<StatusRegister> ON | OFF
Status register conditions

<Connection> ON | OFF
Open/close connection events

<RemoteLocalEvents> ON | OFF
Local/remote transition events

<Locking> ON | OFF
Remote locking events

### 24.28.2 Diagnostic remote trace commands

TRACe:REMote:MODE:FILE:DEXecution:DURation ..... 2472
TRACe:REMote:MODE:FILE:RPC ..... 2472
TRACe:REMote:MODE:FILE:PARSer ..... 2472
TRACe:REMote:MODE:FILE:FUNCtions ..... 2472

**TRACe:REMote:MODE:FILE:DEXecution:DURation** <Enable>

Traces the device execution time of a command

**Parameters:**
<Enable> ON | OFF

**TRACe:REMote:MODE:FILE:RPC** <Enable>

Enables and disables output of rpc calls to remote trace.

**Parameters:**
<Enable> ON | OFF

**TRACe:REMote:MODE:FILE:PARSer** <Enable>

Enables and disables output of parser transitions to remote trace.

**Parameters:**
<Enable> ON | OFF

**TRACe:REMote:MODE:FILE:FUNCtions** <Enable>

Enables and disables output of function names to remote trace.

**Parameters:**
<Enable> ON | OFF

# 24.29 Deprecated commands

The following commands are provided for compatibility to previous oscilloscope versions only. For new remote control programs, use the specified alternative commands.

## 24.29.1 Base instrument

| Legacy command | Replaced by | Comment |
|---|---|---|
| BUSFormat | For parallel buses:<br>BUS<m>:PARallel:DATA:FORMat on page 2317<br>For serial buses:<br>BUS<m>:FORMat on page 1639 | Replaced since FW 4.70 |
| EXPort:RESult:SELect<br>EXPort:RESult:NUMeric<br>EXPort:RESult:NAME<br>EXPort:RESult:SAVE | | Functionality is not supported.<br>Removed since FW 5.20 |
| SEARch:TRIGger:DATatoclock[:STATe]<br>SEARch:TRIGger:DATatoclock:ACOPy<br>SEARch:TRIGger:DATatoclock:CEDGe<br>SEARch:TRIGger:DATatoclock:CLEVel<br>SEARch:TRIGger:DATatoclock:CSOurce<br>SEARch:TRIGger:DATatoclock:HTIMe<br>SEARch:TRIGger:DATatoclock:STIMe | SEARch:TRIGger:SETHold[:STATe] on page 1568<br>SEARch:TRIGger:SETHold:ACOPy on page 1569<br>SEARch:TRIGger:SETHold:CEDGe on page 1584<br>SEARch:TRIGger:SETHold:CLEVel on page 1585<br>SEARch:TRIGger:SETHold:CSOurce on page 1585<br>SEARch:TRIGger:SETHold:HTIMe on page 1585<br>SEARch:TRIGger:SETHold:STIMe on page 1586 | Replaced since FW 5.00 |
| TRIGger<m>:DATatoclock:CSOurce[:VALue]<br>TRIGger<m>:DATatoclock:CSOurce:EDGE<br>TRIGger<m>:DATatoclock:CSOurce:LEVel<br>TRIGger<m>:DATatoclock:HTIMe<br>TRIGger<m>:DATatoclock:STIMe | TRIGger<m>:SETHold:CSOurce[:VALue] on page 1322<br>TRIGger<m>:SETHold:CSOurce:EDGE on page 1321<br>TRIGger<m>:SETHold:CSOurce:LEVel on page 1322<br>TRIGger<m>:SETHold:HTIMe on page 1323<br>TRIGger<m>:SETHold:STIMe on page 1323 | Replaced since FW 5.00 |
| TRIGger<m>:OUT:STATe | TRIGger<m>:OUT:ACTion on page 1374 | Replaced since FW 5.00 |

| Legacy command | Replaced by | Comment |
|---|---|---|
| TRIGger<m>:PARallel:DATatoclock:CSOurce[:VALue]<br>TRIGger<m>:PARallel:DATatoclock:CSOurce:EDGE<br>TRIGger<m>:PARallel:DATatoclock:STIMe<br>TRIGger<m>:PARallel:DATatoclock:HTIMe | TRIGger<m>:PARallel:SETHold:CSOurce[:VALue] on page 2323<br>TRIGger<m>:PARallel:SETHold:CSOurce:EDGE on page 2327<br>TRIGger<m>:PARallel:SETHold:HTIMe on page 2327<br>TRIGger<m>:PARallel:SETHold:STIMe on page 2327 | Replaced since FW 5.00 |
| TRIGger<m>:SEQuence:MODE | TRIGger<m>:SEQuence:TYPE on page 1367 | Replaced since FW 5.10 |

### 24.29.2 Options

| Legacy command | Replaced by | Comment |
|---|---|---|
| **RFFE** | | |
| SEARch:TRIGger:RFFE:ERRor:BP<br>SEARch:TRIGger:RFFE:ERRor:NOResponse<br>SEARch:TRIGger:RFFE:ERRor:USEQuence | none | Functionality is no longer supported since FW 4.50. |
| **I2C** | | |
| TRIGger<m>:I2C:MODE | TRIGger<m>:I2C:TYPE on page 1647 | Replaced since FW 5.20 |
| **SPI** | | |
| TRIGger<m>:SPI:MODE<br>SEARch:TRIGger:SPI:MODE | TRIGger<m>:SPI:TYPE on page 1682<br>SEARch:TRIGger:SPI:TYPE on page 1691 | Replaced since FW 5.20 |
| **Ethernet** | | |
| SEARch:TRIGger:ETHernet:ERRor:SELect<br>SEARch:TRIGger:ETHernet:FRAMe:SELect | SEARch:TRIGger:ETHernet:TYPE on page 1842 | Replaced since FW 5.20 |
| **Advanced jitter analysis** | | |
| ADVJitter<m>::RESult:TYPBathtub<br>ADVJitter<m>::RESult:TYPSynteye | | Functionality is not supported. |
| ADVJitter<m>:SIGNal:UIOFfset | ADVJitter<m>:SIGNal:SAMTime on page 2383 | Replaced since FW 5.10 |
| **Eye diagram analysis** | | |
| EYE<m>:HORizontal:UINTerval:BITRate<br>EYE<m>:TREFerence[:SOURce]<br>EYE<m>:TREFerence:SLOPe<br>EYE<m>:TREFerence:LEVel | | Functionality is not supported since FW 5.20 |

# 25 Maintenance and support

The instrument does not need periodic maintenance. Only the cleaning of the instrument is essential.

To protect the front panel and to transport the instrument to another workplace safely and easily, various accessories are provided. Refer to the data sheet for available covers and cases and their order numbers.

The adjustment of the OCXO oscillator is described in the service manual.

## 25.1 Cleaning

How to clean the product is described in "Cleaning the product" on page 15.

Do not use any liquids for cleaning. Cleaning agents, solvents, acids and bases can damage the front panel labeling, plastic parts and display.

## 25.2 Power cables for R&S RTP

Rohde & Schwarz provides power cables with different properties for use in different countries.

***Table 25-1: List of power cables***

| Material No. | Material name | Recommended for |
|---|---|---|
| 3586.6406.00 | Power cable 2.5 m, CEE 7/7 - C19 | Europe |
| 3630.6882.00 | Power cable 2.5 m, NEMA 5/15 - C19 | USA |
| 3640.3834.00 | Power cable 2.5 m, SAA3 - C19 | China, Australia |
| 3629.0032.00 | Power cord 2.5 m, BS1363 - C19 | UK |
| 3639.9132.00 | Power cable 2.5 m - C19-YP12 | Japan |
| 3630.6899.00 | Power cable 2.5 m, ZA3 - C19 | South Africa, India |
| 3630.6901.00 | Power cable 2.5 m, 12G - C19 | Switzerland |
| 3630.6924.00 | Power cable 2.5 m, BR/3/20 - C19 | Brazil |
| 3658.0566.00 | Power cable 2.5 m, IL/3 – C19 | Israel |

## 25.3 Contacting customer support

**Technical support – where and when you need it**

For quick, expert help with any Rohde & Schwarz product, contact our customer support center. A team of highly qualified engineers provides support and works with you to find a solution to your query on any aspect of the operation, programming or applications of Rohde & Schwarz products.

**Contact information**

Contact our customer support center at www.rohde-schwarz.com/support, or follow this QR code:

*Figure 25-1: QR code to the Rohde & Schwarz support page*

## 25.4 Information for technical support

If you encounter problems that you cannot solve yourself, contact your Rohde & Schwarz support center, see Contacting customer support.

The support center finds solutions more quickly and efficiently, if you provide them with information on the instrument and an error description. To create, collect and save the required information, you can create a service report.

The service report is a ZIP file with a complete bug report, all relevant setup information, reporting and log files, alignment files, and the instrument configuration.

1. Open the "Settings" > "Maintenance" dialog.
2. Select the "Service" tab.
3. Tap "Service reporter: Start".

   The service reporter creates the report and saves it as ZIP file directly on the Windows desktop.
4. Create a support ticket that describes the problem, and attach the report file.

On the instrument, you can find log files, the device footprint and report files, and other information in the "AppData" folder on the Windows desktop (`C:\ProgramData\Rohde-Schwarz\RTx`). The "Maintenance" dialog box also pro-

vides information on your instrument configuration which can be helpful in case you need support.

## 25.5 Data security

If you have to send the instrument to the service, or if the instrument is used in a secured environment, consider the document "Instrument Security Procedures" that is delivered on the R&S RTP web page.

Instrument configuration data and user data are stored on a removable hard disk only. Thus it is sufficient to remove the hard disk before the instrument leaves a secured environment. Details are given in the document mentioned above.

## 25.6 Transporting

### Lifting and carrying

See: "Lifting and carrying the instrument" on page 12

### Packing

Use the original packaging material. It consists of antistatic wrap for electrostatic protection and packing material designed for the product.

If you do not have the original packaging, use similar materials that provide the same level of protection. You can also contact your local Rohde & Schwarz service center for advice.

### Securing

When moving the R&S RTP in a vehicle or using transporting equipment, make sure that the R&S RTP is properly secured. Only use items intended for securing objects.

### Transport altitude

Unless otherwise specified in the data sheet, the maximum transport altitude without pressure compensation is 4500 m above sea level.

## 25.7 Storage

Protect the product against dust. Ensure that the environmental conditions, e.g. temperature range and climatic load, meet the values specified in the data sheet.

## 25.8 Disposal

Rohde & Schwarz is committed to making careful, ecologically sound use of natural resources and minimizing the environmental footprint of our products. Help us by disposing of waste in a way that causes minimum environmental impact.

**Disposing of electrical and electronic equipment**

A product that is labeled as follows cannot be disposed of in normal household waste after it has come to the end of its life. Even disposal via the municipal collection points for waste electrical and electronic equipment is not permitted.

*Figure 25-2: Labeling in line with EU directive WEEE*

Rohde & Schwarz has developed a disposal concept for the eco-friendly disposal or recycling of waste material. As a manufacturer, Rohde & Schwarz completely fulfills its obligation to take back and dispose of electrical and electronic waste. Contact your local service representative to dispose of the product.

# List of commands

*CAL?....1197
*CLS....1197
*ESE....1197
*ESR?....1197
*IDN?....1197
*IST?....1198
*OPC....1198
*OPT?....1198
*PCB....1199
*PRE....1199
*PSC....1199
*RCL....1199
*RST....1199
*SAV....1200
*SRE....1200
*STB?....1200
*TRG....1200
*TST?....1200
*WAI....1201
ACQuire:ARESet:COUNt....1246
ACQuire:ARESet:IMMediate....1245
ACQuire:ARESet:MODE....1246
ACQuire:ARESet:TIME....1246
ACQuire:AVAilable?....1406
ACQuire:CDTA....1243
ACQuire:COUNt....1245
ACQuire:CURRent?....1378
ACQuire:DRESolution?....2321
ACQuire:INTerpolate....1242
ACQuire:MUWaveform....1243
ACQuire:POINts:AADJust....1239
ACQuire:POINts:ARATe?....1239
ACQuire:POINts:AUTO....1238
ACQuire:POINts:DVALue?....2321
ACQuire:POINts:MAXimum....1239
ACQuire:POINts:ZVALue?....1292
ACQuire:POINts[:VALue]....1241
ACQuire:RESolution....1240
ACQuire:SEGMented:AUToreplay....1247
ACQuire:SEGMented:MAX....1247
ACQuire:SEGMented:STATe....1247
ACQuire:SRATe....1240
ACQuire:SRReal....1240
ACQuire:ZRESolution?....1293
ADVJitter<m>:DCOMposition:CLEBasicmeas....2394
ADVJitter<m>:DCOMposition:CLEJitcomp....2394
ADVJitter<m>:DCOMposition:CLENoisecomp....2395

ADVJitter<m>:DCOMposition:COMPonents<n>:ENABle....2390
ADVJitter<m>:DCOMposition:ENOBu....2406
ADVJitter<m>:DCOMposition:ENP....2405
ADVJitter<m>:DCOMposition:JINTerleaved....2403
ADVJitter<m>:DCOMposition:MRESplength?....2404
ADVJitter<m>:DCOMposition:NHORizontal....2405
ADVJitter<m>:DCOMposition:NVERtical....2405
ADVJitter<m>:DCOMposition:POSStep....2404
ADVJitter<m>:DCOMposition:RESPlength....2404
ADVJitter<m>:DCOMposition:RLEStimate....2404
ADVJitter<m>:DCOMposition:SELBasicmeas....2394
ADVJitter<m>:DCOMposition:SELJitcomp....2394
ADVJitter<m>:DCOMposition:SELNoisecomp....2395
ADVJitter<m>:DCOMposition:SPJLabels....2403
ADVJitter<m>:DCOMposition:SPJTable....2402
ADVJitter<m>:DCOMposition:SPNLabels....2403
ADVJitter<m>:DCOMposition:TARBer....2396
ADVJitter<m>:RESult:BATHtub....2396
ADVJitter<m>:RESult:COMPonents<n>:HISTogram....2390
ADVJitter<m>:RESult:COMPonents<n>:OFFSet....2402
ADVJitter<m>:RESult:COMPonents<n>:SCALe....2401
ADVJitter<m>:RESult:COMPonents<n>:SPECtrum....2391
ADVJitter<m>:RESult:COMPonents<n>:STATistics:MAXPeakpeak?....2392
ADVJitter<m>:RESult:COMPonents<n>:STATistics:MAXStddev?....2392
ADVJitter<m>:RESult:COMPonents<n>:STATistics:MINPeakpeak?....2392
ADVJitter<m>:RESult:COMPonents<n>:STATistics:PEAKpeak?....2392
ADVJitter<m>:RESult:COMPonents<n>:STATistics:STDDev?....2392
ADVJitter<m>:RESult:COMPonents<n>:STATistics:WFMCount?....2392
ADVJitter<m>:RESult:COMPonents<n>:STATistics[:MINStddev]?....2392
ADVJitter<m>:RESult:COMPonents<n>:TRACk....2391
ADVJitter<m>:RESult:ENABle....2389
ADVJitter<m>:RESult:LEVels....2395
ADVJitter<m>:RESult:NBAThtub....2397
ADVJitter<m>:RESult:OBUJ....2397
ADVJitter<m>:RESult:OBUN....2398
ADVJitter<m>:RESult:PERiodic<n>:COUNt?....2392
ADVJitter<m>:RESult:PERiodic<n>:DIRection?....2392
ADVJitter<m>:RESult:PERiodic<n>:VALue?....2393
ADVJitter<m>:RESult:PERiodic<n>[:FREQuency]?....2393
ADVJitter<m>:RESult:PHORizontal....2398
ADVJitter<m>:RESult:PJ....2397
ADVJitter<m>:RESult:PN....2397
ADVJitter<m>:RESult:PVERtical....2398
ADVJitter<m>:RESult:STATistics....2390
ADVJitter<m>:RESult:STEPresponse....2390
ADVJitter<m>:RESult:SYNTheticeye....2398
ADVJitter<m>:RESult:TRANsitions....2395
ADVJitter<m>:RESult:TYPBathtub....2396
ADVJitter<m>:RESult:TYPSynteye....2399
ADVJitter<m>:RESult:UNIT....2391

ADVJitter<m>:RESult:VERTical:BHContinuous ..... 2400
ADVJitter<m>:RESult:VERTical:BMTYpe ..... 2400
ADVJitter<m>:RESult:VERTical:CONTinuous ..... 2400
ADVJitter<m>:RESult:VERTical:HBINs ..... 2399
ADVJitter<m>:RESult:VERTical:JSContinuous ..... 2400
ADVJitter<m>:RESult:VERTical:NHContinuous ..... 2401
ADVJitter<m>:RESult:VERTical:NSContinuous ..... 2401
ADVJitter<m>:SIGNal:BITRate ..... 2380
ADVJitter<m>:SIGNal:CDRMode ..... 2381
ADVJitter<m>:SIGNal:CFEStimation ..... 2382
ADVJitter<m>:SIGNal:CLEStimation ..... 2382
ADVJitter<m>:SIGNal:CLFFactor ..... 2382
ADVJitter<m>:SIGNal:CLFRequency ..... 2382
ADVJitter<m>:SIGNal:CLKSource ..... 2381
ADVJitter<m>:SIGNal:CLOFfset ..... 2383
ADVJitter<m>:SIGNal:DATRate? ..... 2383
ADVJitter<m>:SIGNal:EDGE ..... 2381
ADVJitter<m>:SIGNal:GATE:COUPling ..... 2383
ADVJitter<m>:SIGNal:GATE:CURSor ..... 2384
ADVJitter<m>:SIGNal:GATE:RANGe:ABSolute:STARt ..... 2384
ADVJitter<m>:SIGNal:GATE:RANGe:ABSolute:STOP ..... 2384
ADVJitter<m>:SIGNal:GATE:RANGe:MODE ..... 2385
ADVJitter<m>:SIGNal:GATE:RANGe:RELative:STARt ..... 2385
ADVJitter<m>:SIGNal:GATE:RANGe:RELative:STOP ..... 2385
ADVJitter<m>:SIGNal:GATE:STATe ..... 2386
ADVJitter<m>:SIGNal:GATE:ZDIagram ..... 2385
ADVJitter<m>:SIGNal:QUICkmeas ..... 2380
ADVJitter<m>:SIGNal:REFSource ..... 2381
ADVJitter<m>:SIGNal:SAMTime ..... 2383
ADVJitter<m>:SIGNal:SERStandard ..... 2386
ADVJitter<m>:SIGNal:SOURce ..... 2380
ADVJitter<m>:SIGNal:TYPE ..... 2380
AUToscale ..... 1238
BUS<m>:ARINc:BRMode ..... 1811
BUS<m>:ARINc:BRValue ..... 1811
BUS<m>:ARINc:MAXGap:BITS ..... 1812
BUS<m>:ARINc:MAXGap:SELect ..... 1811
BUS<m>:ARINc:MINGap:BITS ..... 1812
BUS<m>:ARINc:MINGap:SELect ..... 1812
BUS<m>:ARINc:POLarity ..... 1813
BUS<m>:ARINc:PRESet ..... 1813
BUS<m>:ARINc:SOURce ..... 1811
BUS<m>:ARINc:THReshold:HIGH ..... 1813
BUS<m>:ARINc:THReshold:LOW ..... 1813
BUS<m>:ARINc:WCOunt? ..... 1818
BUS<m>:ARINc:WORD<n>:DATA? ..... 1818
BUS<m>:ARINc:WORD<n>:LABel? ..... 1819
BUS<m>:ARINc:WORD<n>:PATTern? ..... 1819
BUS<m>:ARINc:WORD<n>:SDI? ..... 1819
BUS<m>:ARINc:WORD<n>:SSM? ..... 1820

BUS<m>:ARINc:WORD<n>:STARt? 1820
BUS<m>:ARINc:WORD<n>:STATe? 1820
BUS<m>:ARINc:WORD<n>:STOP? 1821
BUS<m>:ARINc:WORD<n>:SYMBol? 1821
BUS<m>:CAN:BITRate 1710
BUS<m>:CAN:DATA:SOURce 1708
BUS<m>:CAN:DATA:THReshold 1710
BUS<m>:CAN:FCOunt? 1723
BUS<m>:CAN:FDATa:DBITrate 1711
BUS<m>:CAN:FDATa:ENABle 1711
BUS<m>:CAN:FDATa:FRAMe<n>:SCValue? 1730
BUS<m>:CAN:FDATa:FRAMe<n>:STANdard? 1723
BUS<m>:CAN:FDATa:JWIDth 1713
BUS<m>:CAN:FDATa:PSTandard 1709
BUS<m>:CAN:FDATa:SAMPlepoint 1711
BUS<m>:CAN:FDATa:T1Segment 1712
BUS<m>:CAN:FDATa:T2Segment 1712
BUS<m>:CAN:FRAMe<n>:ACKState? 1726
BUS<m>:CAN:FRAMe<n>:ACKValue? 1726
BUS<m>:CAN:FRAMe<n>:BITRate? 1727
BUS<m>:CAN:FRAMe<n>:BSEPosition? 1729
BUS<m>:CAN:FRAMe<n>:BYTE<o>:STATe? 1730
BUS<m>:CAN:FRAMe<n>:BYTE<o>:VALue? 1730
BUS<m>:CAN:FRAMe<n>:CSSTate? 1726
BUS<m>:CAN:FRAMe<n>:CSValue? 1727
BUS<m>:CAN:FRAMe<n>:DATA? 1726
BUS<m>:CAN:FRAMe<n>:DLCState? 1726
BUS<m>:CAN:FRAMe<n>:DLCValue? 1728
BUS<m>:CAN:FRAMe<n>:FERCause? 1729
BUS<m>:CAN:FRAMe<n>:IDSTate? 1726
BUS<m>:CAN:FRAMe<n>:IDTYpe? 1728
BUS<m>:CAN:FRAMe<n>:IDValue? 1728
BUS<m>:CAN:FRAMe<n>:NDBYtes? 1724
BUS<m>:CAN:FRAMe<n>:SDATa? 1749
BUS<m>:CAN:FRAMe<n>:SDEXport? 1729
BUS<m>:CAN:FRAMe<n>:STARt? 1725
BUS<m>:CAN:FRAMe<n>:STATus? 1723
BUS<m>:CAN:FRAMe<n>:STOP? 1725
BUS<m>:CAN:FRAMe<n>:SYMBol? 1725
BUS<m>:CAN:FRAMe<n>:TYPE? 1725
BUS<m>:CAN:JWIDth 1713
BUS<m>:CAN:SAMPlepoint 1711
BUS<m>:CAN:T1Segment 1712
BUS<m>:CAN:T2Segment 1712
BUS<m>:CAN:TECHnology 1710
BUS<m>:CAN:TYPE 1709
BUS<m>:CDR:BITRate 2247
BUS<m>:CDR:PLL:BWIDth 2247
BUS<m>:CDR:PLL:DAMPing 2247
BUS<m>:CDR:PLL:ORDer 2248

BUS<m>:CDR:PLL:RELBwidth ... 2248
BUS<m>:CDR:RESults ... 2248
BUS<m>:CDR:SYNC ... 2248
BUS<m>:CMSB:ADDFrame ... 2054
BUS<m>:CMSB:BITRate:ENABle ... 2053
BUS<m>:CMSB:BITRate:VALue ... 2054
BUS<m>:CMSB:CLR ... 2055
BUS<m>:CMSB:CODing ... 2044
BUS<m>:CMSB:EXRBits ... 2058
BUS<m>:CMSB:FCOunt? ... 2055
BUS<m>:CMSB:FILTer:ENABle ... 2066
BUS<m>:CMSB:FILTer:ERENable ... 2067
BUS<m>:CMSB:FILTer:ERRor<n>:ENABle ... 2067
BUS<m>:CMSB:FILTer:FIENable ... 2068
BUS<m>:CMSB:FILTer:FRAMe<n>:ENABle ... 2067
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:BIT ... 2068
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:DMAX ... 2068
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:DMIN ... 2069
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:DOPerator ... 2069
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:ENABle ... 2069
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:IMAX ... 2070
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:IMIN ... 2070
BUS<m>:CMSB:FILTer:FRAMe<n>:FLD<o>:IOPerator ... 2070
BUS<m>:CMSB:FILTer:FRENable ... 2068
BUS<m>:CMSB:FRAMe<n>:APPend ... 2055
BUS<m>:CMSB:FRAMe<n>:CCOunt? ... 2056
BUS<m>:CMSB:FRAMe<n>:CELL<o>:BITCount ... 2056
BUS<m>:CMSB:FRAMe<n>:CELL<o>:BITorder ... 2057
BUS<m>:CMSB:FRAMe<n>:CELL<o>:CLMN ... 2058
BUS<m>:CMSB:FRAMe<n>:CELL<o>:CONDition ... 2057
BUS<m>:CMSB:FRAMe<n>:CELL<o>:CRGB ... 2058
BUS<m>:CMSB:FRAMe<n>:CELL<o>:FORMat ... 2057
BUS<m>:CMSB:FRAMe<n>:CELL<o>:NAME ... 2056
BUS<m>:CMSB:FRAMe<n>:TYPE ... 2055
BUS<m>:CMSB:GAPTime:ENABle ... 2054
BUS<m>:CMSB:GAPTime:VALue ... 2054
BUS<m>:CMSB:LOAD ... 2059
BUS<m>:CMSB:MANChester:CPHase ... 2047
BUS<m>:CMSB:MANChester:DATA ... 2045
BUS<m>:CMSB:MANChester:POLarity ... 2045
BUS<m>:CMSB:MANChester:THReshold:COUPling ... 2046
BUS<m>:CMSB:MANChester:THReshold:HIGH ... 2045
BUS<m>:CMSB:MANChester:THReshold:LOW ... 2046
BUS<m>:CMSB:MANChester:THReshold:PRESet ... 2046
BUS<m>:CMSB:NRZ:CLCK ... 2047
BUS<m>:CMSB:NRZ:CPHase ... 2049
BUS<m>:CMSB:NRZ:CPOLarity ... 2049
BUS<m>:CMSB:NRZ:DATA ... 2048
BUS<m>:CMSB:NRZ:ENAPolarity ... 2050
BUS<m>:CMSB:NRZ:ENBLe ... 2050

BUS<m>:CMSB:NRZ:HYSTeresis:CLCK ..... 2051
BUS<m>:CMSB:NRZ:HYSTeresis:DATA ..... 2051
BUS<m>:CMSB:NRZ:HYSTeresis:ENBLe ..... 2052
BUS<m>:CMSB:NRZ:IDLPolarity ..... 2048
BUS<m>:CMSB:NRZ:POLarity ..... 2050
BUS<m>:CMSB:NRZ:THReshold:CLCK ..... 2051
BUS<m>:CMSB:NRZ:THReshold:COUPling ..... 2053
BUS<m>:CMSB:NRZ:THReshold:DATA ..... 2051
BUS<m>:CMSB:NRZ:THReshold:ENBLe ..... 2052
BUS<m>:CMSB:NRZ:THReshold:PRESet ..... 2052
BUS<m>:CMSB:RCOunt? ..... 2072
BUS<m>:CMSB:RESult<n>:CCOunt? ..... 2075
BUS<m>:CMSB:RESult<n>:CELL<o>:NAME? ..... 2075
BUS<m>:CMSB:RESult<n>:CELL<o>:STARt? ..... 2075
BUS<m>:CMSB:RESult<n>:CELL<o>:STATe? ..... 2076
BUS<m>:CMSB:RESult<n>:CELL<o>:STOP? ..... 2075
BUS<m>:CMSB:RESult<n>:CELL<o>:VALue? ..... 2077
BUS<m>:CMSB:RESult<n>:CONE? ..... 2074
BUS<m>:CMSB:RESult<n>:CTHRee? ..... 2074
BUS<m>:CMSB:RESult<n>:CTWO? ..... 2074
BUS<m>:CMSB:RESult<n>:STARt? ..... 2073
BUS<m>:CMSB:RESult<n>:STATe? ..... 2072
BUS<m>:CMSB:RESult<n>:STOP? ..... 2073
BUS<m>:CMSB:RESult<n>:TYPE? ..... 2073
BUS<m>:CMSB:SAVE ..... 2059
BUS<m>:CTLE:DCGain ..... 2164
BUS<m>:CTLE:DCGain ..... 2249
BUS<m>:CTLE:ENBLe ..... 2164
BUS<m>:CTLE:ENBLe ..... 2249
BUS<m>:CTLE:P1FReq ..... 2165
BUS<m>:CTLE:P1FReq ..... 2250
BUS<m>:CTLE:P2FReq ..... 2165
BUS<m>:CTLE:P2FReq ..... 2250
BUS<m>:CTLE:Z1FReq ..... 2165
BUS<m>:CTLE:Z1FReq ..... 2250
BUS<m>:DDR:BITRate ..... 2296
BUS<m>:DDR:DATA:HYSTeresis ..... 2294
BUS<m>:DDR:DATA:SOURce ..... 2294
BUS<m>:DDR:DATA:THReshold ..... 2294
BUS<m>:DDR:RESult:FCOunt? ..... 2296
BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:FVAL? ..... 2297
BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:STARt? ..... 2297
BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:STATus? ..... 2297
BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:STOP? ..... 2298
BUS<m>:DDR:RESult:FRAMe<n>:FLD<o>:VAL? ..... 2298
BUS<m>:DDR:RESult:FRAMe<n>:NWRDs? ..... 2299
BUS<m>:DDR:RESult:FRAMe<n>:STARt? ..... 2299
BUS<m>:DDR:RESult:FRAMe<n>:STATe? ..... 2299
BUS<m>:DDR:RESult:FRAMe<n>:STOP? ..... 2300
BUS<m>:DDR:RESult:FRAMe<n>:TYPE? ..... 2300

BUS<m>:DDR:STANdard....2294
BUS<m>:DDR:STRBe:HYSTeresis....2295
BUS<m>:DDR:STRBe:SOURce....2295
BUS<m>:DDR:STRBe:THReshold....2295
BUS<m>:DPHY:CONFig....1950
BUS<m>:DPHY:CP:HSPeed:HYSTeresis....1954
BUS<m>:DPHY:CP:HSPeed:THReshold....1954
BUS<m>:DPHY:CP:PROBe....1950
BUS<m>:DPHY:CP:SOURce....1950
BUS<m>:DPHY:DNZero:LPOWer:THLower....1953
BUS<m>:DPHY:DNZero:LPOWer:THUPper....1952
BUS<m>:DPHY:DNZero:PROBe....1953
BUS<m>:DPHY:DNZero:SOURce....1953
BUS<m>:DPHY:DPONe:HSPeed:HYSTeresis....1954
BUS<m>:DPHY:DPONe:HSPeed:THReshold....1954
BUS<m>:DPHY:DPONe:PROBe....1954
BUS<m>:DPHY:DPONe:SOURce....1953
BUS<m>:DPHY:DPTHree:HSPeed:HYSTeresis....1954
BUS<m>:DPHY:DPTHree:HSPeed:THReshold....1954
BUS<m>:DPHY:DPTHree:PROBe....1954
BUS<m>:DPHY:DPTHree:SOURce....1953
BUS<m>:DPHY:DPTWo:HSPeed:HYSTeresis....1954
BUS<m>:DPHY:DPTWo:HSPeed:THReshold....1954
BUS<m>:DPHY:DPTWo:PROBe....1954
BUS<m>:DPHY:DPTWo:SOURce....1953
BUS<m>:DPHY:DPZero:HSPeed:HYSTeresis....1954
BUS<m>:DPHY:DPZero:HSPeed:THReshold....1954
BUS<m>:DPHY:DPZero:LPOWer:THLower....1953
BUS<m>:DPHY:DPZero:LPOWer:THUPper....1952
BUS<m>:DPHY:DPZero:PROBe....1953
BUS<m>:DPHY:DPZero:SOURce....1953
BUS<m>:DPHY:DRATe....1951
BUS<m>:DPHY:DSPData....1951
BUS<m>:DPHY:RESult:FCOunt?....1963
BUS<m>:DPHY:RESult:FRAMe<n>:CS?....1963
BUS<m>:DPHY:RESult:FRAMe<n>:DATA?....1963
BUS<m>:DPHY:RESult:FRAMe<n>:DTName?....1964
BUS<m>:DPHY:RESult:FRAMe<n>:DTYPe?....1964
BUS<m>:DPHY:RESult:FRAMe<n>:ECC?....1965
BUS<m>:DPHY:RESult:FRAMe<n>:NUMPackets?....1968
BUS<m>:DPHY:RESult:FRAMe<n>:PACKet<o>:IDX?....1965
BUS<m>:DPHY:RESult:FRAMe<n>:PACKet<o>:VALue?....1965
BUS<m>:DPHY:RESult:FRAMe<n>:STARt?....1966
BUS<m>:DPHY:RESult:FRAMe<n>:STATe?....1966
BUS<m>:DPHY:RESult:FRAMe<n>:STOP?....1967
BUS<m>:DPHY:RESult:FRAMe<n>:TYPE?....1967
BUS<m>:DPHY:RESult:FRAMe<n>:VCHannel?....1968
BUS<m>:DPHY:SMLanes....1955
BUS<m>:DPHY:THCoupling....1951
BUS<m>:DPHY:THPReset....1952

BUS<m>:DPHY:VARiant....1952
BUS<m>:EBTB:BITDetermi....2093
BUS<m>:EBTB:BITRate....2089
BUS<m>:EBTB:DIFFerential:SOURce....2089
BUS<m>:EBTB:DIFFerential:THRHigh....2089
BUS<m>:EBTB:DIFFerential:THRLow....2090
BUS<m>:EBTB:DISF....2090
BUS<m>:EBTB:DMINus:SOURce....2090
BUS<m>:EBTB:DMINus:THReshold....2091
BUS<m>:EBTB:DPLus:SOURce....2091
BUS<m>:EBTB:DPLus:THReshold....2091
BUS<m>:EBTB:EBTRate?....2089
BUS<m>:EBTB:FAUToscale....2093
BUS<m>:EBTB:FCSY....2092
BUS<m>:EBTB:SCOunt?....2092
BUS<m>:EBTB:SCSY....2092
BUS<m>:EBTB:SYMBol<n>:BYTE?....2098
BUS<m>:EBTB:SYMBol<n>:DATA?....2096
BUS<m>:EBTB:SYMBol<n>:KDCode?....2098
BUS<m>:EBTB:SYMBol<n>:STARt?....2097
BUS<m>:EBTB:SYMBol<n>:STATus?....2097
BUS<m>:EBTB:SYMBol<n>:STOP?....2098
BUS<m>:EBTB:SYMBol<n>:TENBit?....2099
BUS<m>:EBTB:SYNC....2092
BUS<m>:EBTB:TYPE....2093
BUS<m>:EBTB:USCS....2093
BUS<m>:ETHernet:BITRate....1830
BUS<m>:ETHernet:POLarity....1828
BUS<m>:ETHernet:PRESet....1829
BUS<m>:ETHernet:SOURce....1828
BUS<m>:ETHernet:THReshold:HIGH....1829
BUS<m>:ETHernet:THReshold:LOW....1829
BUS<m>:ETHernet:VARiant....1828
BUS<m>:ETHernet:WCOunt?....1837
BUS<m>:ETHernet:WORD<n>:BITRate?....1841
BUS<m>:ETHernet:WORD<n>:BYTE<o>:VALue?....1841
BUS<m>:ETHernet:WORD<n>:CRC?....1840
BUS<m>:ETHernet:WORD<n>:DATA?....1840
BUS<m>:ETHernet:WORD<n>:DESTaddress?....1839
BUS<m>:ETHernet:WORD<n>:DSYMbol?....1840
BUS<m>:ETHernet:WORD<n>:FTYPe?....1837
BUS<m>:ETHernet:WORD<n>:NUMWords?....1842
BUS<m>:ETHernet:WORD<n>:SRCaddress?....1839
BUS<m>:ETHernet:WORD<n>:SSYMbol?....1841
BUS<m>:ETHernet:WORD<n>:STARt?....1838
BUS<m>:ETHernet:WORD<n>:STATe?....1838
BUS<m>:ETHernet:WORD<n>:STOP?....1838
BUS<m>:ETHernet:WORD<n>:TYPE?....1839
BUS<m>:EXPResult:DETail....1640
BUS<m>:EXPResult:SAVE....1640

BUS<m>:EXPResult:TIME ... 1641
BUS<m>:FAUToset ... 1638
BUS<m>:FORMat ... 1639
BUS<m>:HBTO:ATTN ... 1853
BUS<m>:HBTO:FDAM ... 1853
BUS<m>:HBTO:FDAP ... 1853
BUS<m>:HBTO:FDIF ... 1853
BUS<m>:HBTO:FTYP ... 1854
BUS<m>:HBTO:MODE ... 1855
BUS<m>:HBTO:POLarity ... 1855
BUS<m>:HBTO:RDAM ... 1854
BUS<m>:HBTO:RDAP ... 1854
BUS<m>:HBTO:RDIF ... 1854
BUS<m>:HBTO:RESult:FCOunt? ... 1864
BUS<m>:HBTO:RESult:FRAMe<n>:BITRate? ... 1868
BUS<m>:HBTO:RESult:FRAMe<n>:CRC? ... 1867
BUS<m>:HBTO:RESult:FRAMe<n>:DATA? ... 1867
BUS<m>:HBTO:RESult:FRAMe<n>:DESTaddress? ... 1866
BUS<m>:HBTO:RESult:FRAMe<n>:DSYMbol? ... 1868
BUS<m>:HBTO:RESult:FRAMe<n>:FTYPe? ... 1865
BUS<m>:HBTO:RESult:FRAMe<n>:NUMWords? ... 1868
BUS<m>:HBTO:RESult:FRAMe<n>:SRCaddress? ... 1867
BUS<m>:HBTO:RESult:FRAMe<n>:SSYMbol? ... 1868
BUS<m>:HBTO:RESult:FRAMe<n>:STARt? ... 1866
BUS<m>:HBTO:RESult:FRAMe<n>:STATe? ... 1865
BUS<m>:HBTO:RESult:FRAMe<n>:STOP? ... 1866
BUS<m>:HBTO:RESult:FRAMe<n>:TYPE? ... 1867
BUS<m>:HBTO:RESult:FRAMe<n>:WORD<o>:TYPE? ... 1869
BUS<m>:HBTO:RESult:FRAMe<n>:WORD<o>:VALue? ... 1869
BUS<m>:HBTO:RESult:FRAMe<n>:WORD<o>:VSTR? ... 1869
BUS<m>:HBTO:RTYP ... 1855
BUS<m>:HBTO:SYMRate ... 1856
BUS<m>:HBTO:THReshold ... 1855
BUS<m>:I2C:FCOunt? ... 1653
BUS<m>:I2C:FRAMe<n>:AACCess? ... 1653
BUS<m>:I2C:FRAMe<n>:ACCess? ... 1654
BUS<m>:I2C:FRAMe<n>:ACOMplete? ... 1654
BUS<m>:I2C:FRAMe<n>:ADBStart? ... 1654
BUS<m>:I2C:FRAMe<n>:ADDRess? ... 1655
BUS<m>:I2C:FRAMe<n>:ADEVice? ... 1655
BUS<m>:I2C:FRAMe<n>:AMODe? ... 1656
BUS<m>:I2C:FRAMe<n>:ASTart? ... 1656
BUS<m>:I2C:FRAMe<n>:BCOunt? ... 1658
BUS<m>:I2C:FRAMe<n>:BITRate? ... 1656
BUS<m>:I2C:FRAMe<n>:BYTE<o>:ACCess? ... 1659
BUS<m>:I2C:FRAMe<n>:BYTE<o>:ACKStart? ... 1659
BUS<m>:I2C:FRAMe<n>:BYTE<o>:COMPlete? ... 1660
BUS<m>:I2C:FRAMe<n>:BYTE<o>:STARt? ... 1660
BUS<m>:I2C:FRAMe<n>:BYTE<o>:VALue? ... 1660
BUS<m>:I2C:FRAMe<n>:DATA? ... 1653

BUS<m>:I2C:FRAMe<n>:RWBStart? 1657
BUS<m>:I2C:FRAMe<n>:STARt? 1658
BUS<m>:I2C:FRAMe<n>:STATus? 1657
BUS<m>:I2C:FRAMe<n>:STOP? 1658
BUS<m>:I2C:FRAMe<n>:SYMBol? 1658
BUS<m>:I2C:RWBit 1645
BUS<m>:I2C:SCL:SOURce 1643
BUS<m>:I2C:SCL:THReshold 1644
BUS<m>:I2C:SDA:SOURce 1644
BUS<m>:I2C:SDA:THReshold 1645
BUS<m>:I2C:TECHnology 1645
BUS<m>:LIN:BITRate 1754
BUS<m>:LIN:DATA:SOURce 1752
BUS<m>:LIN:DATA:THReshold 1753
BUS<m>:LIN:FCOunt? 1762
BUS<m>:LIN:FRAMe<n>:BITRate? 1767
BUS<m>:LIN:FRAMe<n>:BYTE<o>:STATe? 1768
BUS<m>:LIN:FRAMe<n>:BYTE<o>:VALue? 1768
BUS<m>:LIN:FRAMe<n>:CSSTate? 1767
BUS<m>:LIN:FRAMe<n>:CSValue? 1767
BUS<m>:LIN:FRAMe<n>:DATA? 1764
BUS<m>:LIN:FRAMe<n>:IDPValue? 1766
BUS<m>:LIN:FRAMe<n>:IDSTate? 1765
BUS<m>:LIN:FRAMe<n>:IDValue? 1765
BUS<m>:LIN:FRAMe<n>:SDATa? 1763
BUS<m>:LIN:FRAMe<n>:SDEXport? 1763
BUS<m>:LIN:FRAMe<n>:STARt? 1763
BUS<m>:LIN:FRAMe<n>:STATus? 1762
BUS<m>:LIN:FRAMe<n>:STOP? 1763
BUS<m>:LIN:FRAMe<n>:SYMBol? 1764
BUS<m>:LIN:FRAMe<n>:SYSTate? 1766
BUS<m>:LIN:FRAMe<n>:VERSion? 1764
BUS<m>:LIN:POLarity 1754
BUS<m>:LIN:STANdard 1754
BUS<m>:LIN:TECHnology 1753
BUS<m>:MDIO:CLOCk:SOURce 2104
BUS<m>:MDIO:CLOCk:THReshold:HIGH 2105
BUS<m>:MDIO:CLOCk:THReshold:LOW 2105
BUS<m>:MDIO:COUPling 2107
BUS<m>:MDIO:DATA:SOURce 2105
BUS<m>:MDIO:DATA:THReshold:HIGH 2106
BUS<m>:MDIO:DATA:THReshold:LOW 2106
BUS<m>:MDIO:PRESet 2106
BUS<m>:MDIO:WCOunt? 2110
BUS<m>:MDIO:WORD<n>:DATA? 2110
BUS<m>:MDIO:WORD<n>:PHYS? 2110
BUS<m>:MDIO:WORD<n>:REGI? 2111
BUS<m>:MDIO:WORD<n>:ST? 2111
BUS<m>:MDIO:WORD<n>:STARt? 2111
BUS<m>:MDIO:WORD<n>:STATe? 2112

BUS<m>:MDIO:WORD<n>:STOP? 2112
BUS<m>:MDIO:WORD<n>:SYMBol? 2113
BUS<m>:MDIO:WORD<n>:TYPE? 2113
BUS<m>:MILStd:MAXResponse:BITS 1783
BUS<m>:MILStd:MAXResponse:SELect 1784
BUS<m>:MILStd:MINGap:BITS 1784
BUS<m>:MILStd:MINGap:SELect 1784
BUS<m>:MILStd:POLarity 1784
BUS<m>:MILStd:PRESet 1785
BUS<m>:MILStd:SOURce 1783
BUS<m>:MILStd:THReshold:HIGH 1785
BUS<m>:MILStd:THReshold:LOW 1785
BUS<m>:MILStd:WCOunt? 1797
BUS<m>:MILStd:WORD<n>:DATA? 1797
BUS<m>:MILStd:WORD<n>:INFO? 1798
BUS<m>:MILStd:WORD<n>:RTADdress? 1798
BUS<m>:MILStd:WORD<n>:STARt? 1798
BUS<m>:MILStd:WORD<n>:STATus? 1799
BUS<m>:MILStd:WORD<n>:STOP? 1799
BUS<m>:MILStd:WORD<n>:SYMBol? 1800
BUS<m>:MILStd:WORD<n>:TYPE? 1800
BUS<m>:MPHY:DLANes 1983
BUS<m>:MPHY:DONE:SOURce 1984
BUS<m>:MPHY:DONE:THReshold 1984
BUS<m>:MPHY:DTHRee:SOURce 1984
BUS<m>:MPHY:DTHRee:THReshold 1984
BUS<m>:MPHY:DTWO:SOURce 1984
BUS<m>:MPHY:DTWO:THReshold 1984
BUS<m>:MPHY:DZERo:SOURce 1983
BUS<m>:MPHY:DZERo:THReshold 1984
BUS<m>:MPHY:RESult:FCOunt? 2004
BUS<m>:MPHY:RESult:FRAMe<n>:CCOunt? 2004
BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:DATA? 2005
BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:STARt? 2005
BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:STATe? 2006
BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:STOP? 2006
BUS<m>:MPHY:RESult:FRAMe<n>:CELL<o>:TYPE? 2006
BUS<m>:MPHY:RESult:FRAMe<n>:DATA? 2007
BUS<m>:MPHY:RESult:FRAMe<n>:FTYPe? 2002
BUS<m>:MPHY:RESult:FRAMe<n>:INFO? 2007
BUS<m>:MPHY:RESult:FRAMe<n>:SSFT? 2007
BUS<m>:MPHY:RESult:FRAMe<n>:STARt? 2008
BUS<m>:MPHY:RESult:FRAMe<n>:STATe? 2003
BUS<m>:MPHY:RESult:FRAMe<n>:STOP? 2008
BUS<m>:MPHY:SCRMode 1985
BUS<m>:MPHY:SMLanes 1985
BUS<m>:MPHY:THCoupling 1984
BUS<m>:MPHY:THPReset 1985
BUS<m>:MPHY:VARiant 1983
BUS<m>:NEWList 1639

BUS<m>:PARallel:BIT<n>:DESKew....2316
BUS<m>:PARallel:BIT<n>:LABel....2317
BUS<m>:PARallel:BIT<n>[:STATe]....2314
BUS<m>:PARallel:CLEar....2319
BUS<m>:PARallel:CLOCk....2319
BUS<m>:PARallel:CLON....2318
BUS<m>:PARallel:CLSLope....2319
BUS<m>:PARallel:DATA:FORMat....2317
BUS<m>:PARallel:DATA:FORMat....2333
BUS<m>:PARallel:DATA:HEADer?....2333
BUS<m>:PARallel:DATA[:VALues]?....2333
BUS<m>:PARallel:DECTable<n>:COUNt?....2320
BUS<m>:PARallel:DECTable<n>:DATA?....2320
BUS<m>:PARallel:DECTable<n>:SHOW....2320
BUS<m>:PARallel:DESoffset....2316
BUS<m>:PARallel:DISPlay:BTYP....2318
BUS<m>:PARallel:DISPlay:SHBU....2318
BUS<m>:PARallel:DISPlay:SHDI....2317
BUS<m>:PARallel:HYSTeresis<n>....2315
BUS<m>:PARallel:STATe....2313
BUS<m>:PARallel:TECHnology....2315
BUS<m>:PARallel:THCoupling....2315
BUS<m>:PARallel:THReshold<n>....2314
BUS<m>:PCIE:DFE:AUTO....2251
BUS<m>:PCIE:DFE:COEFficient....2251
BUS<m>:PCIE:DFE:ENBLe....2251
BUS<m>:PCIE:DFE:TCONt....2251
BUS<m>:PCIE:DSCRambling....2244
BUS<m>:PCIE:FIDLe....2244
BUS<m>:PCIE:GEN....2245
BUS<m>:PCIE:LNKW....2245
BUS<m>:PCIE:LONE:SOURce....2246
BUS<m>:PCIE:LONE:THRHigh....2246
BUS<m>:PCIE:LONE:THRLow....2246
BUS<m>:PCIE:LTHRee:SOURce....2246
BUS<m>:PCIE:LTHRee:THRHigh....2246
BUS<m>:PCIE:LTHRee:THRLow....2246
BUS<m>:PCIE:LTWO:SOURce....2246
BUS<m>:PCIE:LTWO:THRHigh....2246
BUS<m>:PCIE:LTWO:THRLow....2246
BUS<m>:PCIE:LZER:SOURce....2245
BUS<m>:PCIE:LZER:THRHigh....2246
BUS<m>:PCIE:LZER:THRLow....2246
BUS<m>:PCIE:RESult:PACKet<n>:ADDR?....2263
BUS<m>:PCIE:RESult:PACKet<n>:BITRate?....2271
BUS<m>:PCIE:RESult:PACKet<n>:CPID?....2264
BUS<m>:PCIE:RESult:PACKet<n>:CPS?....2264
BUS<m>:PCIE:RESult:PACKet<n>:DATA?....2264
BUS<m>:PCIE:RESult:PACKet<n>:DFC?....2265
BUS<m>:PCIE:RESult:PACKet<n>:FCOunt?....2265

BUS<m>:PCIE:RESult:PACKet<n>:FLD<o>:FVAL? 2265
BUS<m>:PCIE:RESult:PACKet<n>:FLD<o>:STATus? 2266
BUS<m>:PCIE:RESult:PACKet<n>:FLD<o>:TYPE? 2267
BUS<m>:PCIE:RESult:PACKet<n>:FLD<o>:VAL? 2267
BUS<m>:PCIE:RESult:PACKet<n>:HFC? 2267
BUS<m>:PCIE:RESult:PACKet<n>:LEN? 2268
BUS<m>:PCIE:RESult:PACKet<n>:RQID? 2268
BUS<m>:PCIE:RESult:PACKet<n>:SEQN? 2268
BUS<m>:PCIE:RESult:PACKet<n>:STARt? 2268
BUS<m>:PCIE:RESult:PACKet<n>:STATe? 2269
BUS<m>:PCIE:RESult:PACKet<n>:STOP? 2269
BUS<m>:PCIE:RESult:PACKet<n>:TAG? 2270
BUS<m>:PCIE:RESult:PACKet<n>:TYPE? 2270
BUS<m>:PCIE:RESult:PCOunt? 2271
BUS<m>:PCIE:SMLanes 2246
BUS<m>:RESult 1638
BUS<m>:RFFE:CLOCk:SOURce 1917
BUS<m>:RFFE:CLOCk:THReshold:HIGH 1919
BUS<m>:RFFE:CLOCk:THReshold:HYSTeresis 1919
BUS<m>:RFFE:CLOCk:THReshold:LOW 1919
BUS<m>:RFFE:COUPling 1918
BUS<m>:RFFE:DATA:SOURce 1917
BUS<m>:RFFE:DATA:THReshold:HIGH 1918
BUS<m>:RFFE:DATA:THReshold:HYSTeresis 1918
BUS<m>:RFFE:DATA:THReshold:LOW 1918
BUS<m>:RFFE:FCOunt? 1932
BUS<m>:RFFE:GFILter 1920
BUS<m>:RFFE:GFWidth 1920
BUS<m>:RFFE:MINGap:SELect 1920
BUS<m>:RFFE:MINGap:TIME 1921
BUS<m>:RFFE:PRESet 1917
BUS<m>:RFFE:RDMD 1921
BUS<m>:RFFE:SEQuence<n>:ADDRess? 1934
BUS<m>:RFFE:SEQuence<n>:BCOunt? 1935
BUS<m>:RFFE:SEQuence<n>:BYTE<o>:NAME? 1937
BUS<m>:RFFE:SEQuence<n>:BYTE<o>:STATe? 1937
BUS<m>:RFFE:SEQuence<n>:BYTE<o>:VALue? 1938
BUS<m>:RFFE:SEQuence<n>:DATA? 1935
BUS<m>:RFFE:SEQuence<n>:PADone? 1936
BUS<m>:RFFE:SEQuence<n>:PADZero? 1936
BUS<m>:RFFE:SEQuence<n>:PCTRl? 1936
BUS<m>:RFFE:SEQuence<n>:RBTRate? 1936
BUS<m>:RFFE:SEQuence<n>:SADD? 1934
BUS<m>:RFFE:SEQuence<n>:STARt? 1933
BUS<m>:RFFE:SEQuence<n>:STATe? 1932
BUS<m>:RFFE:SEQuence<n>:STOP? 1933
BUS<m>:RFFE:SEQuence<n>:SYMBol? 1935
BUS<m>:RFFE:SEQuence<n>:TYPE? 1933
BUS<m>:RFFE:SEQuence<n>:WBTRate? 1937
BUS<m>:RFFE:VERSion 1921

BUS<m>:SETReflevels....1638
BUS<m>:SPI:BORDer....1678
BUS<m>:SPI:FCOunt?....1686
BUS<m>:SPI:FRAMe<n>:BITRate?....1686
BUS<m>:SPI:FRAMe<n>:COUNt?....1686
BUS<m>:SPI:FRAMe<n>:DATA?....1685
BUS<m>:SPI:FRAMe<n>:STARt?....1687
BUS<m>:SPI:FRAMe<n>:STATus?....1686
BUS<m>:SPI:FRAMe<n>:STOP?....1687
BUS<m>:SPI:FRAMe<n>:WCOunt?....1688
BUS<m>:SPI:FRAMe<n>:WORD<o>:FMISo?....1689
BUS<m>:SPI:FRAMe<n>:WORD<o>:FMOSi?....1690
BUS<m>:SPI:FRAMe<n>:WORD<o>:MISO?....1689
BUS<m>:SPI:FRAMe<n>:WORD<o>:MOSI?....1690
BUS<m>:SPI:FRAMe<n>:WORD<o>:STARt?....1688
BUS<m>:SPI:FRAMe<n>:WORD<o>:STOP?....1688
BUS<m>:SPI:FRCondition....1681
BUS<m>:SPI:MISO:POLarity....1680
BUS<m>:SPI:MISO:SOURce....1679
BUS<m>:SPI:MISO:THReshold....1681
BUS<m>:SPI:MOSI:POLarity....1681
BUS<m>:SPI:MOSI:SOURce....1680
BUS<m>:SPI:MOSI:THReshold....1681
BUS<m>:SPI:SCLK:SOURce....1678
BUS<m>:SPI:SCLK:THReshold....1681
BUS<m>:SPI:SSELect:POLarity....1679
BUS<m>:SPI:SSELect:SOURce....1679
BUS<m>:SPI:SSELect:THReshold....1681
BUS<m>:SPI:TIMeout....1682
BUS<m>:SPI:WSIZe....1678
BUS<m>:SWIRe:BPOSition....2226
BUS<m>:SWIRe:COUPling....2228
BUS<m>:SWIRe:DATA:HYSTeresis....2226
BUS<m>:SWIRe:DATA:SOURce....2226
BUS<m>:SWIRe:DATA:THReshold....2227
BUS<m>:SWIRe:MGAP....2227
BUS<m>:SWIRe:PRESet....2229
BUS<m>:SWIRe:RESults:FCOunt....2233
BUS<m>:SWIRe:RESults:FRAMe<n>:DATA?....2233
BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:FVAL?....2235
BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:STARt?....2235
BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:STATus?....2236
BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:STOP?....2236
BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:TYPE?....2237
BUS<m>:SWIRe:RESults:FRAMe<n>:FLD<o>:VAL?....2237
BUS<m>:SWIRe:RESults:FRAMe<n>:STARt?....2233
BUS<m>:SWIRe:RESults:FRAMe<n>:STATe?....2234
BUS<m>:SWIRe:RESults:FRAMe<n>:STOP?....2234
BUS<m>:SWIRe:RESults:FRAMe<n>:TYPE?....2234
BUS<m>:SWIRe:STRBe:HYSTeresis....2227

BUS<m>:SWIRe:STRBe:SOURce....2228
BUS<m>:SWIRe:STRBe:THReshold....2228
BUS<m>:SWIRe:SYSLect....2228
BUS<m>:SYMBols....1639
BUS<m>:TBTO:ATTN....1883
BUS<m>:TBTO:FDAM....1884
BUS<m>:TBTO:FDAP....1884
BUS<m>:TBTO:FDIF....1884
BUS<m>:TBTO:FTYP....1885
BUS<m>:TBTO:MODE....1886
BUS<m>:TBTO:POLarity....1886
BUS<m>:TBTO:RDAM....1885
BUS<m>:TBTO:RDAP....1885
BUS<m>:TBTO:RDIF....1885
BUS<m>:TBTO:RESult:FCOunt?....1887
BUS<m>:TBTO:RESult:FRAMe<n>:BITRate?....1887
BUS<m>:TBTO:RESult:FRAMe<n>:CRC?....1888
BUS<m>:TBTO:RESult:FRAMe<n>:DATA?....1888
BUS<m>:TBTO:RESult:FRAMe<n>:DESTaddress?....1888
BUS<m>:TBTO:RESult:FRAMe<n>:DSYMbol?....1889
BUS<m>:TBTO:RESult:FRAMe<n>:FTYPe?....1889
BUS<m>:TBTO:RESult:FRAMe<n>:NUMWords?....1889
BUS<m>:TBTO:RESult:FRAMe<n>:SRCaddress?....1890
BUS<m>:TBTO:RESult:FRAMe<n>:SSYMbol?....1890
BUS<m>:TBTO:RESult:FRAMe<n>:STARt?....1890
BUS<m>:TBTO:RESult:FRAMe<n>:STATe?....1891
BUS<m>:TBTO:RESult:FRAMe<n>:STOP?....1891
BUS<m>:TBTO:RESult:FRAMe<n>:TYPE?....1892
BUS<m>:TBTO:RESult:FRAMe<n>:WORD<o>:TYPE?....1892
BUS<m>:TBTO:RESult:FRAMe<n>:WORD<o>:VALue?....1892
BUS<m>:TBTO:RTYP....1886
BUS<m>:TBTO:SYMRate....1886
BUS<m>:THReshold....1638
BUS<m>:TYPE....1637
BUS<m>:TYPE....1642
BUS<m>:UART:BAUDrate....1700
BUS<m>:UART:BITRate....1700
BUS<m>:UART:EWORd....1703
BUS<m>:UART:PACKets....1702
BUS<m>:UART:PARity....1700
BUS<m>:UART:POLarity....1701
BUS<m>:UART:RX:SOURce....1698
BUS<m>:UART:RX:THReshold....1698
BUS<m>:UART:SBIT....1701
BUS<m>:UART:SSIZe....1701
BUS<m>:UART:TECHnology....1699
BUS<m>:UART:TOUT....1702
BUS<m>:UART:TX:SOURce....1698
BUS<m>:UART:TX:THReshold....1699
BUS<m>:UART:WORD<n>:BITRate?....1706

BUS<m>:UART:WORD<n>:COUNt? 1706
BUS<m>:UART:WORD<n>:RXValue? 1706
BUS<m>:UART:WORD<n>:SOURce? 1707
BUS<m>:UART:WORD<n>:STARt? 1707
BUS<m>:UART:WORD<n>:STATe? 1707
BUS<m>:UART:WORD<n>:TXValue? 1706
BUS<m>:USB:DIFFerential:SOURce 2121
BUS<m>:USB:DIFFerential:THReshold 2122
BUS<m>:USB:DMINus:SOURce 2121
BUS<m>:USB:DMINus:THReshold 2122
BUS<m>:USB:DPLus:SOURce 2121
BUS<m>:USB:DPLus:THReshold 2121
BUS<m>:USB:PACKet<n>:ADDRess? 2137
BUS<m>:USB:PACKet<n>:CRC? 2137
BUS<m>:USB:PACKet<n>:DATA? 2137
BUS<m>:USB:PACKet<n>:ENDPoint? 2138
BUS<m>:USB:PACKet<n>:ET? 2138
BUS<m>:USB:PACKet<n>:FRAMe? 2138
BUS<m>:USB:PACKet<n>:PID? 2135
BUS<m>:USB:PACKet<n>:PORT? 2138
BUS<m>:USB:PACKet<n>:SC? 2139
BUS<m>:USB:PACKet<n>:SEU? 2139
BUS<m>:USB:PACKet<n>:STARt? 2139
BUS<m>:USB:PACKet<n>:STATus? 2140
BUS<m>:USB:PACKet<n>:STOP? 2140
BUS<m>:USB:PCOunt? 2140
BUS<m>:USB:TECHnology 2120
BUS<m>:USBPd:DETail 2201
BUS<m>:USBPd:HYSTeresis 2201
BUS<m>:USBPd:RESult:FCOunt? 2209
BUS<m>:USBPd:RESult:FRAMe<n>:DATA? 2210
BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:FVAL? 2210
BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:STARt? 2211
BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:STATus? 2210
BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:STOP? 2211
BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:TYPE? 2212
BUS<m>:USBPd:RESult:FRAMe<n>:FLD<o>:VAL? 2212
BUS<m>:USBPd:RESult:FRAMe<n>:INFO? 2212
BUS<m>:USBPd:RESult:FRAMe<n>:STARt? 2213
BUS<m>:USBPd:RESult:FRAMe<n>:STATe? 2213
BUS<m>:USBPd:RESult:FRAMe<n>:STOP? 2213
BUS<m>:USBPd:RESult:FRAMe<n>:TYPE? 2214
BUS<m>:USBPd:SOURce 2201
BUS<m>:USBPd:THRBottom 2202
BUS<m>:USBPd:THReshold 2201
BUS<m>:USBPd:THRMid 2202
BUS<m>:USBPd:THRTop 2202
BUS<m>:USBThree:BITR 2166
BUS<m>:USBThree:DFE:COEFficient 2166
BUS<m>:USBThree:DFE:ENBLe 2166

BUS<m>:USBThree:DFE:ESTCoeff....2166
BUS<m>:USBThree:DFE:TCONt....2167
BUS<m>:USBThree:POLarity....2162
BUS<m>:USBThree:PROTocol....2167
BUS<m>:USBThree:RESult:FCOunt?....2186
BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:FVAL?....2189
BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:STARt?....2188
BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:STATus?....2188
BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:STOP?....2189
BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:TYPE?....2188
BUS<m>:USBThree:RESult:FRAMe<n>:FLD<o>:VAL?....2189
BUS<m>:USBThree:RESult:FRAMe<n>:INFO?....2186
BUS<m>:USBThree:RESult:FRAMe<n>:STARt?....2187
BUS<m>:USBThree:RESult:FRAMe<n>:STATe?....2187
BUS<m>:USBThree:RESult:FRAMe<n>:STOP?....2187
BUS<m>:USBThree:RESult:FRAMe<n>:TYPE?....2186
BUS<m>:USBThree:SCRMode....2163
BUS<m>:USBThree:SOURce....2162
BUS<m>:USBThree:THRHigh....2163
BUS<m>:USBThree:THRLow....2163
BUS<m>:USBThree:TYPE....2164
BUS<m>:ZCOupling....1640
BUS<m>[:STATe]....1637
CALCulate:MATH<m>:ARIThmetics....1401
CALCulate:MATH<m>:DATA:HEADer?....1404
CALCulate:MATH<m>:DATA:HEADer?....1538
CALCulate:MATH<m>:DATA:STYPe?....1404
CALCulate:MATH<m>:DATA:STYPe?....1538
CALCulate:MATH<m>:DATA[:VALues]?....1405
CALCulate:MATH<m>:DATA[:VALues]?....1539
CALCulate:MATH<m>:ENVSelection....1401
CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:ADJusted?....1527
CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:AUTO....1527
CALCulate:MATH<m>:FFT:BANDwidth[:RESolution]:RATio....1528
CALCulate:MATH<m>:FFT:BANDwidth[:RESolution][:VALue]....1528
CALCulate:MATH<m>:FFT:CFRequency....1526
CALCulate:MATH<m>:FFT:COUPled:WITH<1..8>....1537
CALCulate:MATH<m>:FFT:FRAMe:ARIThmetics....1530
CALCulate:MATH<m>:FFT:FRAMe:COVerage?....1530
CALCulate:MATH<m>:FFT:FRAMe:MAXCount....1530
CALCulate:MATH<m>:FFT:FRAMe:OFACtor....1531
CALCulate:MATH<m>:FFT:FULLspan....1526
CALCulate:MATH<m>:FFT:GATE:ABSolute:STARt....1532
CALCulate:MATH<m>:FFT:GATE:ABSolute:STOP....1532
CALCulate:MATH<m>:FFT:GATE:COUPling....1531
CALCulate:MATH<m>:FFT:GATE:MODE....1533
CALCulate:MATH<m>:FFT:GATE:RELative:STARt....1533
CALCulate:MATH<m>:FFT:GATE:RELative:STOP....1533
CALCulate:MATH<m>:FFT:GATE:ZCOupling....1534
CALCulate:MATH<m>:FFT:GATE[:STATe]....1534

CALCulate:MATH<m>:FFT:LOGScale....1525
CALCulate:MATH<m>:FFT:MAGNitude:LEVel....1534
CALCulate:MATH<m>:FFT:MAGNitude:RANGe....1535
CALCulate:MATH<m>:FFT:MAGNitude:SCALe....1535
CALCulate:MATH<m>:FFT:PHASe:SCALe....1536
CALCulate:MATH<m>:FFT:PHASe:SUPPression....1536
CALCulate:MATH<m>:FFT:PHASe:THReshold....1537
CALCulate:MATH<m>:FFT:PHASe:UNWRap....1537
CALCulate:MATH<m>:FFT:SPAN....1526
CALCulate:MATH<m>:FFT:SPECtrogram:CMODe....1540
CALCulate:MATH<m>:FFT:SPECtrogram:STATe....1540
CALCulate:MATH<m>:FFT:SPECtrogram:TIMeline<n>:POSition....1541
CALCulate:MATH<m>:FFT:SPECtrogram:TIMeline<n>:STATe....1541
CALCulate:MATH<m>:FFT:STARt....1525
CALCulate:MATH<m>:FFT:STOP....1525
CALCulate:MATH<m>:FFT:TYPE....1527
CALCulate:MATH<m>:FFT:USEColtab....1540
CALCulate:MATH<m>:FFT:WINDow:TYPE....1529
CALCulate:MATH<m>:STATe....1401
CALCulate:MATH<m>:VERTical:MAXimum....1403
CALCulate:MATH<m>:VERTical:OFFSet....1402
CALCulate:MATH<m>:VERTical:RANGe....1403
CALCulate:MATH<m>:VERTical:SCALe....1403
CALCulate:MATH<m>[:EXPRession][:DEFine]....1400
CALibration:DATE?....1232
CALibration:DATE?....2461
CALibration:RESult?....1232
CALibration:RESult?....2462
CALibration:SOURce:FREQuency....1301
CALibration:SOURce:STATe....1301
CALibration:TIME?....1232
CALibration:TIME?....2462
CDR:HARDware:BITRate....1332
CDR:HARDware:BREStimate....1332
CDR:HARDware:ESLope....1332
CDR:HARDware:SERStandard....1331
CDR:HARDware:SOURce....1331
CDR:HARDware:UIOFfset....1333
CDR:SOFTware<m>:ALGorithm....2406
CDR:SOFTware<m>:BITRate....2408
CDR:SOFTware<m>:ESBRate:BREStimate....2408
CDR:SOFTware<m>:ESBRate:CONTinuous....2408
CDR:SOFTware<m>:ESBRate:SOURce....2407
CDR:SOFTware<m>:ESLope....2409
CDR:SOFTware<m>:PLL:BWIDth....2410
CDR:SOFTware<m>:PLL:DAMPing....2411
CDR:SOFTware<m>:PLL:ORDer....2410
CDR:SOFTware<m>:PLL:RELBwidth....2411
CDR:SOFTware<m>:RESults....2410
CDR:SOFTware<m>:REVLockpll....2411

CDR:SOFTware<m>:SAMTime....2412
CDR:SOFTware<m>:SERStandard....2407
CDR:SOFTware<m>:SYNC....2409
CHANnel<m>:BANDwidth....1251
CHANnel<m>:COUPling....1248
CHANnel<m>:CPLing....1251
CHANnel<m>:DIGFilter:CUToff....1298
CHANnel<m>:DIGFilter:STATe....1298
CHANnel<m>:EATScale....1257
CHANnel<m>:EATTenuation....1258
CHANnel<m>:EXPortstate....1616
CHANnel<m>:IMPedance....1252
CHANnel<m>:INVert....1250
CHANnel<m>:IQ:CFRequency....2370
CHANnel<m>:IQ:DATA:HEADer?....2373
CHANnel<m>:IQ:DATA[:VALues]?....2372
CHANnel<m>:IQ:INPMode....2368
CHANnel<m>:IQ:INPType....2368
CHANnel<m>:IQ:SBIF....2371
CHANnel<m>:IQ:SBRF....2370
CHANnel<m>:IQ:WARNing:MESSage?....2372
CHANnel<m>:IQ:WARNing[:STATe]?....2371
CHANnel<m>:OFFSet....1250
CHANnel<m>:OVERload....1252
CHANnel<m>:POSition....1250
CHANnel<m>:RANGe....1249
CHANnel<m>:SCALe....1249
CHANnel<m>:SKEW:EXTended....1300
CHANnel<m>:SKEW:MANual....1299
CHANnel<m>:SKEW:TIME....1300
CHANnel<m>:STATe....1248
CHANnel<m>[:WAVeform<n>]:ARITHmetics....1244
CHANnel<m>[:WAVeform<n>]:DATA:HEADer?....1253
CHANnel<m>[:WAVeform<n>]:DATA[:VALues]?....1253
CHANnel<m>[:WAVeform<n>]:HISTory:CURRent....1406
CHANnel<m>[:WAVeform<n>]:HISTory:PLAY....1408
CHANnel<m>[:WAVeform<n>]:HISTory:REPLay....1408
CHANnel<m>[:WAVeform<n>]:HISTory:STARt....1406
CHANnel<m>[:WAVeform<n>]:HISTory:STOP....1407
CHANnel<m>[:WAVeform<n>]:HISTory:TPACq....1407
CHANnel<m>[:WAVeform<n>]:HISTory:TSABsolute?....1409
CHANnel<m>[:WAVeform<n>]:HISTory:TSDate?....1408
CHANnel<m>[:WAVeform<n>]:HISTory:TSRelative?....1409
CHANnel<m>[:WAVeform<n>]:HISTory:TSRReference?....1410
CHANnel<m>[:WAVeform<n>]:HISTory[:STATe]....1405
CHANnel<m>[:WAVeform<n>]:TYPE....1244
CHANnel<m>[:WAVeform<n>][:STATe]....1243
CURSor<m>:AOFF....1414
CURSor<m>:FFT:SETCenter....1424
CURSor<m>:FFT:TOCenter....1424

CURSor<m>:FUNCtion....1414
CURSor<m>:LABel:X<1..2>....1422
CURSor<m>:LABel:Y<1..2>....1422
CURSor<m>:LABel[:LABel]....1421
CURSor<m>:MAXimum:LEFT....1425
CURSor<m>:MAXimum:NEXT....1425
CURSor<m>:MAXimum:RIGHt....1425
CURSor<m>:MAXimum[:PEAK]....1425
CURSor<m>:PEXCursion....1594
CURSor<m>:SIAD....1422
CURSor<m>:SOURce....1415
CURSor<m>:SSOurce....1416
CURSor<m>:STATe....1414
CURSor<m>:STYLe....1223
CURSor<m>:THReshold....1594
CURSor<m>:TRACking[:STATe]....1418
CURSor<m>:USSource....1417
CURSor<m>:X1ENvelope....1420
CURSor<m>:X1Position....1418
CURSor<m>:X2ENvelope....1421
CURSor<m>:X2Position....1418
CURSor<m>:XCOupling....1419
CURSor<m>:XDELta:INVerse?....1423
CURSor<m>:XDELta[:VALue]?....1423
CURSor<m>:Y1Position....1419
CURSor<m>:Y2Position....1420
CURSor<m>:YCOupling....1420
CURSor<m>:YDELta:SLOPe....1424
CURSor<m>:YDELta[:VALue]?....1423
DEEMbedding<m>:ADD....2432
DEEMbedding<m>:BANDwidth....2434
DEEMbedding<m>:CADC?....2443
DEEMbedding<m>:CCOunt?....2433
DEEMbedding<m>:CMPDelay....2435
DEEMbedding<m>:COMPonent<n>:IDENtical....2441
DEEMbedding<m>:COMPonent<n>:IPONe....2442
DEEMbedding<m>:COMPonent<n>:IPTWo....2442
DEEMbedding<m>:COMPonent<n>:LOAD:IMPedance?....2439
DEEMbedding<m>:COMPonent<n>:LOAD:REMProbeload....2439
DEEMbedding<m>:COMPonent<n>:MODE....2438
DEEMbedding<m>:COMPonent<n>:NAME....2438
DEEMbedding<m>:COMPonent<n>:OPONe....2442
DEEMbedding<m>:COMPonent<n>:OPTWo....2442
DEEMbedding<m>:COMPonent<n>:PCOunt?....2438
DEEMbedding<m>:COMPonent<n>:SPONe....2442
DEEMbedding<m>:COMPonent<n>:SPTWo....2442
DEEMbedding<m>:COMPonent<n>:TYPE....2437
DEEMbedding<m>:COMPonent<n>[:STATe]....2437
DEEMbedding<m>:DELete....2436
DEEMbedding<m>:MATTenuation?....2443

DEEMbedding<m>:MGAin?....2444
DEEMbedding<m>:NAME....2435
DEEMbedding<m>:OPEN....2436
DEEMbedding<m>:PRESet....2435
DEEMbedding<m>:REALtime....2434
DEEMbedding<m>:REMove....2433
DEEMbedding<m>:SAVE....2436
DEEMbedding<m>[:STATe]....2432
DIAGnostic:SERVice:CHANnelcount?....1234
DIAGnostic:SERVice:COMPutername....1206
DIAGnostic:SERVice:FWVersion?....1233
DIAGnostic:SERVice:PARTnumber....1234
DIAGnostic:SERVice:PWD....2463
DIAGnostic:SERVice:SERialnumber?....1234
DIAGnostic:SERVice:STST:EXECute....2462
DIAGnostic:SERVice:STST:STATe?....1233
DIAGnostic:SERVice:STST:STATe?....2462
DIAGnostic:SERVice:WFAModel?....2461
DIAGnostic:SERVice:WFASeries?....2461
DIAGnostic:SERVice:WFAType?....2461
DIFFerential<m>:AOUTput....1295
DIFFerential<m>:BOUTput....1295
DIFFerential<m>:COMMon:OFFSet....1297
DIFFerential<m>:COMMon:POSition....1297
DIFFerential<m>:COMMon:SCALe....1296
DIFFerential<m>:COMMon:STATe....1296
DIFFerential<m>:COUPling....1297
DIFFerential<m>:DIFFerential:OFFSet....1297
DIFFerential<m>:DIFFerential:POSition....1297
DIFFerential<m>:DIFFerential:SCALe....1296
DIFFerential<m>:DIFFerential:STATe....1296
DIFFerential<m>:NSIGnal:STATe....1296
DIFFerential<m>:NSIGnal[:SELect]....1295
DIFFerential<m>:PSIGnal:STATe....1296
DIFFerential<m>:PSIGnal[:SELect]....1295
DIFFerential<m>:STATe....1295
DIGital<m>:DATA:HEADer?....2332
DIGital<m>:DATA[:VALues]?....2332
DIGital<m>:DESKew....2312
DIGital<m>:DISPlay....2310
DIGital<m>:HYSTeresis....2311
DIGital<m>:LABel....2312
DIGital<m>:TECHnology....2310
DIGital<m>:THCoupling....2311
DIGital<m>:THReshold....2311
DISPlay:CLR....1231
DISPlay:COLor:PALette:ADD....1208
DISPlay:COLor:PALette:COUNt?....1208
DISPlay:COLor:PALette:POINt:ADD....1208
DISPlay:COLor:PALette:POINt:COUNt?....1209

DISPlay:COLor:PALette:POINt:INSert....1208
DISPlay:COLor:PALette:POINt:REMove....1209
DISPlay:COLor:PALette:POINt[:VALue]....1209
DISPlay:COLor:PALette:REMove....1208
DISPlay:COLor:SIGNal<m>:ASSign....1207
DISPlay:COLor:SIGNal<m>:COLor....1207
DISPlay:COLor:SIGNal<m>:USE....1207
DISPlay:DIAGram:CROSshair....1210
DISPlay:DIAGram:FINegrid....1210
DISPlay:DIAGram:GRID....1210
DISPlay:DIAGram:LABels....1210
DISPlay:DIAGram:STYLe....1230
DISPlay:DIAGram:TITLe....1215
DISPlay:DIAGram:XFIXed....1211
DISPlay:DIAGram:YFIXed....1211
DISPlay:EXTended:PORDialogs....1222
DISPlay:EXTended:POSDialogs....1222
DISPlay:GATE:TRANsparency....1215
DISPlay:INTensity....1230
DISPlay:PERSistence:INFinite....1229
DISPlay:PERSistence:RESet....1230
DISPlay:PERSistence:TIME....1229
DISPlay:PERSistence[:STATe]....1229
DISPlay:RESultboxes:CUPosition....1223
DISPlay:RESultboxes:DEPosition....1640
DISPlay:RESultboxes:MEPosition....1224
DISPlay:SIGBar[:STATe]....1215
DISPlay:SIGNal:LABel:ADD....1217
DISPlay:SIGNal:LABel:FONTsize....1217
DISPlay:SIGNal:LABel:HORizontal:ABSolute:POSition....1221
DISPlay:SIGNal:LABel:HORizontal:RELative:POSition....1222
DISPlay:SIGNal:LABel:IDSList....1219
DISPlay:SIGNal:LABel:POSMode....1217
DISPlay:SIGNal:LABel:REMove....1219
DISPlay:SIGNal:LABel:SHOW....1219
DISPlay:SIGNal:LABel:STATe....1220
DISPlay:SIGNal:LABel:TEXT....1221
DISPlay:SIGNal:LABel:VERTical:ABSolute:POSition....1221
DISPlay:SIGNal:LABel:VERTical:RELative:POSition....1222
DISPlay:TRIGger:LINes....1307
EFRontend:COUNt....2446
EFRontend:CSELect<m>....2446
EFRontend:PRESet....2447
EFRontend:SETup....2447
EFRontend<m>:ALIGn:FILE....2460
EFRontend<m>:ALIGn:STATe....2460
EFRontend<m>:AMPLmode....2448
EFRontend<m>:ATTenuation:AUTO....2449
EFRontend<m>:ATTenuation:STEP:LIST?....2449
EFRontend<m>:ATTenuation[:VALue]....2449

EFRontend<m>:CONNection:CONFig ... 2445
EFRontend<m>:CONNection:CSTate? ... 2445
EFRontend<m>:CONNection:ERRor? ... 2445
EFRontend<m>:CONNection:STATe ... 2446
EFRontend<m>:FREQuency:BAND:COUNt? ... 2450
EFRontend<m>:FREQuency:BAND<n>:LIST? ... 2450
EFRontend<m>:FREQuency:BAND<n>:LOWer? ... 2450
EFRontend<m>:FREQuency:BAND<n>:UPPer? ... 2451
EFRontend<m>:FREQuency:BCONfig:AUTO ... 2451
EFRontend<m>:FREQuency:BCONfig:COUNt? ... 2451
EFRontend<m>:FREQuency:BCONfig:IFRange? ... 2451
EFRontend<m>:FREQuency:BCONfig:LIST? ... 2452
EFRontend<m>:FREQuency:BCONfig:RFRange? ... 2452
EFRontend<m>:FREQuency:BCONfig:SELect ... 2452
EFRontend<m>:FREQuency:IFBWidth:LIMit? ... 2453
EFRontend<m>:FREQuency:IFRequency:MAX? ... 2455
EFRontend<m>:FREQuency:IFRequency:MIN? ... 2456
EFRontend<m>:FREQuency:IFRequency:SIDeband? ... 2456
EFRontend<m>:FREQuency:IFRequency:VALue? ... 2453
EFRontend<m>:FREQuency:LOSCillator:FREQuency? ... 2453
EFRontend<m>:FREQuency:LOSCillator:INPut:FREQuency? ... 2454
EFRontend<m>:FREQuency:LOSCillator:MODE ... 2454
EFRontend<m>:FREQuency:LOSCillator:OUTPut:FREQuency? ... 2454
EFRontend<m>:FREQuency:LOSCillator:OUTPut:STATe ... 2455
EFRontend<m>:FREQuency:REFerence:LIST? ... 2457
EFRontend<m>:FREQuency:REFerence[:VALue] ... 2456
EFRontend<m>:FREQuency:RFRequency:MAX? ... 2457
EFRontend<m>:FREQuency:RFRequency:MIN? ... 2457
EFRontend<m>:FREQuency:RFRequency:VALue ... 2458
EFRontend<m>:GAIN? ... 2458
EFRontend<m>:IDN? ... 2446
EFRontend<m>:IFRequency:ATTenuation? ... 2458
EFRontend<m>:RFConnector ... 2459
EFRontend<m>:RFINput:LEVel ... 2459
EFRontend<m>:TYPE:LIST? ... 2447
EFRontend<m>:UPDate ... 2459
EXECutable:NAME ... 1635
EXECutable:PARameter ... 1635
EXECutable:WDIRectory ... 1636
EXPort:EYE:DATA? ... 1623
EXPort:EYE:NAME ... 1624
EXPort:EYE:SAVE ... 1624
EXPort:EYE:SELect ... 1624
EXPort:EYE:SOURce? ... 1624
EXPort:HISTogram:DATA? ... 1623
EXPort:HISTogram:INCidence ... 1622
EXPort:HISTogram:NAME ... 1622
EXPort:HISTogram:SAVE ... 1623
EXPort:HISTogram:SELect ... 1622
EXPort:MEASurement:DATA? ... 1626

EXPort:MEASurement:NAME....1625
EXPort:MEASurement:SAVE....1626
EXPort:MEASurement:SELect....1625
EXPort:MEASurement:TYPE....1625
EXPort:RESult:DECSymbol....1634
EXPort:RESult:DELimiter....1635
EXPort:WAVeform:CURSorset....1618
EXPort:WAVeform:DISPlayoff....1621
EXPort:WAVeform:DLOGging....1619
EXPort:WAVeform:FASTexport....1621
EXPort:WAVeform:ICDR....1622
EXPort:WAVeform:INCXvalues....1620
EXPort:WAVeform:MEAS....1619
EXPort:WAVeform:MULTichannel....1615
EXPort:WAVeform:NAME....1616
EXPort:WAVeform:RAW....1621
EXPort:WAVeform:SAVE....1617
EXPort:WAVeform:SCOPe....1617
EXPort:WAVeform:SOURce....1614
EXPort:WAVeform:STARt....1617
EXPort:WAVeform:STOP....1618
EXPort:WAVeform:TIMestamps....1620
EXPort:WAVeform:ZOOM....1618
EYE<m>:DISPlay:MODE....2419
EYE<m>:DISPlay:PERSistence:INFinite....2420
EYE<m>:DISPlay:PERSistence:RESet....2420
EYE<m>:DISPlay:PERSistence:STATe....2420
EYE<m>:DISPlay:PERSistence:TIME....2420
EYE<m>:FILTer:BPATtern:CENTer....2426
EYE<m>:FILTer:BPATtern:MODE....2426
EYE<m>:FILTer:BPATtern:PREFix<1..8>....2426
EYE<m>:FILTer:BPATtern:RESet....2427
EYE<m>:FILTer:BPATtern:SLEFt....2426
EYE<m>:FILTer:BPATtern:SRIGht....2427
EYE<m>:FILTer:BPATtern:SUFFix<1..5>....2426
EYE<m>:FILTer:DDR:FTYPe....2427
EYE<m>:FILTer:DDR:LENerror....2427
EYE<m>:FILTer:STATe....2425
EYE<m>:HORizontal:ABSolute:POSition....2414
EYE<m>:HORizontal:ABSolute:RANGe....2414
EYE<m>:HORizontal:UINTerval:POSition....2415
EYE<m>:HORizontal:UINTerval:RANGe....2415
EYE<m>:HORizontal[:UNIT]....2414
EYE<m>:MEASurement:MAXPrecision....2419
EYE<m>:MSLices....2419
EYE<m>:QUALify:GATE:ABSolute:STARt....2422
EYE<m>:QUALify:GATE:ABSolute:STOP....2422
EYE<m>:QUALify:GATE:COUPling....2421
EYE<m>:QUALify:GATE:CURSor....2423
EYE<m>:QUALify:GATE:MODE....2422

EYE<m>:QUALify:GATE:RELative:STARt....2422
EYE<m>:QUALify:GATE:RELative:STOP....2422
EYE<m>:QUALify:GATE:ZDIagram....2423
EYE<m>:QUALify:GATE[:STATe]....2421
EYE<m>:QUALify:SIGNal:CONDition....2424
EYE<m>:QUALify:SIGNal:LSELect....2425
EYE<m>:QUALify:SIGNal:SOURce....2424
EYE<m>:QUALify:SIGNal[:STATe]....2424
EYE<m>:RESult:SLICe<n>:COUNt?....2428
EYE<m>:RESult:SLICe<n>:LPOSition?....2428
EYE<m>:RESult:SLICe<n>[:POSition]?....2428
EYE<m>:RESult:TACQs?....2429
EYE<m>:RESult:TSLices?....2429
EYE<m>:SOURce....2413
EYE<m>:TIMReference:CLOCk:EDGE....2415
EYE<m>:TIMReference:CLOCk:FACTor....2415
EYE<m>:TIMReference:CLOCk:FREQuency....2416
EYE<m>:TIMReference:CLOCk:FREStimation....2416
EYE<m>:TIMReference:CLOCk:LEVel....2416
EYE<m>:TIMReference:CLOCk:OFEStimation....2417
EYE<m>:TIMReference:CLOCk:OFFSet....2417
EYE<m>:TIMReference:CLOCk:SAMTime....2417
EYE<m>:TIMReference:CLOCk:SOURce....2418
EYE<m>:TIMReference:PROTocol:SELect....2418
EYE<m>:TIMReference[:SOURce]....2418
EYE<m>:ZCOupling....2431
EYE<m>:ZDIagram....2431
EYE<m>[:STATe]....2413
FORMat:BORDer....1203
FORMat:BORDer....1303
FORMat:BPATtern....1203
FORMat[:DATA]....1201
GENerator:ALIGnment:DC:ABORt....2354
GENerator:ALIGnment:DC:RESult:DATE?....2354
GENerator:ALIGnment:DC:RESult:TIME?....2354
GENerator:ALIGnment:DC:RESult[:STATe]?....2354
GENerator:ALIGnment:DC[:STARt]....2354
GENerator:SYNC[:COMBination]....2356
GPIB:ADDRess....1204
GPIB:TERMinator....1204
HCOPy:CMAP<m>:DEFault....1629
HCOPy:DESTination<1..2>....1627
HCOPy:DEVice<m>:INVerse....1628
HCOPy:DEVice<m>:LANGuage....1628
HCOPy:IMMediate<m>:NEXT....1630
HCOPy:IMMediate<m>[:DUM]....1630
HCOPy:ISBA....1630
HCOPy:SSD....1629
HCOPy:WBKG....1629
HDEFinition:BWIDth....1302

HDEFinition:RESolution?....1302
HDEFinition:STATe....1302
IQ:BWIDth?....2369
IQ:INSRate....2368
IQ:RBWidth....2369
IQ:RLENgth....2370
IQ:SRATe....2369
IQ:STATe....2367
LAYout:ADD....1211
LAYout:HISTogram:ADD....1476
LAYout:HISTogram:ENABle....1477
LAYout:HISTogram:HORZ:ABSolute:STARt....1479
LAYout:HISTogram:HORZ:ABSolute:STOP....1479
LAYout:HISTogram:HORZ:MODE....1479
LAYout:HISTogram:HORZ:RELative:STARt....1480
LAYout:HISTogram:HORZ:RELative:STOP....1480
LAYout:HISTogram:MODE....1478
LAYout:HISTogram:REMove....1482
LAYout:HISTogram:RESet....1482
LAYout:HISTogram:SOURce....1478
LAYout:HISTogram:VERTical:ABSolute:STARt....1481
LAYout:HISTogram:VERTical:ABSolute:STOP....1481
LAYout:HISTogram:VERTical:MODE....1480
LAYout:HISTogram:VERTical:RELative:STARt....1481
LAYout:HISTogram:VERTical:RELative:STOP....1482
LAYout:REMove....1213
LAYout:SHOW....1213
LAYout:SIGNal:ASSign....1213
LAYout:SIGNal:AXIS....1411
LAYout:SIGNal:UNASsign....1215
LAYout:ZOOM:ADD....1379
LAYout:ZOOM:ADDCoupled....1379
LAYout:ZOOM:HORZ:ABSolute:POSition....1381
LAYout:ZOOM:HORZ:ABSolute:SPAN....1381
LAYout:ZOOM:HORZ:ABSolute:STARt....1381
LAYout:ZOOM:HORZ:ABSolute:STOP....1382
LAYout:ZOOM:HORZ:MODE....1380
LAYout:ZOOM:HORZ:RELative:POSition....1382
LAYout:ZOOM:HORZ:RELative:SPAN....1383
LAYout:ZOOM:HORZ:RELative:STARt....1383
LAYout:ZOOM:HORZ:RELative:STOP....1383
LAYout:ZOOM:ONEDiagram....1216
LAYout:ZOOM:POSCoupling....1380
LAYout:ZOOM:REMove....1387
LAYout:ZOOM:VERTical:ABSolute:POSition....1384
LAYout:ZOOM:VERTical:ABSolute:SPAN....1385
LAYout:ZOOM:VERTical:ABSolute:STARt....1385
LAYout:ZOOM:VERTical:ABSolute:STOP....1385
LAYout:ZOOM:VERTical:MODE....1384
LAYout:ZOOM:VERTical:RELative:POSition....1386

LAYout:ZOOM:VERTical:RELative:SPAN ..... 1386
LAYout:ZOOM:VERTical:RELative:STARt ..... 1387
LAYout:ZOOM:VERTical:RELative:STOP ..... 1387
MEASurement<m>:ADDitional ..... 1431
MEASurement<m>:AMPTime:ALEVel ..... 1440
MEASurement<m>:AMPTime:ARIThmetic:CONStant ..... 1448
MEASurement<m>:AMPTime:ARIThmetic:COUNit ..... 1448
MEASurement<m>:AMPTime:ARIThmetic:CUNState ..... 1449
MEASurement<m>:AMPTime:ARIThmetic:CUUNit ..... 1449
MEASurement<m>:AMPTime:ARIThmetic:CUUState ..... 1450
MEASurement<m>:AMPTime:ARIThmetic:OPERator ..... 1450
MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:AMPTime ..... 1450
MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:GROup ..... 1451
MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:SPECtrum ..... 1451
MEASurement<m>:AMPTime:ARIThmetic:PARameter<n>:TYPE ..... 1452
MEASurement<m>:AMPTime:CLCK<n>:LSELect ..... 1444
MEASurement<m>:AMPTime:CSLope ..... 1444
MEASurement<m>:AMPTime:DATA<n>:LSELect ..... 1445
MEASurement<m>:AMPTime:DELay<n>:DIRection ..... 1441
MEASurement<m>:AMPTime:DELay<n>:ECOunt ..... 1442
MEASurement<m>:AMPTime:DELay<n>:LSELect ..... 1442
MEASurement<m>:AMPTime:DELay<n>:SLOPe ..... 1443
MEASurement<m>:AMPTime:DTOTrigger<n>:LSELect ..... 1445
MEASurement<m>:AMPTime:DTOTrigger<n>:SLOPe ..... 1445
MEASurement<m>:AMPTime:ESLope ..... 1444
MEASurement<m>:AMPTime:LCHeck<n>:LOWer:LIMit ..... 1446
MEASurement<m>:AMPTime:LCHeck<n>:LOWer:MARGin ..... 1447
MEASurement<m>:AMPTime:LCHeck<n>:UPPer:LIMit ..... 1446
MEASurement<m>:AMPTime:LCHeck<n>:UPPer:MARGin ..... 1447
MEASurement<m>:AMPTime:LCHeck<n>:VALid ..... 1446
MEASurement<m>:AMPTime:PFSLope ..... 1440
MEASurement<m>:AMPTime:PSLope ..... 1441
MEASurement<m>:AMPTime:PTCount ..... 1443
MEASurement<m>:AOFF ..... 1432
MEASurement<m>:AON ..... 1432
MEASurement<m>:ARES? ..... 1433
MEASurement<m>:ARNames ..... 1434
MEASurement<m>:CATegory ..... 1429
MEASurement<m>:CLEar ..... 1489
MEASurement<m>:DETThreshold ..... 1440
MEASurement<m>:DISPlay:GROuping ..... 1224
MEASurement<m>:DISPlay:HISTogram ..... 1488
MEASurement<m>:DISPlay:LEVels ..... 1487
MEASurement<m>:DISPlay:RESults ..... 1487
MEASurement<m>:DISPlay:STYLe ..... 1488
MEASurement<m>:ENVSelect ..... 1439
MEASurement<m>:EYEJitter:AUToscale ..... 1458
MEASurement<m>:EYEJitter:DCDistortion:RELative:LEVel ..... 1461
MEASurement<m>:EYEJitter:EYEHeight:ALGorithm ..... 1463
MEASurement<m>:EYEJitter:EYEHeight:RELative:STARt ..... 1463

MEASurement<m>:EYEJitter:EYEHeight:RELative:STOP ..... 1464
MEASurement<m>:EYEJitter:EYEWidth:ALGorithm ..... 1461
MEASurement<m>:EYEJitter:EYEWidth:MEASwindow ..... 1462
MEASurement<m>:EYEJitter:EYEWidth:RELative:STARt ..... 1462
MEASurement<m>:EYEJitter:EYEWidth:RELative:STOP ..... 1463
MEASurement<m>:EYEJitter:GENSettings:MLEHistogram:ALGorithm ..... 1460
MEASurement<m>:EYEJitter:GENSettings:MLEHistogram:THReshold ..... 1460
MEASurement<m>:EYEJitter:GENSettings:RELative:STARt ..... 1460
MEASurement<m>:EYEJitter:GENSettings:RELative:STOP ..... 1461
MEASurement<m>:EYEJitter:JITTer:MEASlevel ..... 1464
MEASurement<m>:EYEJitter:JITTer:RELative:LEVel ..... 1465
MEASurement<m>:EYEJitter:LCHeck<n>:LOWer:LIMit ..... 1459
MEASurement<m>:EYEJitter:LCHeck<n>:LOWer:MARGin ..... 1459
MEASurement<m>:EYEJitter:LCHeck<n>:UPPer:LIMit ..... 1459
MEASurement<m>:EYEJitter:LCHeck<n>:UPPer:MARGin ..... 1459
MEASurement<m>:EYEJitter:LCHeck<n>:VALid ..... 1458
MEASurement<m>:EYEJitter:RTFTime:RELative:STARt ..... 1465
MEASurement<m>:EYEJitter:RTFTime:RELative:STOP ..... 1465
MEASurement<m>:FSRC ..... 1429
MEASurement<m>:GATE:ABSolute:STARt ..... 1504
MEASurement<m>:GATE:ABSolute:STOP ..... 1504
MEASurement<m>:GATE:CCOupling ..... 1505
MEASurement<m>:GATE:CURSor ..... 1504
MEASurement<m>:GATE:GCOupling ..... 1506
MEASurement<m>:GATE:MODE ..... 1503
MEASurement<m>:GATE:RELative:STARt ..... 1504
MEASurement<m>:GATE:RELative:STOP ..... 1504
MEASurement<m>:GATE:ZCOupling ..... 1505
MEASurement<m>:GATE:ZDIagram ..... 1506
MEASurement<m>:GATE[:STATe] ..... 1503
MEASurement<m>:HISTogram:LCHeck<n>:LOWer:LIMit ..... 1486
MEASurement<m>:HISTogram:LCHeck<n>:LOWer:MARGin ..... 1486
MEASurement<m>:HISTogram:LCHeck<n>:UPPer:LIMit ..... 1486
MEASurement<m>:HISTogram:LCHeck<n>:UPPer:MARGin ..... 1486
MEASurement<m>:HISTogram:LCHeck<n>:VALid ..... 1485
MEASurement<m>:HISTogram:PROBability:LIMit ..... 1485
MEASurement<m>:HISTogram:PROBability:TYPE ..... 1484
MEASurement<m>:HISTogram:SELect ..... 1484
MEASurement<m>:JITTer:CCSLope ..... 2374
MEASurement<m>:JITTer:CDRMode ..... 2376
MEASurement<m>:JITTer:DRUNit ..... 2377
MEASurement<m>:JITTer:NCYCles ..... 2376
MEASurement<m>:JITTer:PULSe ..... 2375
MEASurement<m>:JITTer:SKWRelation ..... 2378
MEASurement<m>:JITTer:SKWSlope ..... 2378
MEASurement<m>:JITTer:SOURce<n>:TIESlope ..... 2377
MEASurement<m>:LCHeck ..... 1506
MEASurement<m>:LTMeas:COUNt ..... 1494
MEASurement<m>:LTMeas:TIME ..... 1495
MEASurement<m>:LTMeas[:STATe] ..... 1494

MEASurement<m>:MAIN ... 1430
MEASurement<m>:MNOMeas ... 1490
MEASurement<m>:MULTiple ... 1489
MEASurement<m>:ONViolation:ACQStop ... 1507
MEASurement<m>:ONViolation:BEEP ... 1507
MEASurement<m>:ONViolation:REPort ... 1508
MEASurement<m>:ONViolation:RUNexec ... 1509
MEASurement<m>:ONViolation:TRIGgerout ... 1508
MEASurement<m>:ONViolation:WFMSave ... 1508
MEASurement<m>:OPTical:LCHeck<n>:LOWer:LIMit ... 1475
MEASurement<m>:OPTical:LCHeck<n>:LOWer:MARGin ... 1475
MEASurement<m>:OPTical:LCHeck<n>:UPPer:LIMit ... 1475
MEASurement<m>:OPTical:LCHeck<n>:UPPer:MARGin ... 1475
MEASurement<m>:OPTical:LCHeck<n>:VALid ... 1475
MEASurement<m>:OPTical:OMAMplitude:RELative:STARt ... 1474
MEASurement<m>:OPTical:OMAMplitude:RELative:STOP ... 1474
MEASurement<m>:OPTical:UNIT ... 1474
MEASurement<m>:PROTocol:F2FRame:FLDFrom ... 1496
MEASurement<m>:PROTocol:F2FRame:FLDTo ... 1496
MEASurement<m>:PROTocol:F2FRame:FRMFrom ... 1497
MEASurement<m>:PROTocol:F2FRame:FRMTo ... 1497
MEASurement<m>:PROTocol:F2FRame:VALFrom ... 1497
MEASurement<m>:PROTocol:F2FRame:VALTo ... 1497
MEASurement<m>:PROTocol:FLDValue:FLD ... 1498
MEASurement<m>:PROTocol:FLDValue:FRM ... 1498
MEASurement<m>:PROTocol:FLDValue:TRCK ... 1498
MEASurement<m>:PROTocol:FLDValue:VAL ... 1499
MEASurement<m>:PROTocol:MBITrate:FLD ... 1499
MEASurement<m>:PROTocol:MBITrate:FRM ... 1499
MEASurement<m>:PROTocol:MBITrate:VAL ... 1499
MEASurement<m>:PROTocol:SBITrate:FLD ... 1500
MEASurement<m>:PROTocol:SBITrate:FRM ... 1500
MEASurement<m>:PROTocol:SBITrate:VAL ... 1500
MEASurement<m>:PROTocol:T2FRame:DIRection ... 1500
MEASurement<m>:PROTocol:T2FRame:FLD ... 1501
MEASurement<m>:PROTocol:T2FRame:FRM ... 1501
MEASurement<m>:PROTocol:T2FRame:VALue ... 1501
MEASurement<m>:REFLevel:RESult:BINNer? ... 1522
MEASurement<m>:REFLevel:RESult:BOUTer? ... 1522
MEASurement<m>:REFLevel:RESult:LOWer? ... 1521
MEASurement<m>:REFLevel:RESult:MIDDle? ... 1521
MEASurement<m>:REFLevel:RESult:SIGHigh? ... 1522
MEASurement<m>:REFLevel:RESult:SIGLow? ... 1522
MEASurement<m>:REFLevel:RESult:TINNer? ... 1523
MEASurement<m>:REFLevel:RESult:TOUTer? ... 1523
MEASurement<m>:REFLevel:RESult:UPPer? ... 1521
MEASurement<m>:RESult:AVG? ... 1434
MEASurement<m>:RESult:COUNt? ... 1436
MEASurement<m>:RESult:EVTCount? ... 1434
MEASurement<m>:RESult:INVerse ... 1225

MEASurement<m>:RESult:LABorder....1224
MEASurement<m>:RESult:MAXCount....1470
MEASurement<m>:RESult:NPEak?....1434
MEASurement<m>:RESult:PPEak?....1434
MEASurement<m>:RESult:RELiability?....1434
MEASurement<m>:RESult:RMS?....1434
MEASurement<m>:RESult:SHFRequency....1470
MEASurement<m>:RESult:SHLabels....1471
MEASurement<m>:RESult:STARt?....1436
MEASurement<m>:RESult:STDDev?....1434
MEASurement<m>:RESult:STOP?....1436
MEASurement<m>:RESult:WFMCount?....1434
MEASurement<m>:RESult[:ACTual]?....1434
MEASurement<m>:SOURce....1427
MEASurement<m>:SPECtrum:ARIThmetic:CONStant....1452
MEASurement<m>:SPECtrum:ARIThmetic:COUNit....1453
MEASurement<m>:SPECtrum:ARIThmetic:CUNState....1453
MEASurement<m>:SPECtrum:ARIThmetic:CUUNit....1454
MEASurement<m>:SPECtrum:ARIThmetic:CUUState....1454
MEASurement<m>:SPECtrum:ARIThmetic:OPERator....1454
MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:AMPTime....1454
MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:GROup....1455
MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:SPECtrum....1455
MEASurement<m>:SPECtrum:ARIThmetic:PARameter<n>:TYPE....1456
MEASurement<m>:SPECtrum:ATHReshold....1469
MEASurement<m>:SPECtrum:CPOWer:BANDwidth....1467
MEASurement<m>:SPECtrum:CPOWer:CFRequency....1468
MEASurement<m>:SPECtrum:LCHeck<n>:LOWer:LIMit....1472
MEASurement<m>:SPECtrum:LCHeck<n>:LOWer:MARGin....1473
MEASurement<m>:SPECtrum:LCHeck<n>:UPPer:LIMit....1472
MEASurement<m>:SPECtrum:LCHeck<n>:UPPer:MARGin....1473
MEASurement<m>:SPECtrum:LCHeck<n>:VALid....1471
MEASurement<m>:SPECtrum:NDBDown....1468
MEASurement<m>:SPECtrum:OBANdwidth....1467
MEASurement<m>:SPECtrum:PEXCursion....1468
MEASurement<m>:SPECtrum:RESult<n>:COUNt....1469
MEASurement<m>:SPECtrum:RESult<n>:MODE....1469
MEASurement<m>:SSRC....1429
MEASurement<m>:STATistics:HBINs....1490
MEASurement<m>:STATistics:HISTogram....1490
MEASurement<m>:STATistics:MODE....1491
MEASurement<m>:STATistics:RCOunt....1491
MEASurement<m>:STATistics:RESet....1492
MEASurement<m>:STATistics:RMEascount....1492
MEASurement<m>:STATistics:RTIMe....1492
MEASurement<m>:STATistics[:ENABle]....1489
MEASurement<m>:TRACk:DATA:HEADer?....1502
MEASurement<m>:TRACk:DATA:STYPe?....1502
MEASurement<m>:TRACk:DATA[:VALues]?....1502
MEASurement<m>:TRACk[:STATe]....1502

MEASurement<m>:VERTical:AUTO....1493
MEASurement<m>:VERTical:CONT....1493
MEASurement<m>:VERTical:OFFSet....1493
MEASurement<m>:VERTical:SCALe....1494
MEASurement<m>[:ENABle]....1427
MMEMory:ATTRibute....1610
MMEMory:AUTonaming:DATE....1633
MMEMory:AUTonaming:DEFaultpath....1634
MMEMory:AUTonaming:INDex....1633
MMEMory:AUTonaming:PREFix....1633
MMEMory:AUTonaming:RESall....1634
MMEMory:AUTonaming:RESPath....1634
MMEMory:AUTonaming:TEXT....1633
MMEMory:AUTonaming:TIME....1633
MMEMory:AUTonaming:USERtext....1633
MMEMory:CATalog:LENGth?....1608
MMEMory:CATalog?....1607
MMEMory:CDIRectory....1606
MMEMory:COPY....1608
MMEMory:DATA....1609
MMEMory:DCATalog:LENGth?....1606
MMEMory:DCATalog?....1606
MMEMory:DELete....1609
MMEMory:DRIVes?....1605
MMEMory:LOAD:STATe....1612
MMEMory:MDIRectory....1607
MMEMory:MOVE....1609
MMEMory:MSIS....1605
MMEMory:NAME....1627
MMEMory:RCL....1611
MMEMory:RDIRectory....1607
MMEMory:SAV....1610
MMEMory:STORe:STATe....1611
MTESt:ADD....1542
MTESt:CEMask....1561
MTESt:COLor:BORDer....1228
MTESt:COLor:INTerior....1228
MTESt:COLor:MATCh....1227
MTESt:COLor:UNMatch....1227
MTESt:CONDition....1544
MTESt:CTYPe....1545
MTESt:EYEMask:BOFFset....1563
MTESt:EYEMask:HEIGht<m>[:VALue]....1561
MTESt:EYEMask:HPERiod....1562
MTESt:EYEMask:HPOSition....1564
MTESt:EYEMask:MSKBottom....1563
MTESt:EYEMask:MSKLeft....1562
MTESt:EYEMask:MSKRight....1562
MTESt:EYEMask:MSKTop....1563
MTESt:EYEMask:TBSYmmetric....1564

MTESt:EYEMask:TBWidth ... 1564
MTESt:EYEMask:TOFFset ... 1563
MTESt:EYEMask:TYPE ... 1561
MTESt:EYEMask:VPOSition ... 1565
MTESt:EYEMask:WIDTh<m>[:VALue] ... 1562
MTESt:FILE:DELete ... 1546
MTESt:FILE:NAME ... 1545
MTESt:FILE:OPEN ... 1546
MTESt:FILE:SAVE ... 1546
MTESt:HIGHlight:INFinite ... 1226
MTESt:HIGHlight:STATe ... 1227
MTESt:HIGHlight:TIME ... 1227
MTESt:LABel ... 1226
MTESt:ONViolation:BEEP ... 1555
MTESt:ONViolation:PRINt ... 1556
MTESt:ONViolation:REPort ... 1557
MTESt:ONViolation:RUNexec ... 1557
MTESt:ONViolation:SAVewaveform ... 1556
MTESt:ONViolation:STOP ... 1556
MTESt:ONViolation:TRIGgerout ... 1557
MTESt:REFWfm ... 1553
MTESt:REMove ... 1543
MTESt:REName ... 1226
MTESt:RESult:COUNt:FAILures? ... 1560
MTESt:RESult:COUNt:FSLices? ... 2430
MTESt:RESult:COUNt:FWAVeforms? ... 1559
MTESt:RESult:COUNt:REMaining? ... 1559
MTESt:RESult:COUNt:SLICes? ... 2429
MTESt:RESult:COUNt:WAVeforms? ... 1559
MTESt:RESult:FRATe? ... 1560
MTESt:RESult:STATe? ... 1558
MTESt:RESult:VIOLation<m>:COUNt? ... 2430
MTESt:RESult:VIOLation<m>:STARt? ... 2431
MTESt:RESult:VIOLation<m>:STOP? ... 2431
MTESt:RESult:VIOLation<m>[:SLICe]? ... 2430
MTESt:RESult[:RESult]? ... 1558
MTESt:RST ... 1543
MTESt:SBITnumber? ... 2469
MTESt:SEGMent:ADD ... 1547
MTESt:SEGMent:CLEar ... 1548
MTESt:SEGMent:COUNt? ... 1548
MTESt:SEGMent:INSert ... 1548
MTESt:SEGMent:POINt:ADD ... 1549
MTESt:SEGMent:POINt:COUNt? ... 1550
MTESt:SEGMent:POINt:INSert ... 1549
MTESt:SEGMent:POINt:REMove ... 1550
MTESt:SEGMent:POINt:X ... 1550
MTESt:SEGMent:POINt:Y ... 1551
MTESt:SEGMent:REGion ... 1549
MTESt:SEGMent:REMove ... 1548

MTESt:SEGMent:RESCale:RECalculate....1551
MTESt:SEGMent:RESCale:XFACtor....1552
MTESt:SEGMent:RESCale:XOFFset....1552
MTESt:SEGMent:RESCale:YFACtor....1552
MTESt:SEGMent:RESCale:YOFFset....1553
MTESt:SEGMent:STATe....1547
MTESt:SHOW....1543
MTESt:SOURce....1544
MTESt:TOLerance....1545
MTESt:WFMLupdate....1554
MTESt:WFMRescale:XWIDth....1554
MTESt:WFMRescale:YPOSition....1555
MTESt:WFMRescale:YSTRetch....1555
MTESt:WFMRescale:YWIDth....1554
MTESt[:STATe]....1543
PGENerator:BITRate....2349
PGENerator:ENABle....2349
PGENerator:FILE:OPEN....2349
PGENerator:FILE[:NAME]....2349
PGENerator:HLEVel....2349
PGENerator:PRESet....2350
PGENerator:RUNMode....2350
PGENerator:RUNSingle....2350
PROBe<m>:CONVerter:BANDwidth....1279
PROBe<m>:CONVerter:SELFalign....1279
PROBe<m>:CONVerter:TYPE?....1279
PROBe<m>:DEEMbedding:BANDwidth....1272
PROBe<m>:DEEMbedding:BANDwidth....2441
PROBe<m>:DEEMbedding:LOAD:IMPedance....1274
PROBe<m>:DEEMbedding:MODE....1273
PROBe<m>:DEEMbedding:PMODe....1271
PROBe<m>:DEEMbedding:REMProbeload....1273
PROBe<m>:DEEMbedding:TIPModule:CSPFile....1272
PROBe<m>:DEEMbedding:TIPModule:ZMA<n>:SUBModule....1272
PROBe<m>:DEEMbedding:TIPModule:ZMA<n>:SUBModule....2440
PROBe<m>:DEEMbedding:TIPModule[:SELect]....1271
PROBe<m>:DEEMbedding:TIPModule[:SELect]....2440
PROBe<m>:ID:PARTnumber?....1277
PROBe<m>:ID:PRDate?....1277
PROBe<m>:ID:SRNumber?....1278
PROBe<m>:ID:SWVersion?....1277
PROBe<m>:PMETer:RESults:COMMon?....1261
PROBe<m>:PMETer:RESults:DIFFerential?....1261
PROBe<m>:PMETer:RESults:NEGative?....1262
PROBe<m>:PMETer:RESults:POSitive?....1262
PROBe<m>:PMETer:RESults:SINGle?....1260
PROBe<m>:PMETer:VISibility....1260
PROBe<m>:SETup:ACCoupling....1265
PROBe<m>:SETup:ADVanced:AUDioverload....1267
PROBe<m>:SETup:ADVanced:FILTer....1266

PROBe<m>:SETup:ADVanced:PMToffset....1265
PROBe<m>:SETup:ADVanced:RANGe....1266
PROBe<m>:SETup:ATTenuation:DEFProbe....1274
PROBe<m>:SETup:ATTenuation:MANual....1263
PROBe<m>:SETup:ATTenuation:MODE....1263
PROBe<m>:SETup:ATTenuation:UNIT....1263
PROBe<m>:SETup:ATTenuation[:AUTO]?....1256
PROBe<m>:SETup:BANDwidth?....1256
PROBe<m>:SETup:CAPacitance?....1278
PROBe<m>:SETup:CMOFfset....1264
PROBe<m>:SETup:CMOFfset....1268
PROBe<m>:SETup:DEGauss....1275
PROBe<m>:SETup:DISPlaydiff....1260
PROBe<m>:SETup:DMOFfset....1268
PROBe<m>:SETup:GAIN:MANual....1264
PROBe<m>:SETup:IMPedance?....1278
PROBe<m>:SETup:MODE....1258
PROBe<m>:SETup:NAME?....1256
PROBe<m>:SETup:NOFFset....1269
PROBe<m>:SETup:OFFSet:AZERo....1257
PROBe<m>:SETup:OFFSet:STPRobe....1276
PROBe<m>:SETup:OFFSet:TOMean....1275
PROBe<m>:SETup:OFFSet:USEautozero....1257
PROBe<m>:SETup:OFFSet:ZADJust....1276
PROBe<m>:SETup:POFFset....1269
PROBe<m>:SETup:PRMode....1267
PROBe<m>:SETup:STATe?....1255
PROBe<m>:SETup:TERM:ADJust....1270
PROBe<m>:SETup:TERM:MEASure?....1270
PROBe<m>:SETup:TERM:MODE....1270
PROBe<m>:SETup:TERM:STATe....1269
PROBe<m>:SETup:TYPE?....1255
PROBe<m>:SETup:ZAXV....1265
PROBe<m>:SKEState....1300
PSRC:CLOCkmode....2358
PSRC:DUTYcycle....2358
PSRC:OUTPutlow....2358
PSRC:REPRate....2358
PSRC:RST....2359
PSRC[:STATe]....2357
REFCurve<m>:CDR:ALGorithm?....1398
REFCurve<m>:CDR:BITRate?....1398
REFCurve<m>:CDR:BWIDth?....1399
REFCurve<m>:CDR:RELBwidth?....1399
REFCurve<m>:CDR:SAMTime?....1399
REFCurve<m>:CDR:SOURce?....1400
REFCurve<m>:CLEar....1390
REFCurve<m>:DATA:HEADer?....1395
REFCurve<m>:DATA:STYPe?....1395
REFCurve<m>:DATA[:VALues]?....1396

REFCurve<m>:DELete....1390
REFCurve<m>:HMODe....1393
REFCurve<m>:MULTichannel:IMPort....1397
REFCurve<m>:MULTichannel:NAME....1397
REFCurve<m>:MULTichannel:OPEN....1397
REFCurve<m>:NAME....1389
REFCurve<m>:OPEN....1389
REFCurve<m>:POSition....1392
REFCurve<m>:RESCale:HORizontal:FACTor....1394
REFCurve<m>:RESCale:HORizontal:OFFSet....1394
REFCurve<m>:RESCale:HORizontal:POSition....1395
REFCurve<m>:RESCale:HORizontal:STATe....1394
REFCurve<m>:RESCale:VERTical:FACTor....1392
REFCurve<m>:RESCale:VERTical:OFFSet....1393
REFCurve<m>:RESCale:VERTical:STATe....1392
REFCurve<m>:RESTore....1391
REFCurve<m>:SAVE....1389
REFCurve<m>:SCALe....1391
REFCurve<m>:SOURce....1388
REFCurve<m>:STATe....1388
REFCurve<m>:TOORignal....1392
REFCurve<m>:UPDate....1389
REFCurve<m>:VMODe....1391
REFLevel<m>:ABSolute:BDIStance....1515
REFLevel<m>:ABSolute:BITube....1521
REFLevel<m>:ABSolute:BOTube....1521
REFLevel<m>:ABSolute:HIGH....1514
REFLevel<m>:ABSolute:LLEVel....1517
REFLevel<m>:ABSolute:LOW....1514
REFLevel<m>:ABSolute:MLEVel....1516
REFLevel<m>:ABSolute:TDIStance....1515
REFLevel<m>:ABSolute:TITube....1520
REFLevel<m>:ABSolute:TOTube....1520
REFLevel<m>:ABSolute:ULEVel....1516
REFLevel<m>:AUTO:COUNt....1513
REFLevel<m>:AUTO:MODE....1512
REFLevel<m>:AUTO[:STATe]....1513
REFLevel<m>:LDETection....1509
REFLevel<m>:LMODe....1510
REFLevel<m>:RELative:HYSTeresis....1519
REFLevel<m>:RELative:ITUBe....1520
REFLevel<m>:RELative:LOWer....1518
REFLevel<m>:RELative:MIDDle....1518
REFLevel<m>:RELative:MODE....1510
REFLevel<m>:RELative:OTUBe....1519
REFLevel<m>:RELative:UPPer....1517
REFLevel<m>:USRLevel....1511
REPort:COMMent....1632
REPort:FILE:NAME....1632
REPort:FILE:SAVE....1633

REPort:LANGuage....1631
REPort:LOGO....1632
REPort:LOGType....1631
REPort:PAPersize....1631
REPort:USER....1632
RUN....1235
RUNContinous....1235
RUNSingle....1235
SAVeset:CONFig:PREView....1612
SAVeset:ONEFile:NAME....1612
SAVeset:ONEFile:OPEN....1613
SAVeset:ONEFile:SAVE....1613
SAVeset:ONEFile:USERpref....1613
SEARch:ADD....1566
SEARch:ALL....1567
SEARch:CLEar....1566
SEARch:GATE:ABSolute:STARt....1596
SEARch:GATE:ABSolute:STOP....1596
SEARch:GATE:MODE....1595
SEARch:GATE:RELative:STARt....1596
SEARch:GATE:RELative:STOP....1597
SEARch:GATE:SHOW....1595
SEARch:GATE:ZCOupling....1597
SEARch:GATE:ZDIagram....1597
SEARch:GATE[:STATe]....1595
SEARch:ONLine....1567
SEARch:REMove....1566
SEARch:RESDiagram:HORZ:ABSolute:POSition....1598
SEARch:RESDiagram:HORZ:ABSolute:SPAN....1599
SEARch:RESDiagram:HORZ:MODE....1599
SEARch:RESDiagram:HORZ:RELative:POSition....1599
SEARch:RESDiagram:HORZ:RELative:SPAN....1600
SEARch:RESDiagram:SHOW....1600
SEARch:RESDiagram:VERT:ABSolute:POSition....1600
SEARch:RESDiagram:VERT:ABSolute:SPAN....1601
SEARch:RESDiagram:VERT:MODE....1601
SEARch:RESDiagram:VERT:RELative:POSition....1601
SEARch:RESDiagram:VERT:RELative:SPAN....1602
SEARch:RESult:ARINc:WCOunt?....1826
SEARch:RESult:ARINc:WORD<m>:DATA?....1825
SEARch:RESult:ARINc:WORD<m>:LABel?....1825
SEARch:RESult:ARINc:WORD<m>:PATTern?....1825
SEARch:RESult:ARINc:WORD<m>:SDI?....1826
SEARch:RESult:ARINc:WORD<m>:SSM?....1825
SEARch:RESult:ARINc:WORD<m>:STARt?....1827
SEARch:RESult:ARINc:WORD<m>:STATe?....1827
SEARch:RESult:ARINc:WORD<m>:STOP?....1826
SEARch:RESult:ARINc:WORD<m>:SYMBol?....1826
SEARch:RESult:CAN:FCOunt?....1741
SEARch:RESult:CAN:FDATa:FRAMe<m>:SCValue?....1745

SEARch:RESult:CAN:FDATa:FRAMe<m>:STANdard?....1745
SEARch:RESult:CAN:FRAMe<m>:ACKState?....1743
SEARch:RESult:CAN:FRAMe<m>:ACKValue?....1741
SEARch:RESult:CAN:FRAMe<m>:BSEPosition?....1742
SEARch:RESult:CAN:FRAMe<m>:BYTE<n>:STATe?....1742
SEARch:RESult:CAN:FRAMe<m>:BYTE<n>:VALue?....1742
SEARch:RESult:CAN:FRAMe<m>:CSSTate?....1743
SEARch:RESult:CAN:FRAMe<m>:CSValue?....1743
SEARch:RESult:CAN:FRAMe<m>:DATA?....1743
SEARch:RESult:CAN:FRAMe<m>:DLCState?....1743
SEARch:RESult:CAN:FRAMe<m>:DLCValue?....1744
SEARch:RESult:CAN:FRAMe<m>:FERCause?....1744
SEARch:RESult:CAN:FRAMe<m>:IDSTate?....1743
SEARch:RESult:CAN:FRAMe<m>:IDTYpe?....1744
SEARch:RESult:CAN:FRAMe<m>:IDValue?....1744
SEARch:RESult:CAN:FRAMe<m>:NDBYtes?....1737
SEARch:RESult:CAN:FRAMe<m>:SDATa?....1749
SEARch:RESult:CAN:FRAMe<m>:STARt?....1745
SEARch:RESult:CAN:FRAMe<m>:STATus?....1746
SEARch:RESult:CAN:FRAMe<m>:STOP?....1746
SEARch:RESult:CAN:FRAMe<m>:SYMBol?....1746
SEARch:RESult:CAN:FRAMe<m>:TYPE?....1747
SEARch:RESult:CMSB<m>:FCOunt?....2082
SEARch:RESult:CMSB<m>:FRAMe<n>:CCOunt?....2083
SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:NAME?....2083
SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:STARt?....2083
SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:STATe?....2084
SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:STOP?....2084
SEARch:RESult:CMSB<m>:FRAMe<n>:CELL<o>:VALue?....2085
SEARch:RESult:CMSB<m>:FRAMe<n>:CONE?....2085
SEARch:RESult:CMSB<m>:FRAMe<n>:CTHRee?....2086
SEARch:RESult:CMSB<m>:FRAMe<n>:CTWO?....2086
SEARch:RESult:CMSB<m>:FRAMe<n>:STARt?....2086
SEARch:RESult:CMSB<m>:FRAMe<n>:STATe?....2087
SEARch:RESult:CMSB<m>:FRAMe<n>:STOP?....2087
SEARch:RESult:CMSB<m>:FRAMe<n>:TYPE?....2088
SEARch:RESult:DDR:FCOunt?....2305
SEARch:RESult:DDR:FRAMe<m>:FLD<n>:FVAL?....2306
SEARch:RESult:DDR:FRAMe<m>:FLD<n>:STARt?....2306
SEARch:RESult:DDR:FRAMe<m>:FLD<n>:STATus?....2306
SEARch:RESult:DDR:FRAMe<m>:FLD<n>:STOP?....2307
SEARch:RESult:DDR:FRAMe<m>:FLD<n>:VAL?....2307
SEARch:RESult:DDR:FRAMe<m>:NWRDs?....2308
SEARch:RESult:DDR:FRAMe<m>:STARt?....2308
SEARch:RESult:DDR:FRAMe<m>:STATe?....2308
SEARch:RESult:DDR:FRAMe<m>:STOP?....2309
SEARch:RESult:DDR:FRAMe<m>:TYPE?....2309
SEARch:RESult:DPHY:FCOunt?....1978
SEARch:RESult:DPHY:FRAMe<m>:CS?....1978
SEARch:RESult:DPHY:FRAMe<m>:DATA?....1978

SEARch:RESult:DPHY:FRAMe<m>:DTName?....1978
SEARch:RESult:DPHY:FRAMe<m>:DTYPe?....1979
SEARch:RESult:DPHY:FRAMe<m>:ECC?....1979
SEARch:RESult:DPHY:FRAMe<m>:PACKet<n>:IDX?....1979
SEARch:RESult:DPHY:FRAMe<m>:PACKet<n>:VALue?....1980
SEARch:RESult:DPHY:FRAMe<m>:STARt?....1980
SEARch:RESult:DPHY:FRAMe<m>:STATe?....1980
SEARch:RESult:DPHY:FRAMe<m>:STOP?....1981
SEARch:RESult:DPHY:FRAMe<m>:TYPE?....1981
SEARch:RESult:DPHY:FRAMe<m>:VCHannel?....1982
SEARch:RESult:EBTB:SCOunt?....2102
SEARch:RESult:EBTB:SYMBol<m>:DATA?....2102
SEARch:RESult:EBTB:SYMBol<m>:STARt?....2103
SEARch:RESult:EBTB:SYMBol<m>:STATus?....2103
SEARch:RESult:EBTB:SYMBol<m>:STOP?....2103
SEARch:RESult:ETHernet:WCOunt?....1848
SEARch:RESult:ETHernet:WORD<m>:BYTE<n>:VALue?....1852
SEARch:RESult:ETHernet:WORD<m>:CRC?....1851
SEARch:RESult:ETHernet:WORD<m>:DATA?....1850
SEARch:RESult:ETHernet:WORD<m>:DESTaddress?....1849
SEARch:RESult:ETHernet:WORD<m>:DSYMbol?....1851
SEARch:RESult:ETHernet:WORD<m>:FTYPe?....1850
SEARch:RESult:ETHernet:WORD<m>:SRCaddress?....1850
SEARch:RESult:ETHernet:WORD<m>:SSYMbol?....1851
SEARch:RESult:ETHernet:WORD<m>:STARt?....1849
SEARch:RESult:ETHernet:WORD<m>:STATe?....1848
SEARch:RESult:ETHernet:WORD<m>:STOP?....1849
SEARch:RESult:ETHernet:WORD<m>:TYPE?....1850
SEARch:RESult:HBTO:FCOunt?....1878
SEARch:RESult:HBTO:FRAMe<m>:CRC?....1881
SEARch:RESult:HBTO:FRAMe<m>:DATA?....1881
SEARch:RESult:HBTO:FRAMe<m>:DESTaddress?....1880
SEARch:RESult:HBTO:FRAMe<m>:DSYMbol?....1882
SEARch:RESult:HBTO:FRAMe<m>:FTYPe?....1882
SEARch:RESult:HBTO:FRAMe<m>:NUMWords?....1881
SEARch:RESult:HBTO:FRAMe<m>:SRCaddress?....1880
SEARch:RESult:HBTO:FRAMe<m>:SSYMbol?....1882
SEARch:RESult:HBTO:FRAMe<m>:STARt?....1879
SEARch:RESult:HBTO:FRAMe<m>:STATe?....1879
SEARch:RESult:HBTO:FRAMe<m>:STOP?....1880
SEARch:RESult:HBTO:FRAMe<m>:TYPE?....1879
SEARch:RESult:HBTO:FRAMe<m>:WORD<n>:TYPE?....1883
SEARch:RESult:HBTO:FRAMe<m>:WORD<n>:VALue?....1883
SEARch:RESult:I2C:FCOunt?....1671
SEARch:RESult:I2C:FRAMe<m>:AACCess?....1672
SEARch:RESult:I2C:FRAMe<m>:ACCess?....1672
SEARch:RESult:I2C:FRAMe<m>:ACOMplete?....1673
SEARch:RESult:I2C:FRAMe<m>:ADBStart?....1673
SEARch:RESult:I2C:FRAMe<m>:ADDRess?....1673
SEARch:RESult:I2C:FRAMe<m>:ADEVice?....1674

SEARch:RESult:I2C:FRAMe<m>:AMODe? 1674
SEARch:RESult:I2C:FRAMe<m>:ASTart? 1674
SEARch:RESult:I2C:FRAMe<m>:BCOunt? 1675
SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:ACCess? 1675
SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:ACKStart? 1676
SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:COMPlete? 1676
SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:STARt? 1676
SEARch:RESult:I2C:FRAMe<m>:BYTE<n>:VALue? 1677
SEARch:RESult:I2C:FRAMe<m>:DATA? 1674
SEARch:RESult:I2C:FRAMe<m>:RWBStart? 1675
SEARch:RESult:I2C:FRAMe<m>:STARt? 1672
SEARch:RESult:I2C:FRAMe<m>:STATus? 1671
SEARch:RESult:I2C:FRAMe<m>:STOP? 1672
SEARch:RESult:I2C:FRAMe<m>:SYMBol? 1675
SEARch:RESult:LIMit 1602
SEARch:RESult:LIN:FCOunt? 1777
SEARch:RESult:LIN:FRAMe<m>:BYTE<n>:STATe? 1782
SEARch:RESult:LIN:FRAMe<m>:BYTE<n>:VALue? 1782
SEARch:RESult:LIN:FRAMe<m>:CSSTate? 1779
SEARch:RESult:LIN:FRAMe<m>:CSValue? 1780
SEARch:RESult:LIN:FRAMe<m>:DATA? 1779
SEARch:RESult:LIN:FRAMe<m>:IDPValue? 1781
SEARch:RESult:LIN:FRAMe<m>:IDSTate? 1780
SEARch:RESult:LIN:FRAMe<m>:IDValue? 1780
SEARch:RESult:LIN:FRAMe<m>:SDATa? 1779
SEARch:RESult:LIN:FRAMe<m>:SDEXport? 1779
SEARch:RESult:LIN:FRAMe<m>:STARt? 1778
SEARch:RESult:LIN:FRAMe<m>:STATus? 1778
SEARch:RESult:LIN:FRAMe<m>:STOP? 1778
SEARch:RESult:LIN:FRAMe<m>:SYMBol? 1781
SEARch:RESult:LIN:FRAMe<m>:SYSTate? 1781
SEARch:RESult:LIN:FRAMe<m>:VERSion? 1782
SEARch:RESult:MDIO:WCOunt? 2119
SEARch:RESult:MDIO:WORD<m>:DATA? 2116
SEARch:RESult:MDIO:WORD<m>:PHYS? 2116
SEARch:RESult:MDIO:WORD<m>:REGI? 2117
SEARch:RESult:MDIO:WORD<m>:ST? 2117
SEARch:RESult:MDIO:WORD<m>:STARt? 2117
SEARch:RESult:MDIO:WORD<m>:STATe? 2118
SEARch:RESult:MDIO:WORD<m>:STOP? 2118
SEARch:RESult:MDIO:WORD<m>:SYMBol? 2119
SEARch:RESult:MDIO:WORD<m>:TYPE? 2119
SEARch:RESult:MILStd:WCOunt? 1808
SEARch:RESult:MILStd:WORD<m>:DATA? 1810
SEARch:RESult:MILStd:WORD<m>:INFO? 1810
SEARch:RESult:MILStd:WORD<m>:RTADdress? 1809
SEARch:RESult:MILStd:WORD<m>:STARt? 1808
SEARch:RESult:MILStd:WORD<m>:STATus? 1808
SEARch:RESult:MILStd:WORD<m>:STOP? 1809
SEARch:RESult:MILStd:WORD<m>:SYMBol? 1809

SEARch:RESult:MILStd:WORD<m>:TYPE? 1808
SEARch:RESult:MPHY:FCOunt? 2026
SEARch:RESult:MPHY:FRAMe<m>:CCOunt? 2026
SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:DATA? 2027
SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:STARt? 2027
SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:STATe? 2028
SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:STOP? 2028
SEARch:RESult:MPHY:FRAMe<m>:CELL<n>:TYPE? 2027
SEARch:RESult:MPHY:FRAMe<m>:DATA? 2028
SEARch:RESult:MPHY:FRAMe<m>:FTYPe? 2029
SEARch:RESult:MPHY:FRAMe<m>:STARt? 2030
SEARch:RESult:MPHY:FRAMe<m>:STATe? 2029
SEARch:RESult:MPHY:FRAMe<m>:STOP? 2031
SEARch:RESult:PCIE:PACKet<m>:ADDR? 2287
SEARch:RESult:PCIE:PACKet<m>:CPID? 2287
SEARch:RESult:PCIE:PACKet<m>:CPS? 2288
SEARch:RESult:PCIE:PACKet<m>:DATA? 2288
SEARch:RESult:PCIE:PACKet<m>:DFC? 2288
SEARch:RESult:PCIE:PACKet<m>:FCOunt? 2289
SEARch:RESult:PCIE:PACKet<m>:FLD<n>:FVAL? 2289
SEARch:RESult:PCIE:PACKet<m>:FLD<n>:STATus? 2289
SEARch:RESult:PCIE:PACKet<m>:FLD<n>:TYPE? 2289
SEARch:RESult:PCIE:PACKet<m>:FLD<n>:VAL? 2290
SEARch:RESult:PCIE:PACKet<m>:HFC? 2290
SEARch:RESult:PCIE:PACKet<m>:LEN? 2290
SEARch:RESult:PCIE:PACKet<m>:RQID? 2291
SEARch:RESult:PCIE:PACKet<m>:SEQN? 2291
SEARch:RESult:PCIE:PACKet<m>:STARt? 2291
SEARch:RESult:PCIE:PACKet<m>:STATe? 2291
SEARch:RESult:PCIE:PACKet<m>:STOP? 2292
SEARch:RESult:PCIE:PACKet<m>:TAG? 2292
SEARch:RESult:PCIE:PACKet<m>:TYPE? 2292
SEARch:RESult:PCIE:PCOunt? 2293
SEARch:RESult:RFFE:FCOunt? 1945
SEARch:RESult:RFFE:SEQuence<m>:ADDRess? 1947
SEARch:RESult:RFFE:SEQuence<m>:BCOunt? 1947
SEARch:RESult:RFFE:SEQuence<m>:BYTE<n>:STATe? 1948
SEARch:RESult:RFFE:SEQuence<m>:BYTE<n>:VALue? 1948
SEARch:RESult:RFFE:SEQuence<m>:DATA? 1948
SEARch:RESult:RFFE:SEQuence<m>:SADD? 1946
SEARch:RESult:RFFE:SEQuence<m>:STARt? 1946
SEARch:RESult:RFFE:SEQuence<m>:STATe? 1946
SEARch:RESult:RFFE:SEQuence<m>:STOP? 1947
SEARch:RESult:RFFE:SEQuence<m>:SYMBol? 1948
SEARch:RESult:RFFE:SEQuence<m>:TYPE? 1945
SEARch:RESult:SHOW 1602
SEARch:RESult:SORT:ASCending 1603
SEARch:RESult:SORT[:MODE] 1603
SEARch:RESult:SPI:FCOunt? 1694
SEARch:RESult:SPI:FRAMe<m>:COUNt? 1694

SEARch:RESult:SPI:FRAMe<m>:DATA? 1694
SEARch:RESult:SPI:FRAMe<m>:STARt? 1695
SEARch:RESult:SPI:FRAMe<m>:STATus? 1695
SEARch:RESult:SPI:FRAMe<m>:STOP? 1695
SEARch:RESult:SPI:FRAMe<m>:WCOunt? 1696
SEARch:RESult:SPI:FRAMe<m>:WORD<n>:MISO? 1696
SEARch:RESult:SPI:FRAMe<m>:WORD<n>:MOSI? 1696
SEARch:RESult:SPI:FRAMe<m>:WORD<n>:STARt? 1696
SEARch:RESult:SPI:FRAMe<m>:WORD<n>:STOP? 1697
SEARch:RESult:SWIRe:FCOunt? 2241
SEARch:RESult:SWIRe:FRAMe<m>:DATA? 2242
SEARch:RESult:SWIRe:FRAMe<m>:STARt? 2242
SEARch:RESult:SWIRe:FRAMe<m>:STATe? 2242
SEARch:RESult:SWIRe:FRAMe<m>:STOP? 2243
SEARch:RESult:SWIRe:FRAMe<m>:TYPE? 2243
SEARch:RESult:TBTO:FCOunt? 1911
SEARch:RESult:TBTO:FRAMe<m>:CRC? 1911
SEARch:RESult:TBTO:FRAMe<m>:DATA? 1911
SEARch:RESult:TBTO:FRAMe<m>:DESTaddress? 1911
SEARch:RESult:TBTO:FRAMe<m>:DSYMbol? 1912
SEARch:RESult:TBTO:FRAMe<m>:FTYPe? 1912
SEARch:RESult:TBTO:FRAMe<m>:NUMWords? 1913
SEARch:RESult:TBTO:FRAMe<m>:SRCaddress? 1913
SEARch:RESult:TBTO:FRAMe<m>:SSYMbol? 1913
SEARch:RESult:TBTO:FRAMe<m>:STARt? 1914
SEARch:RESult:TBTO:FRAMe<m>:STATe? 1914
SEARch:RESult:TBTO:FRAMe<m>:STOP? 1914
SEARch:RESult:TBTO:FRAMe<m>:TYPE? 1915
SEARch:RESult:TBTO:FRAMe<m>:WORD<n>:TYPE? 1915
SEARch:RESult:TBTO:FRAMe<m>:WORD<n>:VALue? 1915
SEARch:RESult:TBTO:FRAMe<m>:WORD<n>:VSTR? 1916
SEARch:RESult:USB:PACKet<m>:ADDRess? 2157
SEARch:RESult:USB:PACKet<m>:CRC? 2157
SEARch:RESult:USB:PACKet<m>:DATA? 2158
SEARch:RESult:USB:PACKet<m>:ENDPoint? 2158
SEARch:RESult:USB:PACKet<m>:ET? 2158
SEARch:RESult:USB:PACKet<m>:FRAMe? 2159
SEARch:RESult:USB:PACKet<m>:PID? 2159
SEARch:RESult:USB:PACKet<m>:PORT? 2159
SEARch:RESult:USB:PACKet<m>:SC? 2160
SEARch:RESult:USB:PACKet<m>:SEU? 2160
SEARch:RESult:USB:PACKet<m>:STARt? 2160
SEARch:RESult:USB:PACKet<m>:STATus? 2161
SEARch:RESult:USB:PACKet<m>:STOP? 2161
SEARch:RESult:USB:PCOunt? 2161
SEARch:RESult:USBPd:FCOunt? 2220
SEARch:RESult:USBPd:FRAMe<m>:DATA? 2221
SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:FVAL? 2221
SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:STARt? 2222
SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:STATus? 2221

SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:STOP? 2222
SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:TYPE? 2223
SEARch:RESult:USBPd:FRAMe<m>:FLD<n>:VAL? 2223
SEARch:RESult:USBPd:FRAMe<m>:INFO? 2223
SEARch:RESult:USBPd:FRAMe<m>:STARt? 2224
SEARch:RESult:USBPd:FRAMe<m>:STATe? 2224
SEARch:RESult:USBPd:FRAMe<m>:STOP? 2225
SEARch:RESult:USBPd:FRAMe<m>:TYPE? 2225
SEARch:RESult:USBThree:FCOunt? 2197
SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:FVAL? 2200
SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:STARt? 2199
SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:STATus? 2198
SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:STOP? 2199
SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:TYPE? 2198
SEARch:RESult:USBThree:FRAMe<m>:FLD<n>:VAL? 2199
SEARch:RESult:USBThree:FRAMe<m>:INFO? 2196
SEARch:RESult:USBThree:FRAMe<m>:STARt? 2197
SEARch:RESult:USBThree:FRAMe<m>:STATe? 2197
SEARch:RESult:USBThree:FRAMe<m>:STOP? 2198
SEARch:RESult:USBThree:FRAMe<m>:TYPE? 2196
SEARch:RESult[:ALL]? 1603
SEARch:SOURce 1566
SEARch:TRIGger:ARINc:DATA:CONDition 1822
SEARch:TRIGger:ARINc:DATA:MAX 1823
SEARch:TRIGger:ARINc:DATA:MIN 1822
SEARch:TRIGger:ARINc:ERRor:CODing 1823
SEARch:TRIGger:ARINc:ERRor:PARity 1824
SEARch:TRIGger:ARINc:ERRor:TIMing 1824
SEARch:TRIGger:ARINc:LABel:CONDition 1822
SEARch:TRIGger:ARINc:LABel:MAX 1823
SEARch:TRIGger:ARINc:LABel:MIN 1822
SEARch:TRIGger:ARINc:SDI 1823
SEARch:TRIGger:ARINc:SSM 1823
SEARch:TRIGger:ARINc:TYPE 1821
SEARch:TRIGger:CAN:ACKerror 1738
SEARch:TRIGger:CAN:BITSterror 1738
SEARch:TRIGger:CAN:CRCerror 1739
SEARch:TRIGger:CAN:DCONdition 1735
SEARch:TRIGger:CAN:DLC 1736
SEARch:TRIGger:CAN:DLCCondition 1736
SEARch:TRIGger:CAN:DMAX 1736
SEARch:TRIGger:CAN:DMIN 1735
SEARch:TRIGger:CAN:FDATa:BRS 1740
SEARch:TRIGger:CAN:FDATa:DPOPerator 1737
SEARch:TRIGger:CAN:FDATa:DPOSition 1738
SEARch:TRIGger:CAN:FDATa:DPTO 1738
SEARch:TRIGger:CAN:FDATa:ESI 1740
SEARch:TRIGger:CAN:FDATa:SCERror 1739
SEARch:TRIGger:CAN:FDATa:STANdard 1733
SEARch:TRIGger:CAN:FDATa[:FDF] 1740

SEARch:TRIGger:CAN:FORMerror....1739
SEARch:TRIGger:CAN:FTYPe....1733
SEARch:TRIGger:CAN:ICONdition....1734
SEARch:TRIGger:CAN:IMAX....1734
SEARch:TRIGger:CAN:IMIN....1735
SEARch:TRIGger:CAN:ITYPe....1734
SEARch:TRIGger:CAN:SERRor....1733
SEARch:TRIGger:CAN:SFIDentifier....1732
SEARch:TRIGger:CAN:SFTYpe....1732
SEARch:TRIGger:CAN:SIDData....1732
SEARch:TRIGger:CAN:SSYMbolic....1750
SEARch:TRIGger:CAN:SYMBolic:DMAX....1751
SEARch:TRIGger:CAN:SYMBolic:DMIN....1751
SEARch:TRIGger:CAN:SYMBolic:MSGValue....1750
SEARch:TRIGger:CAN:SYMBolic:SGEValue....1752
SEARch:TRIGger:CAN:SYMBolic:SIGValue....1751
SEARch:TRIGger:CAN:SYMBolic:SSIGnals....1750
SEARch:TRIGger:CAN[:SSOFrame]....1732
SEARch:TRIGger:CMSB:BIT....2078
SEARch:TRIGger:CMSB:DMAX....2079
SEARch:TRIGger:CMSB:DMIN....2079
SEARch:TRIGger:CMSB:DOPerator....2080
SEARch:TRIGger:CMSB:ERENable....2077
SEARch:TRIGger:CMSB:ERRor<m>:ENABle....2077
SEARch:TRIGger:CMSB:FIENable....2078
SEARch:TRIGger:CMSB:FRAMe<m>:ENABle....2080
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:BIT....2078
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMAX....2079
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DMIN....2079
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:DOPerator....2080
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:ENABle....2080
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMAX....2081
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IMIN....2081
SEARch:TRIGger:CMSB:FRAMe<m>:FLD<n>:IOPerator....2081
SEARch:TRIGger:CMSB:FRENable....2078
SEARch:TRIGger:CMSB:IMAX....2081
SEARch:TRIGger:CMSB:IMIN....2081
SEARch:TRIGger:CMSB:IOPerator....2081
SEARch:TRIGger:DDR:DMAX....2302
SEARch:TRIGger:DDR:DMIN....2302
SEARch:TRIGger:DDR:DOPerator....2303
SEARch:TRIGger:DDR:ERENable....2301
SEARch:TRIGger:DDR:ERRor<m>:ENABle....2301
SEARch:TRIGger:DDR:FIENable....2301
SEARch:TRIGger:DDR:FRAMe<m>:ENABle....2303
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMAX....2302
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DMIN....2302
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:DOPerator....2303
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMAX....2304
SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IMIN....2304

SEARch:TRIGger:DDR:FRAMe<m>:FLD<n>:IOPerator....2305
SEARch:TRIGger:DDR:FRENable....2302
SEARch:TRIGger:DDR:FRENable....2303
SEARch:TRIGger:DDR:IMAX....2304
SEARch:TRIGger:DDR:IMIN....2304
SEARch:TRIGger:DDR:IOPerator....2305
SEARch:TRIGger:DPHY:DATA:CONDition....1969
SEARch:TRIGger:DPHY:DATA:MAX....1970
SEARch:TRIGger:DPHY:DATA:MIN....1970
SEARch:TRIGger:DPHY:DIDX:CONDition....1971
SEARch:TRIGger:DPHY:DIDX:MAX....1971
SEARch:TRIGger:DPHY:DIDX:MIN....1971
SEARch:TRIGger:DPHY:DTYPe:CONDition....1972
SEARch:TRIGger:DPHY:DTYPe:MAX....1972
SEARch:TRIGger:DPHY:DTYPe:MIN....1972
SEARch:TRIGger:DPHY:ESCMode:CONDition....1973
SEARch:TRIGger:DPHY:ESCMode:MAX....1973
SEARch:TRIGger:DPHY:ESCMode:MIN....1973
SEARch:TRIGger:DPHY:ESDTa:CONDition....1974
SEARch:TRIGger:DPHY:ESDTa:MAX....1974
SEARch:TRIGger:DPHY:ESDTa:MIN....1975
SEARch:TRIGger:DPHY:ESINdex:CONDition....1975
SEARch:TRIGger:DPHY:ESINdex:MAX....1975
SEARch:TRIGger:DPHY:ESINdex:MIN....1976
SEARch:TRIGger:DPHY:HSVC....1976
SEARch:TRIGger:DPHY:TYPE....1969
SEARch:TRIGger:DPHY:WCOunt:CONDition....1976
SEARch:TRIGger:DPHY:WCOunt:MAX....1977
SEARch:TRIGger:DPHY:WCOunt:MIN....1977
SEARch:TRIGger:EBTB:DISParityerr....2099
SEARch:TRIGger:EBTB:DX....2099
SEARch:TRIGger:EBTB:DY....2100
SEARch:TRIGger:EBTB:EBPA....2100
SEARch:TRIGger:EBTB:GLITcherror....2100
SEARch:TRIGger:EBTB:STYPe....2100
SEARch:TRIGger:EBTB:SYME....2101
SEARch:TRIGger:EBTB:SYMType....2101
SEARch:TRIGger:EBTB:TBPA....2101
SEARch:TRIGger:EBTB:UNK....2102
SEARch:TRIGger:EDGE:ACOPy....1569
SEARch:TRIGger:EDGE:BCOPy....1570
SEARch:TRIGger:EDGE:SLOPe....1570
SEARch:TRIGger:EDGE[:STATe]....1568
SEARch:TRIGger:ETHernet:ERRor:LENGth....1847
SEARch:TRIGger:ETHernet:ERRor:PREamble....1847
SEARch:TRIGger:ETHernet:FRAMe:CCONdition....1846
SEARch:TRIGger:ETHernet:FRAMe:CMAX....1847
SEARch:TRIGger:ETHernet:FRAMe:CMIN....1846
SEARch:TRIGger:ETHernet:FRAMe:DCONdition....1843
SEARch:TRIGger:ETHernet:FRAMe:DMAX....1843

SEARch:TRIGger:ETHernet:FRAMe:DMIN....1843
SEARch:TRIGger:ETHernet:FRAMe:SCONdition....1844
SEARch:TRIGger:ETHernet:FRAMe:SMAX....1844
SEARch:TRIGger:ETHernet:FRAMe:SMIN....1844
SEARch:TRIGger:ETHernet:FRAMe:TCONdition....1845
SEARch:TRIGger:ETHernet:FRAMe:TMAX....1846
SEARch:TRIGger:ETHernet:FRAMe:TMIN....1845
SEARch:TRIGger:ETHernet:TYPE....1842
SEARch:TRIGger:GLITch:ACOPy....1569
SEARch:TRIGger:GLITch:POLarity....1571
SEARch:TRIGger:GLITch:RANGe....1571
SEARch:TRIGger:GLITch:WIDTh....1571
SEARch:TRIGger:GLITch[:STATe]....1568
SEARch:TRIGger:HBTO:CRC:CONDition....1874
SEARch:TRIGger:HBTO:CRC:MAX....1875
SEARch:TRIGger:HBTO:CRC:MIN....1874
SEARch:TRIGger:HBTO:DADDress:CONDition....1871
SEARch:TRIGger:HBTO:DADDress:MAX....1871
SEARch:TRIGger:HBTO:DADDress:MIN....1871
SEARch:TRIGger:HBTO:DATA:DCONdition....1875
SEARch:TRIGger:HBTO:DATA:DMAX....1876
SEARch:TRIGger:HBTO:DATA:DMIN....1875
SEARch:TRIGger:HBTO:DATA:ICONdition....1876
SEARch:TRIGger:HBTO:DATA:IMAX....1877
SEARch:TRIGger:HBTO:DATA:IMIN....1877
SEARch:TRIGger:HBTO:ERRor:CRC....1877
SEARch:TRIGger:HBTO:ERRor:PREamble....1877
SEARch:TRIGger:HBTO:ERRor:SFD....1878
SEARch:TRIGger:HBTO:LENGth:CONDition....1873
SEARch:TRIGger:HBTO:LENGth:MAX....1874
SEARch:TRIGger:HBTO:LENGth:MIN....1873
SEARch:TRIGger:HBTO:SADDress:CONDition....1872
SEARch:TRIGger:HBTO:SADDress:MAX....1873
SEARch:TRIGger:HBTO:SADDress:MIN....1872
SEARch:TRIGger:HBTO:TYPE....1870
SEARch:TRIGger:I2C:ACCess....1665
SEARch:TRIGger:I2C:ACONdition....1664
SEARch:TRIGger:I2C:ADData....1663
SEARch:TRIGger:I2C:ADDO<m>:ADRType....1666
SEARch:TRIGger:I2C:ADDO<m>:ENABle....1666
SEARch:TRIGger:I2C:ADDO<m>:RWBit....1667
SEARch:TRIGger:I2C:ADDO<m>[:VALue]....1666
SEARch:TRIGger:I2C:ADDRess....1665
SEARch:TRIGger:I2C:ADDTo....1665
SEARch:TRIGger:I2C:ADNack....1670
SEARch:TRIGger:I2C:ADOR....1663
SEARch:TRIGger:I2C:AMODe....1664
SEARch:TRIGger:I2C:DCONdition....1668
SEARch:TRIGger:I2C:DMAX....1669
SEARch:TRIGger:I2C:DMIN....1669

SEARch:TRIGger:I2C:DPOPerator ... 1667
SEARch:TRIGger:I2C:DPOSition ... 1668
SEARch:TRIGger:I2C:DPTO ... 1668
SEARch:TRIGger:I2C:DRNack ... 1670
SEARch:TRIGger:I2C:DWNack ... 1670
SEARch:TRIGger:I2C:NACKnowledge ... 1662
SEARch:TRIGger:I2C:RCONdition ... 1662
SEARch:TRIGger:I2C:SADDress ... 1662
SEARch:TRIGger:I2C:SCONdition ... 1661
SEARch:TRIGger:I2C:STCNdition ... 1662
SEARch:TRIGger:INTerval:ACOPy ... 1569
SEARch:TRIGger:INTerval:DELTa ... 1572
SEARch:TRIGger:INTerval:RANGe ... 1573
SEARch:TRIGger:INTerval:SLOPe ... 1572
SEARch:TRIGger:INTerval:WIDTh ... 1573
SEARch:TRIGger:INTerval[:STATe] ... 1568
SEARch:TRIGger:LEVel:RUNT:LOWer ... 1575
SEARch:TRIGger:LEVel:RUNT:UPPer ... 1575
SEARch:TRIGger:LEVel:TRANsition:LOWer ... 1578
SEARch:TRIGger:LEVel:TRANsition:UPPer ... 1578
SEARch:TRIGger:LEVel:WINDow:LOWer ... 1584
SEARch:TRIGger:LEVel:WINDow:UPPer ... 1584
SEARch:TRIGger:LEVel[:VALue] ... 1569
SEARch:TRIGger:LIN:BORDer ... 1774
SEARch:TRIGger:LIN:CHKSerror ... 1776
SEARch:TRIGger:LIN:CRCDatalen ... 1776
SEARch:TRIGger:LIN:DCONdition ... 1773
SEARch:TRIGger:LIN:DLECondition ... 1774
SEARch:TRIGger:LIN:DLENgth ... 1775
SEARch:TRIGger:LIN:DMAX ... 1773
SEARch:TRIGger:LIN:DMIN ... 1773
SEARch:TRIGger:LIN:ERRPattern ... 1776
SEARch:TRIGger:LIN:ICONdition ... 1771
SEARch:TRIGger:LIN:IDENtifieror ... 1770
SEARch:TRIGger:LIN:IDOR<m>:ENABle ... 1772
SEARch:TRIGger:LIN:IDOR<m>[:VALue] ... 1772
SEARch:TRIGger:LIN:IMAX ... 1772
SEARch:TRIGger:LIN:IMIN ... 1771
SEARch:TRIGger:LIN:IPERror ... 1775
SEARch:TRIGger:LIN:SERRor ... 1770
SEARch:TRIGger:LIN:SFIDentifier ... 1770
SEARch:TRIGger:LIN:SIDData ... 1770
SEARch:TRIGger:LIN:SSOFrame ... 1769
SEARch:TRIGger:LIN:STANdard ... 1777
SEARch:TRIGger:LIN:SYERror ... 1775
SEARch:TRIGger:LIN:WUFRame ... 1771
SEARch:TRIGger:MDIO:DATA ... 2113
SEARch:TRIGger:MDIO:FRAMetype ... 2114
SEARch:TRIGger:MDIO:PHYS ... 2114
SEARch:TRIGger:MDIO:REGI ... 2115

SEARch:TRIGger:MDIO:ST....2115
SEARch:TRIGger:MDIO:TYPE....2115
SEARch:TRIGger:MILStd:CDST:ICONdition....1803
SEARch:TRIGger:MILStd:CDST:IMAX....1804
SEARch:TRIGger:MILStd:CDST:IMIN....1804
SEARch:TRIGger:MILStd:CDST:RCONdition....1802
SEARch:TRIGger:MILStd:CDST:RMAX....1803
SEARch:TRIGger:MILStd:CDST:RMIN....1802
SEARch:TRIGger:MILStd:CMD:CCONdition....1803
SEARch:TRIGger:MILStd:CMD:CMAX....1804
SEARch:TRIGger:MILStd:CMD:CMIN....1804
SEARch:TRIGger:MILStd:CMD:RCONdition....1802
SEARch:TRIGger:MILStd:CMD:RMAX....1803
SEARch:TRIGger:MILStd:CMD:RMIN....1802
SEARch:TRIGger:MILStd:CMD:SCONdition....1803
SEARch:TRIGger:MILStd:CMD:SMAX....1804
SEARch:TRIGger:MILStd:CMD:SMIN....1804
SEARch:TRIGger:MILStd:CMD:TR....1806
SEARch:TRIGger:MILStd:DATA:DCONdition....1803
SEARch:TRIGger:MILStd:DATA:DMAX....1804
SEARch:TRIGger:MILStd:DATA:DMIN....1804
SEARch:TRIGger:MILStd:DATA:ICONdition....1805
SEARch:TRIGger:MILStd:DATA:IMAX....1805
SEARch:TRIGger:MILStd:DATA:IMIN....1805
SEARch:TRIGger:MILStd:DATA:RCONdition....1802
SEARch:TRIGger:MILStd:DATA:RMAX....1803
SEARch:TRIGger:MILStd:DATA:RMIN....1802
SEARch:TRIGger:MILStd:ERRor:MANChester....1806
SEARch:TRIGger:MILStd:ERRor:PARity....1806
SEARch:TRIGger:MILStd:ERRor:SYNC....1806
SEARch:TRIGger:MILStd:ERRor:TIMing....1806
SEARch:TRIGger:MILStd:STATus:BCReceived....1806
SEARch:TRIGger:MILStd:STATus:BUSY....1806
SEARch:TRIGger:MILStd:STATus:DBCaccept....1806
SEARch:TRIGger:MILStd:STATus:INSTrument....1806
SEARch:TRIGger:MILStd:STATus:MERRor....1806
SEARch:TRIGger:MILStd:STATus:SREQuest....1807
SEARch:TRIGger:MILStd:STATus:SUBSystem....1807
SEARch:TRIGger:MILStd:STATus:TERMinal....1807
SEARch:TRIGger:MILStd:TPSPecifier....1807
SEARch:TRIGger:MILStd:TYPE....1801
SEARch:TRIGger:MPHY:CRC:CONDition....2019
SEARch:TRIGger:MPHY:CRC:MAX....2020
SEARch:TRIGger:MPHY:CRC:MIN....2020
SEARch:TRIGger:MPHY:CREDit:CONDition....2021
SEARch:TRIGger:MPHY:CREDit:MAX....2022
SEARch:TRIGger:MPHY:CREDit:MIN....2022
SEARch:TRIGger:MPHY:CREQ....2020
SEARch:TRIGger:MPHY:DATA:DCONdition....2016
SEARch:TRIGger:MPHY:DATA:DMAX....2017

SEARch:TRIGger:MPHY:DATA:DMIN....2016
SEARch:TRIGger:MPHY:DATA:ICONdition....2017
SEARch:TRIGger:MPHY:DATA:IMAX....2018
SEARch:TRIGger:MPHY:DATA:IMIN....2018
SEARch:TRIGger:MPHY:ERRor:LCMD....2025
SEARch:TRIGger:MPHY:ERRor:LENGth....2025
SEARch:TRIGger:MPHY:ERRor:REServed....2024
SEARch:TRIGger:MPHY:ERRor:SYMBol....2025
SEARch:TRIGger:MPHY:ERRor:UNKNown....2025
SEARch:TRIGger:MPHY:FSNumber:CONDition....2018
SEARch:TRIGger:MPHY:FSNumber:MAX....2019
SEARch:TRIGger:MPHY:FSNumber:MIN....2019
SEARch:TRIGger:MPHY:LCCType....2010
SEARch:TRIGger:MPHY:LWFour:CONDition....2015
SEARch:TRIGger:MPHY:LWFour:MAX....2015
SEARch:TRIGger:MPHY:LWFour:MIN....2015
SEARch:TRIGger:MPHY:LWONe:CONDition....2011
SEARch:TRIGger:MPHY:LWONe:MAX....2012
SEARch:TRIGger:MPHY:LWONe:MIN....2012
SEARch:TRIGger:MPHY:LWTHree:CONDition....2014
SEARch:TRIGger:MPHY:LWTHree:MAX....2014
SEARch:TRIGger:MPHY:LWTHree:MIN....2014
SEARch:TRIGger:MPHY:LWTWo:CONDition....2012
SEARch:TRIGger:MPHY:LWTWo:MAX....2013
SEARch:TRIGger:MPHY:LWTWo:MIN....2013
SEARch:TRIGger:MPHY:PBEGin:CONDition....2022
SEARch:TRIGger:MPHY:PBEGin:MAX....2023
SEARch:TRIGger:MPHY:PBEGin:MIN....2023
SEARch:TRIGger:MPHY:PFID:CONDition....2023
SEARch:TRIGger:MPHY:PFID:MAX....2024
SEARch:TRIGger:MPHY:PFID:MIN....2024
SEARch:TRIGger:MPHY:RREQ....2021
SEARch:TRIGger:MPHY:SSIC:BIT....2039
SEARch:TRIGger:MPHY:SSIC:DMAX....2039
SEARch:TRIGger:MPHY:SSIC:DMIN....2040
SEARch:TRIGger:MPHY:SSIC:DOPerator....2040
SEARch:TRIGger:MPHY:SSIC:ERENable....2037
SEARch:TRIGger:MPHY:SSIC:ERRor<m>:ENABle....2037
SEARch:TRIGger:MPHY:SSIC:FIENable....2038
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:ENABle....2037
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:BIT....2039
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DMAX....2039
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DMIN....2040
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:DOPerator....2040
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:ENABle....2038
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:ENABle....2038
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IMAX....2041
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IMIN....2042
SEARch:TRIGger:MPHY:SSIC:FRAMe<m>:FLD<n>:IOPerator....2042
SEARch:TRIGger:MPHY:SSIC:FRENable....2037

SEARch:TRIGger:MPHY:SSIC:FRENable ..... 2041
SEARch:TRIGger:MPHY:SSIC:IMAX ..... 2041
SEARch:TRIGger:MPHY:SSIC:IMIN ..... 2042
SEARch:TRIGger:MPHY:SSIC:IOPerator ..... 2042
SEARch:TRIGger:MPHY:TC ..... 2016
SEARch:TRIGger:MPHY:TYPE ..... 2010
SEARch:TRIGger:PATTern:A:LOGic ..... 1587
SEARch:TRIGger:PATTern:A[:ENABle] ..... 1587
SEARch:TRIGger:PATTern:AB:LOGic ..... 1588
SEARch:TRIGger:PATTern:ABCD:LOGic ..... 1588
SEARch:TRIGger:PATTern:ACOPy ..... 1569
SEARch:TRIGger:PATTern:B:LOGic ..... 1587
SEARch:TRIGger:PATTern:B[:ENABle] ..... 1587
SEARch:TRIGger:PATTern:C:LOGic ..... 1587
SEARch:TRIGger:PATTern:C[:ENABle] ..... 1587
SEARch:TRIGger:PATTern:CD:LOGic ..... 1588
SEARch:TRIGger:PATTern:D:LOGic ..... 1587
SEARch:TRIGger:PATTern:D[:ENABle] ..... 1587
SEARch:TRIGger:PATTern:MODE ..... 1588
SEARch:TRIGger:PATTern:TIMeout:MODE ..... 1589
SEARch:TRIGger:PATTern:TIMeout[:TIME] ..... 1589
SEARch:TRIGger:PATTern:WIDTh:DELTa ..... 1591
SEARch:TRIGger:PATTern:WIDTh:RANGe ..... 1590
SEARch:TRIGger:PATTern:WIDTh[:WIDTh] ..... 1590
SEARch:TRIGger:PATTern[:STATe] ..... 1568
SEARch:TRIGger:PCIE:DLLP:FCTL:CRET ..... 2272
SEARch:TRIGger:PCIE:DLLP:FCTL:VCID ..... 2273
SEARch:TRIGger:PCIE:DLLP:MULT:AUTH ..... 2273
SEARch:TRIGger:PCIE:DLLP:MULT:CRET ..... 2273
SEARch:TRIGger:PCIE:DLLP:MULT:DVPT ..... 2273
SEARch:TRIGger:PCIE:DLLP:MULT:HABS ..... 2274
SEARch:TRIGger:PCIE:DLLP:MULT:MIXT ..... 2274
SEARch:TRIGger:PCIE:DLLP:MULT:PHAS ..... 2274
SEARch:TRIGger:PCIE:DLLP:MULT:RESA ..... 2274
SEARch:TRIGger:PCIE:DLLP:MULT:TLPT ..... 2275
SEARch:TRIGger:PCIE:DLLP:MULT:TYPE ..... 2275
SEARch:TRIGger:PCIE:DLLP:MULT:VHFC ..... 2275
SEARch:TRIGger:PCIE:DLLP:MULT:VHGR ..... 2276
SEARch:TRIGger:PCIE:DLLP:MULT:VLNR ..... 2276
SEARch:TRIGger:PCIE:DLLP:POWM ..... 2276
SEARch:TRIGger:PCIE:DLLP:SEQ ..... 2277
SEARch:TRIGger:PCIE:DLLP:TYPE ..... 2277
SEARch:TRIGger:PCIE:DLLP:VPAT ..... 2277
SEARch:TRIGger:PCIE:ERRC:CRC ..... 2277
SEARch:TRIGger:PCIE:ERRC:DISP ..... 2278
SEARch:TRIGger:PCIE:ERRC:ECRC ..... 2278
SEARch:TRIGger:PCIE:ERRC:INVP ..... 2278
SEARch:TRIGger:PCIE:ERRC:LCRC ..... 2278
SEARch:TRIGger:PCIE:ERRC:LENGth ..... 2279
SEARch:TRIGger:PCIE:ERRC:MALFormed ..... 2279

SEARch:TRIGger:PCIE:ERRC:MXPFix ... 2279
SEARch:TRIGger:PCIE:ERRC:PFORder ... 2279
SEARch:TRIGger:PCIE:ERRC:SYCBit ... 2280
SEARch:TRIGger:PCIE:ERRC:UKNP ... 2280
SEARch:TRIGger:PCIE:ERRC:VOORange ... 2280
SEARch:TRIGger:PCIE:OSET:COMP ... 2281
SEARch:TRIGger:PCIE:OSET:EIDE ... 2281
SEARch:TRIGger:PCIE:OSET:EIDL ... 2281
SEARch:TRIGger:PCIE:OSET:FTS ... 2281
SEARch:TRIGger:PCIE:OSET:SDS ... 2280
SEARch:TRIGger:PCIE:OSET:SKIP ... 2282
SEARch:TRIGger:PCIE:OSET:TSONe ... 2282
SEARch:TRIGger:PCIE:OSET:TSTWo ... 2282
SEARch:TRIGger:PCIE:TLP:ADRT ... 2282
SEARch:TRIGger:PCIE:TLP:CFGT ... 2283
SEARch:TRIGger:PCIE:TLP:CPID ... 2283
SEARch:TRIGger:PCIE:TLP:CPLS ... 2283
SEARch:TRIGger:PCIE:TLP:DEID ... 2284
SEARch:TRIGger:PCIE:TLP:MERW ... 2284
SEARch:TRIGger:PCIE:TLP:MSGC ... 2284
SEARch:TRIGger:PCIE:TLP:MSGR ... 2284
SEARch:TRIGger:PCIE:TLP:ORDE ... 2285
SEARch:TRIGger:PCIE:TLP:REID ... 2285
SEARch:TRIGger:PCIE:TLP:SNOO ... 2285
SEARch:TRIGger:PCIE:TLP:SNUM ... 2286
SEARch:TRIGger:PCIE:TLP:TCHN ... 2286
SEARch:TRIGger:PCIE:TLP:TYPE ... 2286
SEARch:TRIGger:PCIE:TYPE ... 2286
SEARch:TRIGger:RFFE:ADDRess:CONDition ... 1941
SEARch:TRIGger:RFFE:ADDRess:MAX ... 1943
SEARch:TRIGger:RFFE:ADDRess:MIN ... 1942
SEARch:TRIGger:RFFE:BCOunt:CONDition ... 1941
SEARch:TRIGger:RFFE:BCOunt:MAX ... 1943
SEARch:TRIGger:RFFE:BCOunt:MIN ... 1942
SEARch:TRIGger:RFFE:DATA:DCON ... 1941
SEARch:TRIGger:RFFE:DATA:DMAX ... 1943
SEARch:TRIGger:RFFE:DATA:DMIN ... 1942
SEARch:TRIGger:RFFE:DATA:ICONdition ... 1942
SEARch:TRIGger:RFFE:DATA:IMAX ... 1943
SEARch:TRIGger:RFFE:DATA:IMIN ... 1942
SEARch:TRIGger:RFFE:ERRor:GAP ... 1944
SEARch:TRIGger:RFFE:ERRor:LENGth ... 1943
SEARch:TRIGger:RFFE:ERRor:PARity ... 1944
SEARch:TRIGger:RFFE:ERRor:SSC ... 1944
SEARch:TRIGger:RFFE:ERRor:VERSion ... 1944
SEARch:TRIGger:RFFE:INTerrupt ... 1943
SEARch:TRIGger:RFFE:SADD:CONDition ... 1941
SEARch:TRIGger:RFFE:SADD:MAX ... 1943
SEARch:TRIGger:RFFE:SADD:MIN ... 1942
SEARch:TRIGger:RFFE:TYPE ... 1939

SEARch:TRIGger:RUNT:ACOPy....1569
SEARch:TRIGger:RUNT:DELTa....1574
SEARch:TRIGger:RUNT:POLarity....1574
SEARch:TRIGger:RUNT:RANGe....1574
SEARch:TRIGger:RUNT:WIDTh....1575
SEARch:TRIGger:RUNT[:STATe]....1568
SEARch:TRIGger:SETHold:ACOPy....1569
SEARch:TRIGger:SETHold:CEDGe....1584
SEARch:TRIGger:SETHold:CLEVel....1585
SEARch:TRIGger:SETHold:CSOurce....1585
SEARch:TRIGger:SETHold:HTIMe....1585
SEARch:TRIGger:SETHold:STIMe....1586
SEARch:TRIGger:SETHold[:STATe]....1568
SEARch:TRIGger:SLEWrate:ACOPy....1569
SEARch:TRIGger:SLEWrate:DELTa....1576
SEARch:TRIGger:SLEWrate:RANGe....1577
SEARch:TRIGger:SLEWrate:SLOPe....1577
SEARch:TRIGger:SLEWrate:TIME....1577
SEARch:TRIGger:SLEWrate[:STATe]....1568
SEARch:TRIGger:SPI:DPOPerator....1692
SEARch:TRIGger:SPI:DPOSition....1692
SEARch:TRIGger:SPI:DPTO....1693
SEARch:TRIGger:SPI:FCONdition....1691
SEARch:TRIGger:SPI:MISopattern....1692
SEARch:TRIGger:SPI:MOSipattern....1692
SEARch:TRIGger:SPI:PALignment....1693
SEARch:TRIGger:SPI:TYPE....1691
SEARch:TRIGger:STATe:A:LOGic....1593
SEARch:TRIGger:STATe:A[:ENABle]....1592
SEARch:TRIGger:STATe:AB:LOGic....1593
SEARch:TRIGger:STATe:ABCD:LOGic....1593
SEARch:TRIGger:STATe:ACOPy....1569
SEARch:TRIGger:STATe:B:LOGic....1593
SEARch:TRIGger:STATe:B[:ENABle]....1592
SEARch:TRIGger:STATe:C:LOGic....1593
SEARch:TRIGger:STATe:C[:ENABle]....1592
SEARch:TRIGger:STATe:CD:LOGic....1593
SEARch:TRIGger:STATe:CEDGe....1592
SEARch:TRIGger:STATe:CLEVel....1592
SEARch:TRIGger:STATe:CSOurce....1591
SEARch:TRIGger:STATe:D:LOGic....1593
SEARch:TRIGger:STATe:D[:ENABle]....1592
SEARch:TRIGger:STATe[:STATe]....1568
SEARch:TRIGger:SWIRe:CTYPe....2238
SEARch:TRIGger:SWIRe:DATA:CONDition....2238
SEARch:TRIGger:SWIRe:DATA:MAX....2239
SEARch:TRIGger:SWIRe:DATA:MIN....2239
SEARch:TRIGger:SWIRe:ERRor:ESC....2239
SEARch:TRIGger:SWIRe:ERRor:PARity....2239
SEARch:TRIGger:SWIRe:TIME:CONDition....2240

SEARch:TRIGger:SWIRe:TIME:MAX....2240
SEARch:TRIGger:SWIRe:TIME:MIN....2241
SEARch:TRIGger:SWIRe:TYPE....2241
SEARch:TRIGger:TBTO:CRC:CONDition....1902
SEARch:TRIGger:TBTO:CRC:MAX....1903
SEARch:TRIGger:TBTO:CRC:MIN....1903
SEARch:TRIGger:TBTO:DADDress:CONDition....1903
SEARch:TRIGger:TBTO:DADDress:MAX....1904
SEARch:TRIGger:TBTO:DADDress:MIN....1904
SEARch:TRIGger:TBTO:DATA:DCONdition....1904
SEARch:TRIGger:TBTO:DATA:DMAX....1905
SEARch:TRIGger:TBTO:DATA:DMIN....1905
SEARch:TRIGger:TBTO:DATA:ICONdition....1905
SEARch:TRIGger:TBTO:DATA:IMAX....1906
SEARch:TRIGger:TBTO:DATA:IMIN....1906
SEARch:TRIGger:TBTO:ERRor:CRC....1906
SEARch:TRIGger:TBTO:ERRor:FEC....1907
SEARch:TRIGger:TBTO:ERRor:OOR....1907
SEARch:TRIGger:TBTO:ERRor:ZERO....1907
SEARch:TRIGger:TBTO:LENGth:CONDition....1908
SEARch:TRIGger:TBTO:LENGth:MAX....1908
SEARch:TRIGger:TBTO:LENGth:MIN....1908
SEARch:TRIGger:TBTO:SADDress:CONDition....1909
SEARch:TRIGger:TBTO:SADDress:MAX....1909
SEARch:TRIGger:TBTO:SADDress:MIN....1909
SEARch:TRIGger:TBTO:TYPE....1910
SEARch:TRIGger:TIMeout:ACOPy....1569
SEARch:TRIGger:TIMeout:RANGe....1579
SEARch:TRIGger:TIMeout:TIME....1579
SEARch:TRIGger:TIMeout[:STATe]....1568
SEARch:TRIGger:USB:ACONdition....2142
SEARch:TRIGger:USB:AMAX....2142
SEARch:TRIGger:USB:AMIN....2142
SEARch:TRIGger:USB:BITSterror....2143
SEARch:TRIGger:USB:CRC5error....2143
SEARch:TRIGger:USB:CRC16error....2143
SEARch:TRIGger:USB:DATA....2144
SEARch:TRIGger:USB:DCONdition....2144
SEARch:TRIGger:USB:DPOPerator....2144
SEARch:TRIGger:USB:DPOSition....2145
SEARch:TRIGger:USB:ECONdition....2145
SEARch:TRIGger:USB:EMAX....2146
SEARch:TRIGger:USB:EMIN....2146
SEARch:TRIGger:USB:FCONdition....2146
SEARch:TRIGger:USB:FMAX....2147
SEARch:TRIGger:USB:FMIN....2147
SEARch:TRIGger:USB:GLITcherror....2147
SEARch:TRIGger:USB:HAND....2147
SEARch:TRIGger:USB:PATT....2148
SEARch:TRIGger:USB:PCONdition....2148

SEARch:TRIGger:USB:PIDerror....2149
SEARch:TRIGger:USB:PMAX....2149
SEARch:TRIGger:USB:PMIN....2148
SEARch:TRIGger:USB:SCONdition....2149
SEARch:TRIGger:USB:SDATa....2150
SEARch:TRIGger:USB:SERRor....2150
SEARch:TRIGger:USB:SHANdshake....2151
SEARch:TRIGger:USB:SMAX....2150
SEARch:TRIGger:USB:SMIN....2150
SEARch:TRIGger:USB:SPEC....2151
SEARch:TRIGger:USB:SSOP....2151
SEARch:TRIGger:USB:SSPE....2151
SEARch:TRIGger:USB:STCO....2152
SEARch:TRIGger:USB:STOKen....2152
SEARch:TRIGger:USB:TCONdition....2153
SEARch:TRIGger:USB:TMAX....2153
SEARch:TRIGger:USB:TMIN....2153
SEARch:TRIGger:USB:TOKen....2154
SEARch:TRIGger:USB:WADD....2154
SEARch:TRIGger:USB:WEND....2154
SEARch:TRIGger:USB:WETCheck....2155
SEARch:TRIGger:USB:WFRN....2155
SEARch:TRIGger:USB:WPAY....2155
SEARch:TRIGger:USB:WPID....2156
SEARch:TRIGger:USB:WPOR....2156
SEARch:TRIGger:USB:WSEU....2156
SEARch:TRIGger:USB:WSTC....2156
SEARch:TRIGger:USBPd:BIT....2217
SEARch:TRIGger:USBPd:DMAX....2217
SEARch:TRIGger:USBPd:DMIN....2218
SEARch:TRIGger:USBPd:DOPerator....2218
SEARch:TRIGger:USBPd:ERENable....2215
SEARch:TRIGger:USBPd:ERRor<m>:ENABle....2215
SEARch:TRIGger:USBPd:FIENable....2216
SEARch:TRIGger:USBPd:FRAMe<m>:ENABle....2216
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:BIT....2217
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMAX....2217
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DMIN....2218
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:DOPerator....2218
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:ENABle....2216
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMAX....2219
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IMIN....2219
SEARch:TRIGger:USBPd:FRAMe<m>:FLD<n>:IOPerator....2220
SEARch:TRIGger:USBPd:FRENable....2216
SEARch:TRIGger:USBPd:IMAX....2219
SEARch:TRIGger:USBPd:IMIN....2219
SEARch:TRIGger:USBPd:IOPerator....2220
SEARch:TRIGger:USBThree:BIT....2194
SEARch:TRIGger:USBThree:DMAX....2193
SEARch:TRIGger:USBThree:DMIN....2193

SEARch:TRIGger:USBThree:DOPerator....2192
SEARch:TRIGger:USBThree:ERENable....2190
SEARch:TRIGger:USBThree:ERRor<m>:ENABle....2190
SEARch:TRIGger:USBThree:FIENable....2191
SEARch:TRIGger:USBThree:FRAMe<m>:ENABle....2191
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:BIT....2194
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DMAX....2193
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DMIN....2193
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:DOPerator....2192
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:ENABle....2191
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMAX....2195
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IMIN....2195
SEARch:TRIGger:USBThree:FRAMe<m>:FLD<n>:IOPerator....2194
SEARch:TRIGger:USBThree:FRENable....2191
SEARch:TRIGger:USBThree:IMAX....2195
SEARch:TRIGger:USBThree:IMIN....2195
SEARch:TRIGger:USBThree:IOPerator....2194
SEARch:TRIGger:WIDTh:ACOPy....1570
SEARch:TRIGger:WIDTh:DELTa....1580
SEARch:TRIGger:WIDTh:POLarity....1580
SEARch:TRIGger:WIDTh:RANGe....1580
SEARch:TRIGger:WIDTh:WIDTh....1581
SEARch:TRIGger:WIDTh[:STATe]....1569
SEARch:TRIGger:WINDow:ACOPy....1570
SEARch:TRIGger:WINDow:DELTa....1582
SEARch:TRIGger:WINDow:RANGe....1582
SEARch:TRIGger:WINDow:TIMerange....1583
SEARch:TRIGger:WINDow:WIDTh....1583
SEARch:TRIGger:WINDow[:STATe]....1569
SENSe[:ROSCillator]:EXTernal:FREQuency....1303
SENSe[:ROSCillator]:SOURce....1303
SINGle....1235
STATus:OPERation:CONDition?....2463
STATus:OPERation:ENABle....2464
STATus:OPERation[:EVENt]?....2463
STATus:QUEStionable:ADCState:CONDition?....2467
STATus:QUEStionable:ADCState:ENABle....2467
STATus:QUEStionable:ADCState:NTRansition....2468
STATus:QUEStionable:ADCState:PTRansition....2468
STATus:QUEStionable:ADCState[:EVENt]?....2467
STATus:QUEStionable:COVerload:CONDition?....2466
STATus:QUEStionable:COVerload:ENABle....2467
STATus:QUEStionable:COVerload:NTRansition....2468
STATus:QUEStionable:COVerload:PTRansition....2468
STATus:QUEStionable:COVerload[:EVENt]?....2467
STATus:QUEStionable:DEEMbedding:CONDition?....2467
STATus:QUEStionable:DEEMbedding:ENABle....2467
STATus:QUEStionable:DEEMbedding:NTRansition....2468
STATus:QUEStionable:DEEMbedding:PTRansition....2468
STATus:QUEStionable:DEEMbedding[:EVENt]?....2467

STATus:QUEStionable:EFRontend:CONDition? 2467
STATus:QUEStionable:EFRontend:ENABle 2467
STATus:QUEStionable:EFRontend:NTRansition 2468
STATus:QUEStionable:EFRontend:PTRansition 2468
STATus:QUEStionable:EFRontend[:EVENt]? 2468
STATus:QUEStionable:FREQuency:CONDition? 2467
STATus:QUEStionable:FREQuency:ENABle 2467
STATus:QUEStionable:FREQuency:EVENt? 2467
STATus:QUEStionable:FREQuency:NTRansition 2468
STATus:QUEStionable:FREQuency:PTRansition 2468
STATus:QUEStionable:LAMPlitude:CONDition? 2467
STATus:QUEStionable:LAMPlitude:ENABle 2467
STATus:QUEStionable:LAMPlitude:NTRansition 2468
STATus:QUEStionable:LAMPlitude:PTRansition 2468
STATus:QUEStionable:LAMPlitude[:EVENt]? 2467
STATus:QUEStionable:LIMit:CONDition? 2467
STATus:QUEStionable:LIMit:ENABle 2467
STATus:QUEStionable:LIMit:NTRansition 2468
STATus:QUEStionable:LIMit:PTRansition 2468
STATus:QUEStionable:LIMit[:EVENt]? 2467
STATus:QUEStionable:MARGin:CONDition? 2467
STATus:QUEStionable:MARGin:ENABle 2467
STATus:QUEStionable:MARGin:NTRansition 2468
STATus:QUEStionable:MARGin:PTRansition 2468
STATus:QUEStionable:MARGin[:EVENt]? 2467
STATus:QUEStionable:MASK:CONDition? 2467
STATus:QUEStionable:MASK:ENABle 2467
STATus:QUEStionable:MASK:NTRansition 2468
STATus:QUEStionable:MASK:PTRansition 2468
STATus:QUEStionable:MASK[:EVENt]? 2468
STATus:QUEStionable:PLL:CONDition? 2467
STATus:QUEStionable:PLL:ENABle 2467
STATus:QUEStionable:PLL:NTRansition 2468
STATus:QUEStionable:PLL:PTRansition 2468
STATus:QUEStionable:PLL[:EVENt]? 2467
STATus:QUEStionable:TEMPerature:CONDition? 2466
STATus:QUEStionable:TEMPerature:ENABle 2467
STATus:QUEStionable:TEMPerature:NTRansition 2468
STATus:QUEStionable:TEMPerature:PTRansition 2468
STATus:QUEStionable:TEMPerature[:EVENt]? 2467
STATus:QUEStionable:ZVCoverload:CONDition? 2467
STATus:QUEStionable:ZVCoverload:ENABle 2467
STATus:QUEStionable:ZVCoverload:NTRansition 2468
STATus:QUEStionable:ZVCoverload:PTRansition 2468
STATus:QUEStionable:ZVCoverload[:EVENt]? 2468
STOP 1235
SWTRigger:HISTory 1374
SYSTem:DATE 1205
SYSTem:DEVice:ID? 1206
SYSTem:DFPRint 1233

SYSTem:DISPlay:MESSage:STATe ..... 1204
SYSTem:DISPlay:MESSage[:TEXT] ..... 1204
SYSTem:DISPlay:UPDate ..... 1203
SYSTem:KLOCk ..... 1203
SYSTem:LANGuage ..... 1205
SYSTem:PRESet ..... 1205
SYSTem:TIME? ..... 1206
TDRT:BW ..... 2363
TDRT:CALibration:OPEN:NAME ..... 2360
TDRT:CALibration:OPEN[:EXECute] ..... 2360
TDRT:CALibration:SAVE:NAME ..... 2362
TDRT:CALibration:SAVE[:EXECute] ..... 2362
TDRT:CALibration:USECaldata ..... 2363
TDRT:CONFig ..... 2360
TDRT:DEFault ..... 2360
TDRT:DOMain ..... 2364
TDRT:PERMittivity ..... 2364
TDRT:PULSe:INVout ..... 2361
TDRT:PULSe:INVReference ..... 2361
TDRT:PULSe:OUT ..... 2361
TDRT:PULSe:REFerence ..... 2361
TDRT:SIGNal<m>:DATA:HEADer? ..... 2365
TDRT:SIGNal<m>:DATA:VALues? ..... 2366
TDRT:SIGNal<m>:OFFSet ..... 2365
TDRT:SIGNal<m>:POSition ..... 2365
TDRT:SIGNal<m>:SCALe ..... 2364
TDRT:SIGNal<m>STATe ..... 2364
TDRT:SLENgth ..... 2362
TDRT:STATe ..... 2363
TDRT:TYPE ..... 2360
TIMebase:DIVisions? ..... 1237
TIMebase:HORizontal:POSition ..... 1237
TIMebase:RACTime? ..... 1532
TIMebase:RANGe ..... 1237
TIMebase:REFerence ..... 1237
TIMebase:ROLL:ENABle ..... 1241
TIMebase:ROLL:MTIMe ..... 1242
TIMebase:ROLL:STATe? ..... 1241
TIMebase:SCALe ..... 1236
TRACe:REMote:MODE:FILE:DEXecution:DURation ..... 2472
TRACe:REMote:MODE:FILE:ENABle ..... 2471
TRACe:REMote:MODE:FILE:FILTer ..... 2471
TRACe:REMote:MODE:FILE:FORMat ..... 2470
TRACe:REMote:MODE:FILE:FUNCtions ..... 2472
TRACe:REMote:MODE:FILE:NAME ..... 2470
TRACe:REMote:MODE:FILE:PARSer ..... 2472
TRACe:REMote:MODE:FILE:RPC ..... 2472
TRACe:REMote:MODE:FILE:SIZE ..... 2470
TRACe:REMote:MODE:FILE:STARtmode ..... 2470
TRACe:REMote:MODE:FILE:STOPmode ..... 2471

TRIGger<m>:ANEDge:COUPling....1375
TRIGger<m>:ANEDge:CUToff:HIGHpass....1376
TRIGger<m>:ANEDge:CUToff:LOWPass....1376
TRIGger<m>:ANEDge:FILTer....1375
TRIGger<m>:ANEDge:GND....1376
TRIGger<m>:ANEDge:SLOPe....1377
TRIGger<m>:ARINc:DATA:CONDition....1814
TRIGger<m>:ARINc:DATA:MAX....1815
TRIGger<m>:ARINc:DATA:MIN....1815
TRIGger<m>:ARINc:ERRor:CODing....1815
TRIGger<m>:ARINc:ERRor:PARity....1815
TRIGger<m>:ARINc:LABel:CONDition....1816
TRIGger<m>:ARINc:LABel:MAX....1816
TRIGger<m>:ARINc:LABel:MIN....1816
TRIGger<m>:ARINc:MAXGap:BITS....1817
TRIGger<m>:ARINc:MAXGap:SELect....1817
TRIGger<m>:ARINc:MINGap:BITS....1817
TRIGger<m>:ARINc:MINGap:SELect....1816
TRIGger<m>:ARINc:SDI....1817
TRIGger<m>:ARINc:SSM....1817
TRIGger<m>:ARINc:TYPE....1814
TRIGger<m>:CAN:ACKerror....1721
TRIGger<m>:CAN:BITSterror....1721
TRIGger<m>:CAN:BORDer....1719
TRIGger<m>:CAN:CRCerror....1721
TRIGger<m>:CAN:DCONdition....1718
TRIGger<m>:CAN:DLC....1719
TRIGger<m>:CAN:DLCCondition....1719
TRIGger<m>:CAN:DMAX....1718
TRIGger<m>:CAN:DMIN....1718
TRIGger<m>:CAN:FDATa:BRS....1717
TRIGger<m>:CAN:FDATa:DPOPerator....1720
TRIGger<m>:CAN:FDATa:DPOSition....1720
TRIGger<m>:CAN:FDATa:DPTO....1721
TRIGger<m>:CAN:FDATa:ESI....1717
TRIGger<m>:CAN:FDATa:FDF....1717
TRIGger<m>:CAN:FDATa:SCERror....1722
TRIGger<m>:CAN:FDATa:STANdard....1715
TRIGger<m>:CAN:FORMerror....1722
TRIGger<m>:CAN:FTYPe....1715
TRIGger<m>:CAN:ICONdition....1716
TRIGger<m>:CAN:IMAX....1717
TRIGger<m>:CAN:IMIN....1716
TRIGger<m>:CAN:ITYPe....1716
TRIGger<m>:CAN:NDBYtes?....1720
TRIGger<m>:CAN:SYMBolic:DMAX....1748
TRIGger<m>:CAN:SYMBolic:DMIN....1748
TRIGger<m>:CAN:SYMBolic:MSGValue....1747
TRIGger<m>:CAN:SYMBolic:SGEValue....1748
TRIGger<m>:CAN:SYMBolic:SIGValue....1748

TRIGger<m>:CAN:SYMBolic:TSIGnals....1747
TRIGger<m>:CAN:TYPE....1714
TRIGger<m>:CDR:EYE:ALGorithm....1348
TRIGger<m>:CDR:EYE:BITRate....1348
TRIGger<m>:CDR:EYE:BREStimate....1348
TRIGger<m>:CDR:EYE:BREStimate:CONTinuous....1349
TRIGger<m>:CDR:EYE:BWIDth....1349
TRIGger<m>:CDR:EYE:LEVel....1349
TRIGger<m>:CDR:EYE:RELBwidth....1350
TRIGger<m>:CDR:EYE:SAMTime....1350
TRIGger<m>:CDR:EYE:SERStandard....1350
TRIGger<m>:CDR:EYE:SOURce?....1351
TRIGger<m>:CMSB:ADVanced:BIT....2063
TRIGger<m>:CMSB:ADVanced:DMAX....2064
TRIGger<m>:CMSB:ADVanced:DMIN....2064
TRIGger<m>:CMSB:ADVanced:DOPerator....2064
TRIGger<m>:CMSB:ADVanced:ERENable....2065
TRIGger<m>:CMSB:ADVanced:ERRor<n>:ENABle....2065
TRIGger<m>:CMSB:ADVanced:FIENable....2063
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:ENABle....2062
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:BIT....2063
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:DMAX....2064
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:DMIN....2064
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:DOPerator....2064
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:ENABle....2063
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:IMAX....2065
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:IMIN....2066
TRIGger<m>:CMSB:ADVanced:FRAMe<n>:FLD<o>:IOPerator....2066
TRIGger<m>:CMSB:ADVanced:FRENable....2062
TRIGger<m>:CMSB:ADVanced:IMAX....2065
TRIGger<m>:CMSB:ADVanced:IMIN....2066
TRIGger<m>:CMSB:ADVanced:IOPerator....2066
TRIGger<m>:CMSB:ICONdition....2061
TRIGger<m>:CMSB:IMAX....2062
TRIGger<m>:CMSB:IMIN....2061
TRIGger<m>:CMSB:NRZ:WRDLength....2062
TRIGger<m>:CMSB:PATTern....2061
TRIGger<m>:CMSB:TYPE....2060
TRIGger<m>:DPHY:DATA:CONDition....1956
TRIGger<m>:DPHY:DATA:MAX....1957
TRIGger<m>:DPHY:DATA:MIN....1957
TRIGger<m>:DPHY:DIDX:CONDition....1957
TRIGger<m>:DPHY:DIDX:MAX....1957
TRIGger<m>:DPHY:DIDX:MIN....1958
TRIGger<m>:DPHY:DTYPe:CONDition....1958
TRIGger<m>:DPHY:DTYPe:MAX....1958
TRIGger<m>:DPHY:DTYPe:MIN....1959
TRIGger<m>:DPHY:ESCMode:CONDition....1959
TRIGger<m>:DPHY:ESCMode:MAX....1959
TRIGger<m>:DPHY:ESCMode:MIN....1959

TRIGger<m>:DPHY:ESDTa:CONDition....1960
TRIGger<m>:DPHY:ESDTa:MAX....1960
TRIGger<m>:DPHY:ESDTa:MIN....1960
TRIGger<m>:DPHY:ESINdex:CONDition....1961
TRIGger<m>:DPHY:ESINdex:MAX....1961
TRIGger<m>:DPHY:ESINdex:MIN....1961
TRIGger<m>:DPHY:HSVC....1961
TRIGger<m>:DPHY:TYPE....1956
TRIGger<m>:DPHY:WCOunt:CONDition....1962
TRIGger<m>:DPHY:WCOunt:MAX....1962
TRIGger<m>:DPHY:WCOunt:MIN....1962
TRIGger<m>:EBTB:DISParityerr....2094
TRIGger<m>:EBTB:DX....2094
TRIGger<m>:EBTB:DY....2094
TRIGger<m>:EBTB:EBPA....2095
TRIGger<m>:EBTB:GLITcherror....2095
TRIGger<m>:EBTB:SSTYpe....2095
TRIGger<m>:EBTB:STYPe....2096
TRIGger<m>:EBTB:SYME....2095
TRIGger<m>:EBTB:SYMType....2095
TRIGger<m>:EBTB:TBPA....2096
TRIGger<m>:EBTB:UNK....2096
TRIGger<m>:ECOupling....1368
TRIGger<m>:EDGE:SLOPe....1308
TRIGger<m>:ETHernet:ERRor:CRC....1832
TRIGger<m>:ETHernet:ERRor:LENGth....1832
TRIGger<m>:ETHernet:ERRor:PREamble....1832
TRIGger<m>:ETHernet:FRAMe:CCONdition....1832
TRIGger<m>:ETHernet:FRAMe:CMAX....1833
TRIGger<m>:ETHernet:FRAMe:CMIN....1833
TRIGger<m>:ETHernet:FRAMe:DCONdition....1833
TRIGger<m>:ETHernet:FRAMe:DMAX....1834
TRIGger<m>:ETHernet:FRAMe:DMIN....1834
TRIGger<m>:ETHernet:FRAMe:SCONdition....1834
TRIGger<m>:ETHernet:FRAMe:SMAX....1835
TRIGger<m>:ETHernet:FRAMe:SMIN....1835
TRIGger<m>:ETHernet:FRAMe:TCONdition....1835
TRIGger<m>:ETHernet:FRAMe:TMAX....1836
TRIGger<m>:ETHernet:FRAMe:TMIN....1836
TRIGger<m>:ETHernet:PATTern....1831
TRIGger<m>:ETHernet:TYPE....1831
TRIGger<m>:EVENt:BEEP....1372
TRIGger<m>:EVENt:PRINt....1372
TRIGger<m>:EVENt:RUNexec....1373
TRIGger<m>:EVENt:WFMSave....1373
TRIGger<m>:EVENt[:EVENt]....1367
TRIGger<m>:EXTern:OVERload....1377
TRIGger<m>:FINDlevel....1307
TRIGger<m>:FORCe....1371
TRIGger<m>:GLITch:POLarity....1308

TRIGger<m>:GLITch:RANGe....1308
TRIGger<m>:GLITch:WIDTh....1309
TRIGger<m>:HBTO:CRC:CONDition....1860
TRIGger<m>:HBTO:CRC:MAX....1861
TRIGger<m>:HBTO:CRC:MIN....1861
TRIGger<m>:HBTO:DADDress:CONDition....1857
TRIGger<m>:HBTO:DADDress:MAX....1858
TRIGger<m>:HBTO:DADDress:MIN....1858
TRIGger<m>:HBTO:DATA:DCONdition....1861
TRIGger<m>:HBTO:DATA:DMAX....1862
TRIGger<m>:HBTO:DATA:DMIN....1862
TRIGger<m>:HBTO:DATA:ICONdition....1862
TRIGger<m>:HBTO:DATA:IMAX....1863
TRIGger<m>:HBTO:DATA:IMIN....1863
TRIGger<m>:HBTO:ERRor:CRC....1863
TRIGger<m>:HBTO:ERRor:PREamble....1863
TRIGger<m>:HBTO:ERRor:SFD....1864
TRIGger<m>:HBTO:LENGth:CONDition....1859
TRIGger<m>:HBTO:LENGth:MAX....1860
TRIGger<m>:HBTO:LENGth:MIN....1860
TRIGger<m>:HBTO:SADDress:CONDition....1858
TRIGger<m>:HBTO:SADDress:MAX....1859
TRIGger<m>:HBTO:SADDress:MIN....1859
TRIGger<m>:HBTO:TYPE....1857
TRIGger<m>:HOLDoff:AUTotime?....1363
TRIGger<m>:HOLDoff:EVENts....1362
TRIGger<m>:HOLDoff:MAX....1363
TRIGger<m>:HOLDoff:MIN....1362
TRIGger<m>:HOLDoff:MODE....1360
TRIGger<m>:HOLDoff:SCALing....1364
TRIGger<m>:HOLDoff:TIME....1361
TRIGger<m>:I2C:ACCess....1647
TRIGger<m>:I2C:ACONdition....1649
TRIGger<m>:I2C:ADDRess....1649
TRIGger<m>:I2C:ADDTo....1649
TRIGger<m>:I2C:ADNack....1647
TRIGger<m>:I2C:ADOR<n>:ADRType....1650
TRIGger<m>:I2C:ADOR<n>:ENABle....1649
TRIGger<m>:I2C:ADOR<n>:RWBit....1650
TRIGger<m>:I2C:ADOR<n>[:VALue]....1650
TRIGger<m>:I2C:AMODe....1648
TRIGger<m>:I2C:DCONdition....1651
TRIGger<m>:I2C:DMAX....1652
TRIGger<m>:I2C:DMIN....1652
TRIGger<m>:I2C:DPOPerator....1651
TRIGger<m>:I2C:DPOSition....1651
TRIGger<m>:I2C:DPTO....1651
TRIGger<m>:I2C:DRNack....1648
TRIGger<m>:I2C:DWNack....1648
TRIGger<m>:I2C:TYPE....1647

TRIGger<m>:INTerval:DELTa ... 1318
TRIGger<m>:INTerval:RANGe ... 1317
TRIGger<m>:INTerval:SLOPe ... 1317
TRIGger<m>:INTerval:WIDTh ... 1318
TRIGger<m>:LEVel<n>:NOISe:ABSolute ... 1365
TRIGger<m>:LEVel<n>:NOISe:MODE ... 1365
TRIGger<m>:LEVel<n>:NOISe:PERDivision ... 1366
TRIGger<m>:LEVel<n>:NOISe:RELative ... 1366
TRIGger<m>:LEVel<n>:NOISe[:STATe] ... 1364
TRIGger<m>:LEVel<n>:RUNT:LOWer ... 1312
TRIGger<m>:LEVel<n>:RUNT:UPPer ... 1312
TRIGger<m>:LEVel<n>:SLEW:LOWer ... 1319
TRIGger<m>:LEVel<n>:SLEW:UPPer ... 1319
TRIGger<m>:LEVel<n>:WINDow:LOWer ... 1314
TRIGger<m>:LEVel<n>:WINDow:UPPer ... 1314
TRIGger<m>:LEVel<n>[:VALue] ... 1306
TRIGger<m>:LEVel<n>[:VALue] ... 1329
TRIGger<m>:LIN:BORDer ... 1758
TRIGger<m>:LIN:CHKSerror ... 1760
TRIGger<m>:LIN:CRCDatalen ... 1761
TRIGger<m>:LIN:DCONdition ... 1757
TRIGger<m>:LIN:DLECondition ... 1758
TRIGger<m>:LIN:DLENgth ... 1759
TRIGger<m>:LIN:DMAX ... 1757
TRIGger<m>:LIN:DMIN ... 1757
TRIGger<m>:LIN:ERRPattern ... 1760
TRIGger<m>:LIN:ICONdition ... 1756
TRIGger<m>:LIN:IDOR<n>:ENABle ... 1759
TRIGger<m>:LIN:IDOR<n>[:VALue] ... 1759
TRIGger<m>:LIN:IMAX ... 1757
TRIGger<m>:LIN:IMIN ... 1756
TRIGger<m>:LIN:IPERror ... 1760
TRIGger<m>:LIN:STANdard ... 1761
TRIGger<m>:LIN:SYERror ... 1759
TRIGger<m>:LIN:TYPE ... 1755
TRIGger<m>:MDIO:DATA ... 2109
TRIGger<m>:MDIO:FRAMetype ... 2108
TRIGger<m>:MDIO:PHYS ... 2109
TRIGger<m>:MDIO:REGI ... 2109
TRIGger<m>:MDIO:ST ... 2108
TRIGger<m>:MDIO:TYPE ... 2107
TRIGger<m>:MILStd:CDST:ICONdition ... 1791
TRIGger<m>:MILStd:CDST:IMAX ... 1791
TRIGger<m>:MILStd:CDST:IMIN ... 1791
TRIGger<m>:MILStd:CDST:RCONdition ... 1787
TRIGger<m>:MILStd:CDST:RMAX ... 1788
TRIGger<m>:MILStd:CDST:RMIN ... 1788
TRIGger<m>:MILStd:CMD:CCONdition ... 1788
TRIGger<m>:MILStd:CMD:CMAX ... 1789
TRIGger<m>:MILStd:CMD:CMIN ... 1789

TRIGger<m>:MILStd:CMD:RCONdition ... 1787
TRIGger<m>:MILStd:CMD:RMAX ... 1788
TRIGger<m>:MILStd:CMD:RMIN ... 1788
TRIGger<m>:MILStd:CMD:SCONdition ... 1789
TRIGger<m>:MILStd:CMD:SMAX ... 1790
TRIGger<m>:MILStd:CMD:SMIN ... 1790
TRIGger<m>:MILStd:CMD:TR ... 1790
TRIGger<m>:MILStd:DATA:DCONdition ... 1791
TRIGger<m>:MILStd:DATA:DMAX ... 1792
TRIGger<m>:MILStd:DATA:DMIN ... 1792
TRIGger<m>:MILStd:DATA:ICONdition ... 1792
TRIGger<m>:MILStd:DATA:IMAX ... 1793
TRIGger<m>:MILStd:DATA:IMIN ... 1793
TRIGger<m>:MILStd:DATA:RCONdition ... 1787
TRIGger<m>:MILStd:DATA:RMAX ... 1788
TRIGger<m>:MILStd:DATA:RMIN ... 1788
TRIGger<m>:MILStd:ERRor:MANChester ... 1793
TRIGger<m>:MILStd:ERRor:PARity ... 1793
TRIGger<m>:MILStd:ERRor:SYNC ... 1794
TRIGger<m>:MILStd:MAXResponse:BITS ... 1794
TRIGger<m>:MILStd:MAXResponse:SELect ... 1794
TRIGger<m>:MILStd:MINGap:BITS ... 1794
TRIGger<m>:MILStd:MINGap:SELect ... 1795
TRIGger<m>:MILStd:STATus:BCReceived ... 1795
TRIGger<m>:MILStd:STATus:BUSY ... 1795
TRIGger<m>:MILStd:STATus:DBCaccept ... 1795
TRIGger<m>:MILStd:STATus:INSTrument ... 1795
TRIGger<m>:MILStd:STATus:MERRor ... 1796
TRIGger<m>:MILStd:STATus:SREQuest ... 1796
TRIGger<m>:MILStd:STATus:SUBSystem ... 1796
TRIGger<m>:MILStd:STATus:TERMinal ... 1796
TRIGger<m>:MILStd:TPSPecifier ... 1797
TRIGger<m>:MILStd:TYPE ... 1787
TRIGger<m>:MODE ... 1370
TRIGger<m>:MPHY:CRC:CONDition ... 1996
TRIGger<m>:MPHY:CRC:MAX ... 1996
TRIGger<m>:MPHY:CRC:MIN ... 1996
TRIGger<m>:MPHY:CREDit:CONDition ... 1997
TRIGger<m>:MPHY:CREDit:MAX ... 1998
TRIGger<m>:MPHY:CREDit:MIN ... 1998
TRIGger<m>:MPHY:CREQ ... 1997
TRIGger<m>:MPHY:DATA:DCON ... 1993
TRIGger<m>:MPHY:DATA:DMAX ... 1993
TRIGger<m>:MPHY:DATA:DMIN ... 1993
TRIGger<m>:MPHY:DATA:ICONdition ... 1994
TRIGger<m>:MPHY:DATA:IMAX ... 1994
TRIGger<m>:MPHY:DATA:IMIN ... 1994
TRIGger<m>:MPHY:ERRor:LCMD ... 2001
TRIGger<m>:MPHY:ERRor:LENGth ... 2001
TRIGger<m>:MPHY:ERRor:REServed ... 2001

TRIGger<m>:MPHY:ERRor:SYMBol....2001
TRIGger<m>:MPHY:ERRor:UNKNown....2002
TRIGger<m>:MPHY:FSNumber:CONDition....1995
TRIGger<m>:MPHY:FSNumber:MAX....1995
TRIGger<m>:MPHY:FSNumber:MIN....1995
TRIGger<m>:MPHY:LCCType....1987
TRIGger<m>:MPHY:LWFour:CONDition....1991
TRIGger<m>:MPHY:LWFour:MAX....1992
TRIGger<m>:MPHY:LWFour:MIN....1992
TRIGger<m>:MPHY:LWONe:CONDition....1988
TRIGger<m>:MPHY:LWONe:MAX....1989
TRIGger<m>:MPHY:LWONe:MIN....1989
TRIGger<m>:MPHY:LWTHree:CONDition....1990
TRIGger<m>:MPHY:LWTHree:MAX....1991
TRIGger<m>:MPHY:LWTHree:MIN....1991
TRIGger<m>:MPHY:LWTWo:CONDition....1989
TRIGger<m>:MPHY:LWTWo:MAX....1990
TRIGger<m>:MPHY:LWTWo:MIN....1990
TRIGger<m>:MPHY:PBEGin:CONDition....1998
TRIGger<m>:MPHY:PBEGin:MAX....1999
TRIGger<m>:MPHY:PBEGin:MIN....1999
TRIGger<m>:MPHY:PFID:CONDition....1999
TRIGger<m>:MPHY:PFID:MAX....2000
TRIGger<m>:MPHY:PFID:MIN....2000
TRIGger<m>:MPHY:PRPLength....2000
TRIGger<m>:MPHY:RREQ....1997
TRIGger<m>:MPHY:SSIC:BIT....2033
TRIGger<m>:MPHY:SSIC:DMAX....2033
TRIGger<m>:MPHY:SSIC:DMIN....2034
TRIGger<m>:MPHY:SSIC:DOPerator....2034
TRIGger<m>:MPHY:SSIC:ERENable....2032
TRIGger<m>:MPHY:SSIC:ERRor<n>:ENABle....2032
TRIGger<m>:MPHY:SSIC:FIENable....2033
TRIGger<m>:MPHY:SSIC:FRAMe<n>:ENABle....2032
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:BIT....2033
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DMAX....2033
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DMIN....2034
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:DOPerator....2034
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:ENABle....2033
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IMAX....2035
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IMIN....2035
TRIGger<m>:MPHY:SSIC:FRAMe<n>:FLD<o>:IOPerator....2036
TRIGger<m>:MPHY:SSIC:FRENable....2032
TRIGger<m>:MPHY:SSIC:IMAX....2035
TRIGger<m>:MPHY:SSIC:IMIN....2035
TRIGger<m>:MPHY:SSIC:IOPerator....2036
TRIGger<m>:MPHY:TC....1992
TRIGger<m>:MPHY:TYPE....1987
TRIGger<m>:OFFSet:LIMited....1238
TRIGger<m>:OUT:ACTion....1374

TRIGger<m>:OUT:DELay....1372
TRIGger<m>:OUT:PLENgth....1371
TRIGger<m>:OUT:POLarity....1371
TRIGger<m>:PARallel:EDGE:EXPRession[:DEFine]....2323
TRIGger<m>:PARallel:EDGE:SLOPe....2324
TRIGger<m>:PARallel:PATTern:BIT<0..15>....2328
TRIGger<m>:PARallel:PATTern:EXPRession[:DEFine]....2323
TRIGger<m>:PARallel:PATTern:MODE....2329
TRIGger<m>:PARallel:PATTern:TIMeout:MODE....2329
TRIGger<m>:PARallel:PATTern:TIMeout[:TIME]....2329
TRIGger<m>:PARallel:PATTern:WIDTh:DELTa....2331
TRIGger<m>:PARallel:PATTern:WIDTh:RANGe....2330
TRIGger<m>:PARallel:PATTern:WIDTh[:WIDTh]....2330
TRIGger<m>:PARallel:SETHold:CSOurce:EDGE....2327
TRIGger<m>:PARallel:SETHold:CSOurce[:VALue]....2323
TRIGger<m>:PARallel:SETHold:HTIMe....2327
TRIGger<m>:PARallel:SETHold:STIMe....2327
TRIGger<m>:PARallel:SPATtern:CSOurce:EDGE....2331
TRIGger<m>:PARallel:SPATtern:CSOurce[:VALue]....2323
TRIGger<m>:PARallel:SPATtern:EXPRession[:DEFine]....2323
TRIGger<m>:PARallel:SPATtern:PATTern....2331
TRIGger<m>:PARallel:STATe:BIT<0..15>....2328
TRIGger<m>:PARallel:STATe:CSOurce:EDGE....2328
TRIGger<m>:PARallel:STATe:CSOurce:VALue....2323
TRIGger<m>:PARallel:STATe:EXPRession[:DEFine]....2323
TRIGger<m>:PARallel:TIMeout:EXPRession[:DEFine]....2323
TRIGger<m>:PARallel:TIMeout:RANGe....2326
TRIGger<m>:PARallel:TIMeout:TIME....2326
TRIGger<m>:PARallel:TYPE....2323
TRIGger<m>:PARallel:WIDTh:DELTa....2325
TRIGger<m>:PARallel:WIDTh:EXPRession[:DEFine]....2323
TRIGger<m>:PARallel:WIDTh:POLarity....2324
TRIGger<m>:PARallel:WIDTh:RANGe....2325
TRIGger<m>:PARallel:WIDTh:WIDTh....2325
TRIGger<m>:PATTern:MODE....1324
TRIGger<m>:PATTern:TIMeout:MODE....1325
TRIGger<m>:PATTern:TIMeout[:TIME]....1325
TRIGger<m>:PATTern:WIDTh:DELTa....1327
TRIGger<m>:PATTern:WIDTh:RANGe....1326
TRIGger<m>:PATTern:WIDTh[:WIDTh]....1326
TRIGger<m>:PCIE:DLLP:FCTL:CRET....2253
TRIGger<m>:PCIE:DLLP:FCTL:VCID....2253
TRIGger<m>:PCIE:DLLP:MULT:AUTH....2253
TRIGger<m>:PCIE:DLLP:MULT:CRET....2254
TRIGger<m>:PCIE:DLLP:MULT:DVPT....2254
TRIGger<m>:PCIE:DLLP:MULT:HABS....2254
TRIGger<m>:PCIE:DLLP:MULT:MIXT....2254
TRIGger<m>:PCIE:DLLP:MULT:PHAS....2254
TRIGger<m>:PCIE:DLLP:MULT:RESA....2255
TRIGger<m>:PCIE:DLLP:MULT:TLPT....2255

TRIGger<m>:PCIE:DLLP:MULT:TYPE....2255
TRIGger<m>:PCIE:DLLP:MULT:VHFC....2255
TRIGger<m>:PCIE:DLLP:MULT:VHGR....2256
TRIGger<m>:PCIE:DLLP:MULT:VLNR....2256
TRIGger<m>:PCIE:DLLP:POWM....2256
TRIGger<m>:PCIE:DLLP:SEQ....2256
TRIGger<m>:PCIE:DLLP:TYPE....2256
TRIGger<m>:PCIE:DLLP:VPAT....2257
TRIGger<m>:PCIE:ERRC:CRC....2257
TRIGger<m>:PCIE:ERRC:DISP....2257
TRIGger<m>:PCIE:ERRC:ECRC....2257
TRIGger<m>:PCIE:ERRC:INVP....2257
TRIGger<m>:PCIE:ERRC:LCRC....2258
TRIGger<m>:PCIE:ERRC:LENGth....2258
TRIGger<m>:PCIE:ERRC:MALFormed....2258
TRIGger<m>:PCIE:ERRC:MXPFix....2258
TRIGger<m>:PCIE:ERRC:PFORder....2258
TRIGger<m>:PCIE:ERRC:SYCBit....2258
TRIGger<m>:PCIE:ERRC:UKNP....2259
TRIGger<m>:PCIE:ERRC:VOORange....2259
TRIGger<m>:PCIE:OSET:TYPE....2259
TRIGger<m>:PCIE:TLP:ADRT....2259
TRIGger<m>:PCIE:TLP:AT....2262
TRIGger<m>:PCIE:TLP:CFGT....2260
TRIGger<m>:PCIE:TLP:CPID....2260
TRIGger<m>:PCIE:TLP:CPLS....2260
TRIGger<m>:PCIE:TLP:DEID....2260
TRIGger<m>:PCIE:TLP:MERW....2260
TRIGger<m>:PCIE:TLP:MSGC....2261
TRIGger<m>:PCIE:TLP:MSGR....2261
TRIGger<m>:PCIE:TLP:ORDE....2261
TRIGger<m>:PCIE:TLP:REID....2261
TRIGger<m>:PCIE:TLP:SNOO....2262
TRIGger<m>:PCIE:TLP:SNUM....2262
TRIGger<m>:PCIE:TLP:TCHN....2262
TRIGger<m>:PCIE:TLP:TYPE....2262
TRIGger<m>:PCIE:TRGLevel....2253
TRIGger<m>:PCIE:TYPE....2263
TRIGger<m>:POWerline:SLOPe....1356
TRIGger<m>:QUALify<n>:A:LOGic....1359
TRIGger<m>:QUALify<n>:A[:ENABle]....1358
TRIGger<m>:QUALify<n>:AB:LOGic....1359
TRIGger<m>:QUALify<n>:ABCD:LOGic....1359
TRIGger<m>:QUALify<n>:B:LOGic....1359
TRIGger<m>:QUALify<n>:B[:ENABle]....1358
TRIGger<m>:QUALify<n>:C:LOGic....1359
TRIGger<m>:QUALify<n>:C[:ENABle]....1358
TRIGger<m>:QUALify<n>:CD:LOGic....1359
TRIGger<m>:QUALify<n>:D:LOGic....1359
TRIGger<m>:QUALify<n>:D[:ENABle]....1358

TRIGger<m>:QUALify<n>:STATe....1358
TRIGger<m>:RFFE:ADDRess:CONDition....1925
TRIGger<m>:RFFE:ADDRess:MAX....1925
TRIGger<m>:RFFE:ADDRess:MIN....1925
TRIGger<m>:RFFE:BCOunt:CONDition....1925
TRIGger<m>:RFFE:BCOunt:MAX....1926
TRIGger<m>:RFFE:BCOunt:MIN....1926
TRIGger<m>:RFFE:DATA:DCON....1926
TRIGger<m>:RFFE:DATA:DMAX....1927
TRIGger<m>:RFFE:DATA:DMIN....1927
TRIGger<m>:RFFE:DATA:ICONdition....1927
TRIGger<m>:RFFE:DATA:IMAX....1928
TRIGger<m>:RFFE:DATA:IMIN....1928
TRIGger<m>:RFFE:ERRor:BP....1930
TRIGger<m>:RFFE:ERRor:GAP....1931
TRIGger<m>:RFFE:ERRor:LENGth....1930
TRIGger<m>:RFFE:ERRor:NOResponse....1930
TRIGger<m>:RFFE:ERRor:PARity....1930
TRIGger<m>:RFFE:ERRor:SSC....1931
TRIGger<m>:RFFE:ERRor:USEQuence....1931
TRIGger<m>:RFFE:ERRor:VERSion....1931
TRIGger<m>:RFFE:INTerrupt....1930
TRIGger<m>:RFFE:INTerrupt....1931
TRIGger<m>:RFFE:MASK:CONDition....1929
TRIGger<m>:RFFE:MASK:MAX....1929
TRIGger<m>:RFFE:MASK:MIN....1929
TRIGger<m>:RFFE:SADD:CONDition....1928
TRIGger<m>:RFFE:SADD:MAX....1929
TRIGger<m>:RFFE:SADD:MIN....1928
TRIGger<m>:RFFE:TYPE....1922
TRIGger<m>:RFReject?....1299
TRIGger<m>:RUNT:DELTa....1313
TRIGger<m>:RUNT:POLarity....1311
TRIGger<m>:RUNT:RANGe....1312
TRIGger<m>:RUNT:WIDTh....1313
TRIGger<m>:SCOupling....1323
TRIGger<m>:SEQuence:COUNt....1369
TRIGger<m>:SEQuence:DELay....1368
TRIGger<m>:SEQuence:RESet:EVENt....1369
TRIGger<m>:SEQuence:RESet:TIMeout:TIME....1370
TRIGger<m>:SEQuence:RESet:TIMeout[:ENABle]....1369
TRIGger<m>:SEQuence:TYPE....1367
TRIGger<m>:SERPattern:BITPattern:APATtern....1333
TRIGger<m>:SERPattern:BITPattern:BPATtern....1333
TRIGger<m>:SERPattern:BITPattern:PTALength....1334
TRIGger<m>:SERPattern:BITPattern:PTBLength....1334
TRIGger<m>:SERPattern:COMPlex:BORDer....1335
TRIGger<m>:SERPattern:COMPlex:CLOGic....1335
TRIGger<m>:SERPattern:COMPlex:IDLRange....1335
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:BOFFset....1337

TRIGger<m>:SERPattern:COMPlex:PATTern<n>:ENABle ... 1337
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:FRPTtern ... 1337
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:PATLength ... 1338
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:PCONdition ... 1338
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:SRNGe ... 1339
TRIGger<m>:SERPattern:COMPlex:PATTern<n>:TOPTtern ... 1339
TRIGger<m>:SERPattern:COMPlex:SOFLength ... 1336
TRIGger<m>:SERPattern:COMPlex:SOFPattern ... 1337
TRIGger<m>:SERPattern:COMPlex:SOFState ... 1336
TRIGger<m>:SERPattern:COMPlex:SOFTimeout ... 1336
TRIGger<m>:SERPattern:EBTB:ALIGnment ... 1340
TRIGger<m>:SERPattern:EBTB:APPend ... 1341
TRIGger<m>:SERPattern:EBTB:CLEar ... 1341
TRIGger<m>:SERPattern:EBTB:COUNt? ... 1341
TRIGger<m>:SERPattern:EBTB:ERRor<n>:DISParity ... 1342
TRIGger<m>:SERPattern:EBTB:ERRor<n>:SYMBol ... 1342
TRIGger<m>:SERPattern:EBTB:FID ... 1342
TRIGger<m>:SERPattern:EBTB:SCRMble ... 1344
TRIGger<m>:SERPattern:EBTB:SYMBols<n>:DMINus? ... 1342
TRIGger<m>:SERPattern:EBTB:SYMBols<n>:DPLus? ... 1343
TRIGger<m>:SERPattern:EBTB:SYMBols<n>:EBIT ... 1343
TRIGger<m>:SERPattern:EBTB:SYMBols<n>:KDSYmbol ... 1343
TRIGger<m>:SERPattern:I30B:APATtern ... 1346
TRIGger<m>:SERPattern:I30B:PATTern ... 1346
TRIGger<m>:SERPattern:I30B:SKAFter ... 1346
TRIGger<m>:SERPattern:I30B:SSTP ... 1347
TRIGger<m>:SERPattern:I30B:SSTR ... 1347
TRIGger<m>:SERPattern:I30B:SYMType ... 1347
TRIGger<m>:SERPattern:I32B:APATtern ... 1344
TRIGger<m>:SERPattern:I32B:PATTern ... 1344
TRIGger<m>:SERPattern:I32B:SKAFter ... 1345
TRIGger<m>:SERPattern:I32B:SSTP ... 1345
TRIGger<m>:SERPattern:I32B:SSTR ... 1345
TRIGger<m>:SERPattern:I32B:SYMType ... 1345
TRIGger<m>:SERPattern:PRBS:ERATe ... 1339
TRIGger<m>:SERPattern:PRBS:NBRTe ... 1340
TRIGger<m>:SERPattern:PRBS:PSEQuence ... 1340
TRIGger<m>:SERPattern:PTMDe ... 1328
TRIGger<m>:SETHold:CSOurce:EDGE ... 1321
TRIGger<m>:SETHold:CSOurce:LEVel ... 1322
TRIGger<m>:SETHold:CSOurce[:VALue] ... 1322
TRIGger<m>:SETHold:HTIMe ... 1323
TRIGger<m>:SETHold:STIMe ... 1323
TRIGger<m>:SLEW:DELTa ... 1321
TRIGger<m>:SLEW:RANGe ... 1320
TRIGger<m>:SLEW:RATE ... 1320
TRIGger<m>:SLEW:SLOPe ... 1319
TRIGger<m>:SOURce:SBSelect ... 1642
TRIGger<m>:SOURce[:SELect] ... 1305
TRIGger<m>:SOURce[:SELect] ... 1641

TRIGger<m>:SOURce[:SELect] 2322
TRIGger<m>:SPATtern:CSOurce:EDGE 1330
TRIGger<m>:SPATtern:CSOurce:LEVel 1330
TRIGger<m>:SPATtern:CSOurce[:VALue] 1330
TRIGger<m>:SPATtern:REFSource 1331
TRIGger<m>:SPI:DPOPerator 1683
TRIGger<m>:SPI:DPOSition 1684
TRIGger<m>:SPI:DPTO 1684
TRIGger<m>:SPI:FCONdition 1684
TRIGger<m>:SPI:MISopattern 1684
TRIGger<m>:SPI:MOSipattern 1685
TRIGger<m>:SPI:PALignment 1683
TRIGger<m>:SPI:TYPE 1682
TRIGger<m>:SWIRe:CTYPe 2229
TRIGger<m>:SWIRe:DATA:CONDition 2230
TRIGger<m>:SWIRe:DATA:MAX 2230
TRIGger<m>:SWIRe:DATA:MIN 2230
TRIGger<m>:SWIRe:ERRor:ESC 2231
TRIGger<m>:SWIRe:ERRor:PARity 2231
TRIGger<m>:SWIRe:TIME:CONDition 2231
TRIGger<m>:SWIRe:TIME:MAX 2232
TRIGger<m>:SWIRe:TIME:MIN 2232
TRIGger<m>:SWIRe:TYPE 2232
TRIGger<m>:TBTO:CRC:CONDition 1894
TRIGger<m>:TBTO:CRC:MAX 1895
TRIGger<m>:TBTO:CRC:MIN 1895
TRIGger<m>:TBTO:DADDress:CONDition 1895
TRIGger<m>:TBTO:DADDress:MAX 1896
TRIGger<m>:TBTO:DADDress:MIN 1896
TRIGger<m>:TBTO:DATA:DCONdition 1896
TRIGger<m>:TBTO:DATA:DMAX 1897
TRIGger<m>:TBTO:DATA:DMIN 1897
TRIGger<m>:TBTO:DATA:ICONdition 1897
TRIGger<m>:TBTO:DATA:IMAX 1898
TRIGger<m>:TBTO:DATA:IMIN 1898
TRIGger<m>:TBTO:ERRor:CRC 1898
TRIGger<m>:TBTO:ERRor:FEC 1899
TRIGger<m>:TBTO:ERRor:OOR 1899
TRIGger<m>:TBTO:ERRor:ZERO 1899
TRIGger<m>:TBTO:LENGth:CONDition 1899
TRIGger<m>:TBTO:LENGth:MAX 1900
TRIGger<m>:TBTO:LENGth:MIN 1900
TRIGger<m>:TBTO:SADDress:CONDition 1901
TRIGger<m>:TBTO:SADDress:MAX 1901
TRIGger<m>:TBTO:SADDress:MIN 1901
TRIGger<m>:TBTO:TRGLevel 1894
TRIGger<m>:TBTO:TYPE 1894
TRIGger<m>:TIMeout:RANGe 1316
TRIGger<m>:TIMeout:TIME 1317
TRIGger<m>:TV:CUSTom:LDURation 1355

TRIGger<m>:TV:CUSTom:SCANmode....1355
TRIGger<m>:TV:CUSTom:SDURation....1356
TRIGger<m>:TV:CUSTom:STYPe....1355
TRIGger<m>:TV:LFIeld....1354
TRIGger<m>:TV:LINE....1353
TRIGger<m>:TV:MODE....1352
TRIGger<m>:TV:POLarity....1353
TRIGger<m>:TV:STANdard....1352
TRIGger<m>:TYPE....1306
TRIGger<m>:UART:DATA....1705
TRIGger<m>:UART:DPOPerator....1704
TRIGger<m>:UART:DPOSition....1704
TRIGger<m>:UART:DPTO....1705
TRIGger<m>:UART:FCONdition....1705
TRIGger<m>:UART:SOURce....1704
TRIGger<m>:UART:TYPE....1704
TRIGger<m>:USB:ACONdition....2125
TRIGger<m>:USB:AMAX....2125
TRIGger<m>:USB:AMIN....2125
TRIGger<m>:USB:DATA....2125
TRIGger<m>:USB:DCONdition....2126
TRIGger<m>:USB:DPOPerator....2126
TRIGger<m>:USB:DPOSition....2126
TRIGger<m>:USB:ECONdition....2127
TRIGger<m>:USB:EMAX....2127
TRIGger<m>:USB:EMIN....2127
TRIGger<m>:USB:ERRC....2127
TRIGger<m>:USB:FCONdition....2128
TRIGger<m>:USB:FMAX....2129
TRIGger<m>:USB:FMIN....2128
TRIGger<m>:USB:HAND....2129
TRIGger<m>:USB:PATT....2129
TRIGger<m>:USB:PCONdition....2130
TRIGger<m>:USB:PMAX....2130
TRIGger<m>:USB:PMIN....2130
TRIGger<m>:USB:SCONdition....2130
TRIGger<m>:USB:SMAX....2131
TRIGger<m>:USB:SMIN....2131
TRIGger<m>:USB:SPEC....2131
TRIGger<m>:USB:STCO....2132
TRIGger<m>:USB:TCONdition....2132
TRIGger<m>:USB:TMAX....2133
TRIGger<m>:USB:TMIN....2132
TRIGger<m>:USB:TOKen....2133
TRIGger<m>:USB:TYPE....2123
TRIGger<m>:USB:WADD....2133
TRIGger<m>:USB:WEND....2133
TRIGger<m>:USB:WETCheck....2134
TRIGger<m>:USB:WFRN....2134
TRIGger<m>:USB:WPAY....2134

TRIGger<m>:USB:WPID....2134
TRIGger<m>:USB:WPOR....2134
TRIGger<m>:USB:WSEU....2135
TRIGger<m>:USB:WSTC....2135
TRIGger<m>:USBPd:BIT....2206
TRIGger<m>:USBPd:DMAX....2207
TRIGger<m>:USBPd:DMIN....2207
TRIGger<m>:USBPd:DOPerator....2207
TRIGger<m>:USBPd:ERENable....2205
TRIGger<m>:USBPd:ERRor<n>:ENABle....2205
TRIGger<m>:USBPd:FIENable....2205
TRIGger<m>:USBPd:FRAMe<n>:ENABle....2205
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:BIT....2206
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DMAX....2207
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DMIN....2207
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:DOPerator....2207
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:ENABle....2205
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IMAX....2208
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IMIN....2208
TRIGger<m>:USBPd:FRAMe<n>:FLD<o>:IOPerator....2209
TRIGger<m>:USBPd:FRENable....2205
TRIGger<m>:USBPd:IMAX....2208
TRIGger<m>:USBPd:IMIN....2208
TRIGger<m>:USBPd:IOPerator....2209
TRIGger<m>:USBThree:BIT....2183
TRIGger<m>:USBThree:DMAX....2183
TRIGger<m>:USBThree:DMIN....2183
TRIGger<m>:USBThree:DOPerator....2182
TRIGger<m>:USBThree:ERENable....2181
TRIGger<m>:USBThree:ERRor<n>:ENABle....2181
TRIGger<m>:USBThree:FIENable....2181
TRIGger<m>:USBThree:FRAMe<n>:ENABle....2181
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:BIT....2183
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DMAX....2183
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DMIN....2183
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:DOPerator....2182
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:ENABle....2181
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IMAX....2185
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IMIN....2184
TRIGger<m>:USBThree:FRAMe<n>:FLD<o>:IOPerator....2184
TRIGger<m>:USBThree:FRENable....2181
TRIGger<m>:USBThree:IMAX....2185
TRIGger<m>:USBThree:IMIN....2184
TRIGger<m>:USBThree:IOPerator....2184
TRIGger<m>:WIDTh:DELTa....1311
TRIGger<m>:WIDTh:POLarity....1309
TRIGger<m>:WIDTh:RANGe....1310
TRIGger<m>:WIDTh:WIDTh....1310
TRIGger<m>:WINDow:DELTa....1316
TRIGger<m>:WINDow:RANGe....1314

TRIGger<m>:WINDow:TIME....1315
TRIGger<m>:WINDow:WIDTh....1315
TRIGger<m>:ZONE:EXPRession[:DEFine]....1374
TRPRobe:ID:PARTnumber?....1277
TRPRobe:ID:PRDate?....1277
TRPRobe:ID:SRNumber?....1278
TRPRobe:ID:SWVersion?....1277
TRPRobe:PMETer:RESults:COMMon?....1261
TRPRobe:PMETer:RESults:DIFFerential?....1261
TRPRobe:PMETer:RESults:NEGative?....1262
TRPRobe:PMETer:RESults:POSitive?....1262
TRPRobe:PMETer:RESults:SINGle?....1260
TRPRobe:PMETer:VISibility....1260
TRPRobe:SETup:ACCoupling....1265
TRPRobe:SETup:ADVanced:AUDioverload....1267
TRPRobe:SETup:ADVanced:FILTer....1266
TRPRobe:SETup:ATTenuation:DEFProbe....1274
TRPRobe:SETup:ATTenuation:MANual....1263
TRPRobe:SETup:ATTenuation:MODE....1263
TRPRobe:SETup:ATTenuation:UNIT....1263
TRPRobe:SETup:ATTenuation[:AUTO]?....1256
TRPRobe:SETup:BANDwidth?....1256
TRPRobe:SETup:CAPacitance?....1278
TRPRobe:SETup:CMOFfset....1264
TRPRobe:SETup:DEGauss....1275
TRPRobe:SETup:DISPlaydiff....1260
TRPRobe:SETup:GAIN:MANual....1264
TRPRobe:SETup:IMPedance?....1278
TRPRobe:SETup:NAME?....1256
TRPRobe:SETup:PRMode....1267
TRPRobe:SETup:STATe?....1255
TRPRobe:SETup:TYPE?....1255
TRPRobe:SETup:ZAXV....1265
USRDefined:RST:NAME....1231
USRDefined:RST:OPEN....1231
USRDefined:RST:SAVE....1231
USRDefined:RST[:ENABle]....1231
WAVeform<m>:XYCurve:RATio....1410
WAVeform<m>:XYCurve:STATe....1411
WAVeform<m>:XYCurve:SWAP....1411
WAVeform<m>:XYCurve:XSOurce....1412
WAVeform<m>:XYCurve:YSOurce....1412
WGENerator<m>:ACOPy....2334
WGENerator<m>:ARBGen:COPY....2345
WGENerator<m>:ARBGen:MULTichannel:IMPort....2345
WGENerator<m>:ARBGen:MULTichannel:NAME....2345
WGENerator<m>:ARBGen:MULTichannel:OPEN....2346
WGENerator<m>:ARBGen:NAME....2346
WGENerator<m>:ARBGen:OPEN....2346
WGENerator<m>:ARBGen:RUNMode....2347

WGENerator<m>:ARBGen:RUNSingle ..... 2348
WGENerator<m>:ARBGen:SAMPles? ..... 2347
WGENerator<m>:ARBGen:SELect ..... 2347
WGENerator<m>:ARBGen:SRATe ..... 2348
WGENerator<m>:ARBGen[:SOURce] ..... 2348
WGENerator<m>:COUPling:AMPLitude ..... 2355
WGENerator<m>:COUPling:AWForm ..... 2356
WGENerator<m>:COUPling:NOISe ..... 2356
WGENerator<m>:COUPling:PHASeshift ..... 2355
WGENerator<m>:COUPling:STATe ..... 2355
WGENerator<m>:COUPling[:FREQuency] ..... 2356
WGENerator<m>:FREQuency ..... 2335
WGENerator<m>:FUNCtion:PULSe[:WIDTh] ..... 2336
WGENerator<m>:FUNCtion:RAMP[:SYMMetry] ..... 2336
WGENerator<m>:FUNCtion:SQUare:DCYCle ..... 2337
WGENerator<m>:FUNCtion[:SELect] ..... 2335
WGENerator<m>:MODulation:AM:DCYCle ..... 2338
WGENerator<m>:MODulation:AM:DEPTh ..... 2338
WGENerator<m>:MODulation:AM:FREQuency ..... 2338
WGENerator<m>:MODulation:AM:SYMMetry ..... 2339
WGENerator<m>:MODulation:AM[:FUNCtion] ..... 2339
WGENerator<m>:MODulation:CARRier:FREQuency ..... 2339
WGENerator<m>:MODulation:CARRier:PERiod ..... 2339
WGENerator<m>:MODulation:FM:DCYCle ..... 2340
WGENerator<m>:MODulation:FM:DEViation ..... 2340
WGENerator<m>:MODulation:FM:FREQuency ..... 2340
WGENerator<m>:MODulation:FM:SYMMetry ..... 2341
WGENerator<m>:MODulation:FM[:FUNCtion] ..... 2341
WGENerator<m>:MODulation:FSK:FONE ..... 2341
WGENerator<m>:MODulation:FSK:FTWO ..... 2341
WGENerator<m>:MODulation:FSK[:RATE] ..... 2342
WGENerator<m>:MODulation:NDCLevel ..... 2352
WGENerator<m>:MODulation:NLABsolute? ..... 2353
WGENerator<m>:MODulation:NLPCent ..... 2353
WGENerator<m>:MODulation:NOISe ..... 2353
WGENerator<m>:MODulation:PWM:DCYCle ..... 2342
WGENerator<m>:MODulation:PWM:DEPTh ..... 2342
WGENerator<m>:MODulation:PWM:FREQuency ..... 2343
WGENerator<m>:MODulation:PWM:SYMMetry ..... 2343
WGENerator<m>:MODulation:PWM[:FUNCtion] ..... 2343
WGENerator<m>:MODulation:TYPE ..... 2337
WGENerator<m>:OUTPut[:LOAD] ..... 2350
WGENerator<m>:PERiod ..... 2336
WGENerator<m>:PRESet ..... 2335
WGENerator<m>:SOURce ..... 2334
WGENerator<m>:SWEep:FSTart ..... 2344
WGENerator<m>:SWEep:TIME ..... 2344
WGENerator<m>:SWEep[:FEND] ..... 2344
WGENerator<m>:VOLTage:DCLevel ..... 2351
WGENerator<m>:VOLTage:HIGH ..... 2351

WGENerator<m>:VOLTage:INVersion....2351
WGENerator<m>:VOLTage:LOW....2352
WGENerator<m>:VOLTage:OFFSet....2352
WGENerator<m>:VOLTage[:VPP]....2351
WGENerator<m>[:ENABle]....2334
ZVC:BANDwidth....1281
ZVC:RESCoupled....1283
ZVC:TYPE....1281
ZVC:Z<m>:I<n>:BANDwidth?....1281
ZVC:Z<m>:I<n>:DATA:HEADer?....1293
ZVC:Z<m>:I<n>:DATA[:VALues]?....1293
ZVC:Z<m>:I<n>:IMPedance....1282
ZVC:Z<m>:I<n>:OFFSet....1282
ZVC:Z<m>:I<n>:OVERload:RSTO....1283
ZVC:Z<m>:I<n>:OVERload:VALue?....1284
ZVC:Z<m>:I<n>:POSition....1284
ZVC:Z<m>:I<n>:SCALe....1284
ZVC:Z<m>:I<n>:SHUNt:EVALue....1285
ZVC:Z<m>:I<n>:SHUNt:MAXCurrent....1285
ZVC:Z<m>:I<n>:SHUNt:MAXVoltage....1286
ZVC:Z<m>:I<n>:SHUNt:MODE....1285
ZVC:Z<m>:I<n>:SHUNt:MXCValue?....1286
ZVC:Z<m>:I<n>:SKEW....1286
ZVC:Z<m>:I<n>:ZERComp:DETect....1291
ZVC:Z<m>:I<n>:ZERComp:STATe?....1292
ZVC:Z<m>:I<n>:ZERComp:USE....1291
ZVC:Z<m>:I<n>[:STATe]....1287
ZVC:Z<m>:ID:NAME?....1290
ZVC:Z<m>:ID:PARTnumber?....1290
ZVC:Z<m>:ID:SRNumber?....1291
ZVC:Z<m>:ID:SWVersion?....1291
ZVC:Z<m>:V<n>:BANDwidth?....1287
ZVC:Z<m>:V<n>:DATA:HEADer?....1293
ZVC:Z<m>:V<n>:DATA[:VALues]?....1293
ZVC:Z<m>:V<n>:IMPedance....1288
ZVC:Z<m>:V<n>:OFFSet....1288
ZVC:Z<m>:V<n>:OVERload:RSTO....1288
ZVC:Z<m>:V<n>:OVERload:VALue?....1288
ZVC:Z<m>:V<n>:POSition....1289
ZVC:Z<m>:V<n>:SCALe....1289
ZVC:Z<m>:V<n>:SKEW....1290
ZVC:Z<m>:V<n>[:STATe]....1290

# Index

## Symbols


8b/10b
- Trigger settings ... 787
- Triggering ... 790

100BASE-T1
- Configuration settings ... 646
- Configuring ... 650
- Trigger settings ... 650
- Triggering ... 655

1000BASE-T1
- Configuration settings ... 663
- Configuring ... 667
- Trigger settings ... 667
- Triggering ... 673


## A


Acquisition
- Decimation ... 132
- Modes ... 116
- Single, multiple ... 37
- Start and stop ... 37
- Time ... 117

Acquisition modes ... 116
Acquisition time ... 117
Actions
- Mask test ... 434

Active probe
- Differential ... 121
- Micro button ... 149
- Offset compensation ... 165
- Overview ... 120
- ProbeMeter ... 150
- Setup ... 153

Active waveform ... 52
ADC overflow ... 1159
ADC sample rate ... 115
ADC samples ... 115
Aligning
- Input channels ... 105

Alignment ... 103
Amplitude measurement ... 342
Amplitude/time measurements
- Settings ... 346

Analog
- Parallel bus display ... 942

Annotate ... 60
Appearance
- Color tables ... 79

Application, external ... 110
Apps key ... 43
Area level
- Amplitude/time measurements ... 347

Area measurements ... 345
Arithmetic ... 133
Arranging waveforms ... 53
Auto clear
- Search results ... 469

Auto Norm
- Key (trigger) ... 37

Auto trigger mode ... 37
Auto-logon ... 1133
Auto, trigger mode ... 226
Autocorrelation ... 288
Automatic measurements
- Configuration settings ... 323
- Display settings ... 327
- Gate settings ... 329

Autonaming ... 107
Autoset
- for protocols ... 517
- Key ... 43

AutoZero ... 147
Average ... 133
- FFT ... 412
- Mathematics ... 296

Average count ... 130, 133, 297
Average count (N-single count) ... 130, 133, 297


## B


Bandwidth ... 143
- Instrument ... 114
- Probe ... 114
- Spectrum measurement ... 363

Bench top operation ... 25
Biased correlation ... 288
Big endian
- CAN ... 572
- LIN ... 594

Bit pattern editor ... 524
Bit string
- serial pattern, MSO ... 960

Blackman Harris window (FFT) ... 410
Brochure ... 20
Bug report ... 2476
Burst width ... 343
Bus key ... 39


## C


C<n> keys ... 39
Cable, USB ... 32
Camera key ... 42
CAN
- Configuration settings ... 559
- Data length ... 572
- Data pattern ... 572
- Endianness ... 572
- Frame types ... 570
- ID type ... 570
- Identifier ... 571
- Transfer order ... 572
- Trigger settings ... 567
- Trigger types ... 568

Center frequency
- Spectrum measurement ... 366

Channel
- Bandwidth measurement ... 366
- C<n> keys ... 39
- Center frequency ... 366
- Offset ... 141
- Waveforms ... 51

Channel power
- Spectrum measurement ... 363

Clear status
Remote ... 1197
Clipping ... 1159
Clock
Parallel bus ... 942
Color tables ... 79
Comb
Parallel bus display ... 942
Command sequence
Remote ... 1201
Commands
Finding a command ... 1172
Compensation, passive probes ... 189
Computer name
Changing ... 1139
Configuration
hardware ... 106
Connectors
Channel input ... 32
External trigger input ... 32
Coupled zoom ... 270
quick access ... 59, 263
Coupling ... 112
Channel ... 142
Trigger ... 257
Waveform generator settings ... 991
Cross-correlation ... 288
CSV
Waveform export ... 488
CSV export ... 110
Cursor
Saving data ... 505
Settings for data export ... 490
Used for gating ... 329
Cursor key ... 42
Cursor measurements
Disabling ... 310
Enable ... 312
Envelope waveform ... 315
How to ... 309
Peak excursion ... 319, 367
Peak search ... 317
Results ... 308
Settings ... 312
Cursors ... 59
Configuring display ... 312
Configuring measurement ... 310
Coupling ... 316
Display ... 316
Position ... 314
Start measurement ... 310
Track waveform ... 310, 315
Type ... 314
Custom:
Filter settings ... 775
Custom: Manchester / NRZ
Configuration settings ... 750
Configuring ... 769
Decoding ... 778
Trigger settings ... 770
Triggering ... 775
Customer support ... 2476
Cycle area measurement ... 345
Cycle measurements ... 342

## D

D-PHY
Basics ... 702
Configuration settings ... 703
Configuring ... 706
Triggering ... 712
Data entry ... 64
Data length
LIN ... 594
Data Length Code
CAN ... 572
Data pattern
CAN ... 572
LIN ... 594
Data security ... 2477
Data sheet ... 20
Data2Clock
Search ... 460
Trigger ... 210
Trigger (MSO) ... 955
DDR
Configuration settings ... 916
Configuring ... 919
Decode results ... 923
Decimation ... 116, 132
Decode
Custom protocol, decode layer ... 768
D-PHY, decode layer ... 705
DDR, decode layer ... 918
Ethernet, decode layer ... 632, 648, 665
M-PHY, decode layer ... 722
MDIO, decode layer ... 798
SpaceWire, decode layer ... 873
USBPD, decode layer ... 860
Decode layer
Custom protocol ... 768
D-PHY ... 705
DDR ... 918
Ethernet ... 632, 648, 665
M-PHY ... 722
MDIO ... 798
SpaceWire ... 873
USBPD ... 860
Decode table
MSO ... 948
Decoded bus
Parallel bus display ... 942
Default
File names ... 109
Path for saving ... 109
Default values
Remote ... 1199
Delay ... 343
Delay measurement ... 348
Delete ... 61
Demo ... 59
Device footprint ... 2476
DHCP ... 1138
DHCP server
LAN configuration ... 1139
Diagrams ... 45
Dialog boxes
Usage ... 63
Dialogs ... 48
Differential probe ... 121

Digital channels ..... 938
Configuration settings ..... 938
Configuring ..... 946
Decode table ..... 948
Export ..... 963
Search ..... 964
Triggering ..... 962
Digital filter ..... 145
Digital resolution ..... 944
Digital signals ..... 938
Digital waveforms ..... 51
Display
Intensity ..... 95
Overview ..... 44
XY-diagram ..... 304, 306
Zoom ..... 263
Display elements ..... 48
Diagram ..... 44
Signal bar ..... 44
DisplayPort ..... 30, 35
DNS server
LAN configuration ..... 1139
Duty cycle
Eye ..... 357
Duty cycle measurement ..... 343


## E


Edge
Search conditions ..... 453
Trigger ..... 199
Edge count ..... 346
Edge trigger
MSO ..... 951
Electric power (math. function) ..... 289
Enable registers
Remote ..... 1199
Endianness
CAN ..... 572
LIN ..... 594
Envelope ..... 133
FFT ..... 412
Mathematics ..... 296
Measurements ..... 325
Envelope waveform
Cursor measurement ..... 315
Error report ..... 2476
ESR ..... 1155
Ethernet
Configuration settings ..... 629
Configuring ..... 634
Trigger settings ..... 635
Triggering ..... 639
Event actions
Mask test ..... 434
Event status enable register (ESE)
Remote ..... 1197
Event status register (ESR)
Remote ..... 1197
Export
Data ..... 490
Digital channels ..... 963
History ..... 505
Include horizontal values ..... 494
Integer format ..... 493
Parallel buses ..... 963
Raw values ..... 493
Waveform histograms ..... 497
XML, CSV, BIN ..... 483
Expressions ..... 284
External application ..... 110
External monitor ..... 30, 35
External trigger input ..... 256
External trigger out ..... 230
External trigger output ..... 35
Extinction ratio
Eye ..... 357
Eye
Amplitude ..... 357
Base ..... 357
Bit rate ..... 357
Fall time ..... 357
Height ..... 357
Jitter ..... 357
Measurement results ..... 357
Rise time ..... 357
Top ..... 357
Width ..... 357
Eye measurements
Characteristics ..... 355
Results ..... 357
Settings ..... 358


## F


Fail condition, mask test ..... 434
Fail criteria, mask test ..... 423
Fall time ..... 343
FFT ..... 60, 399
Configuring ..... 403
Fundamentals ..... 399
Gating ..... 412
measurement speed ..... 402
MSO ..... 964
Peak excursion ..... 319, 367
Performance considerations ..... 404
Search ..... 449
Setup ..... 407, 411
Window types ..... 410
File formats
Waveform export ..... 483
File import
Restrictions ..... 483
File names
Default ..... 107, 109
Filter
Custom ..... 775
Find
Remote control command ..... 67
Find trigger level ..... 59
FIR (math. function) ..... 289
FIR filter ..... 292
Flattop2 window (FFT) ..... 410
Formula editor
Reference ..... 284
Using ..... 291
Formulas
Advanced expressions ..... 284
Operator editor ..... 282
Frame types
CAN ..... 570
Frontend
information ..... 106

Frontpanel
information ... 106
Function generator option ... 33

## G

Gain, vertical ... 112
Gate
Coupled to cursor ... 329
Coupled to zoom ... 329
Saving data ... 505
Search ... 465
Search, defining ... 467
Settings ... 329
Settings for data export ... 490
Gate area
Configuring ... 329
Coupling to zoom ... 331, 414, 467
Gating
FFT ... 412
Gaussian window (FFT) ... 410
GBIP connector ... 35
Getting started ... 19
Glitch
Search conditions ... 454
Trigger ... 200
GPIB
Address ... 1151
Remote control interface ... 1150
GPIB bus control
Remote ... 1199
Grid ... 46
Grounding ... 27

## H

Hamming window (FFT) ... 410
Hann window (FFT) ... 410
Hardware
information ... 106
Hardware CDR ... 220
Hardware check ... 106
Hardware zoom ... 59, 263
HD key ... 42
HDMI ... 35
HDMI connector ... 35
Header files ... 483
Help ... 19, 58
Open ... 67
Search for topic ... 68
Using ... 68
High definition ... 136
High res ... 132
High signal level measurement ... 342
Histogram
Saving ... 482
Histogram measurements
Settings ... 375
Histograms ... 60, 368
Area ... 48
averaging ... 339
Characteristics ... 368
Configuring ... 372
Diagram ... 48
Exporting waveform histograms ... 497
Horizontal ... 368
Jitter measurement ... 372
Markers ... 376
Measurement histograms ... 387
Measurement results ... 368
Saving waveform histograms ... 497
Setup ... 373
Toolbar icons ... 372
Tubes ... 339
Vertical ... 368
History
Export ... 490, 505
MSO ... 947
time stamp ... 302
History key ... 42
Hold time ... 349
Holdoff ... 227
TV trigger ... 216
Using ... 195
Horizontal
Label ... 47
Position ... 38
Horizontal position ... 118
Hysteresis
Reference level ... 333
Trigger ... 228

## I

I²C
Basics ... 525
Configuration settings ... 527
Configuring ... 529
Decode results ... 535
Trigger settings ... 530
Triggering ... 535
Icons
Measurement status ... 326
ID type
CAN ... 570
Identification
Remote ... 1197
Identifier
CAN ... 571
LIN ... 593
Import waveforms
Restrictions ... 483
Infinite persistence ... 94, 932, 1080
Instrument settings
Loading ... 475
Recall ... 1199
Save ... 1200
Saving ... 480
Intensity ... 95
Intensity key ... 42
Intensity of display elements ... 42
Interfaces
GPIB ... 1151
LAN ... 1150
Interlaced scanning ... 220
Interleaved sample rate ... 115
Interleaved X/Y ... 494
Interpolation ... 130
Interval
Search conditions ... 458
Trigger ... 207
IP address ... 1150
Changing ... 1138
Remote Desktop ... 1148

IST ..... 1155
IST flag
Remote ..... 1198

## J

Jitter
Measurement via histogram ..... 372
Jitter measurements
Settings ..... 1026

## K

Kaiser Bessel window (FFT) ..... 410
Keep X-grid fixed ..... 82
Keep Y-grid fixed ..... 82
Keyboard
On-screen ..... 64
Keyboard, connecting ..... 30
Keypad ..... 64

## L

Labels
Waveform ..... 86
LAN ..... 1142
Configuration ..... 1137, 1142
Connector ..... 35
Environment ..... 1137
Interface ..... 1150
IP address ..... 1150
LAN configuration ..... 1143
Ping ..... 1145
Remote control interface ..... 1150
Resource string ..... 1150
Level knob ..... 36
Level measurements ..... 342
Limit checks
Actions ..... 395
Configuring ..... 393
Enable ..... 394
Performing ..... 393
Violation settings ..... 395
Limits
Range ..... 394
LIN
Configuration settings ..... 587
Configuring ..... 591
Data length ..... 594
Data pattern ..... 594
Endianness ..... 594
Identifier ..... 593
Transfer order ..... 594
Trigger settings ..... 591
Trigger types ..... 592
Triggering ..... 595
Little endian
CAN ..... 572
LIN ..... 594
Load instrument settings ..... 1199
Loading
User settings ..... 475
Lock touchscreen ..... 42
Logic analyzer ..... 938
Logic key ..... 39
Logic probe connector ..... 33
Logical operations on channels ..... 214, 232
Logical thresholds
MSO, configuring ..... 947
MSO, settings ..... 939
Login ..... 1133
Long-term measurements
Basics ..... 386
Configuring ..... 386
Horizontal scale ..... 391
Number of points ..... 391
Period of time per point ..... 391
Saving ..... 500
Scale mode ..... 391
Settings ..... 387
Low signal level measurement ..... 342

## M

M-PHY
Basics ..... 719
Configuration settings ..... 720
Configuring ..... 722
Decode results ..... 738
Trigger settings ..... 723
Mainboard
information ..... 106
Margins
Range ..... 394
Marker limit
Histograms ..... 376
Markers
Histograms ..... 368, 376
Mask definition
Settings ..... 427
Type ..... 426
Mask setup ..... 427
Mask test ..... 60, 423
Appearance settings ..... 437
Display settings ..... 436, 437
Display, configuring ..... 446
Event actions, configuring ..... 446
Event actions, settings ..... 434
Fail condition ..... 434
Fail criteria ..... 423
Fail criteria, configuring ..... 446
History, testing ..... 448
Performing ..... 447
Results ..... 423
Setting up ..... 446
Starting ..... 447
State, mask test result ..... 447
Test settings ..... 425
Violation tolerance ..... 434
Mask test results
Acq. completed ..... 424
Acq. remaining ..... 424
Acquisition hits ..... 424
Fail rate ..... 424
Sample hits ..... 424
State ..... 424
Test result ..... 425
Mask testing ..... 423
Masks
Creating user mask ..... 441
Definition ..... 423, 427
Load settings ..... 426
Loading ..... 447
modifying user mask ..... 443

Save settings ... 426
Saving ... 447
Segments, point settings ... 429
Segments, settings ... 428
User mask definition ... 427
Waveform mask definition ... 430
Math key ... 40
Math waveforms ... 40, 51
Arithmetic ... 296
Displaying ... 279
FFT ... 399
FFT Gating ... 412
FFT setup ... 407, 411
Formula editor ... 284
Operator editor ... 282
Operators ... 282, 284
Saving ... 279
Scale ... 295
Setup ... 280
Max value measurement ... 342
Maximum sample rate ... 115
MDIO
Configuration settings ... 796
Configuring ... 799
Trigger settings ... 800
Triggering ... 803
MDIO protocol
Basics ... 795
Mean measurement ... 342
Measure key ... 42, 321
Measurement ... 60
Measurement histograms ... 387
Measurement types
Area ... 345
Arithmetic ... 346
Counting ... 346
Eye ... 357
Histograms ... 368
Protocol ... 378
Spectrum ... 363
Time ... 343
Measurements
Amplitude/time ... 341
Configuring ... 322
Delay ... 348
Icons in results table ... 326
Pulse train count ... 351
Results ... 325
Setup/Hold ... 349
Source for analysis ... 388
Starting ... 321
Statistics ... 384
Median
Histograms ... 368
Memory ... 298
Menu ... 48
Micro button ... 120, 149
Min value measurement ... 342
Mixed signal option ... 938
Mixed Signal Option ... 33
Monitor ... 30, 35, 96
Mouse, connecting ... 30
MSO ... 33, 938
FFT ... 964
History ... 947
Resolution ... 944
Time qualification ... 958
Trigger settings ... 950
Zoom ... 947
Multi channel export
Enable ... 491
Remote control ... 1252, 1613
Results in header file ... 486
Multiple measurements ... 391

## N

N dB down
Spectrum ... 366
Network
Environment ... 1137
Noise
Eye ... 357
Hysteresis settings ... 228
Normal trigger mode ... 37
Normal, trigger mode ... 226
Notification ... 47
Numeric data entry ... 64

## O

Occupied bandwidth
Spectrum measurement ... 363, 366
OCXO ... 35, 127
Offset ... 40
Active probe ... 165
Channel ... 141
On/Off key ... 36
One File ... 476
Open source acknowledgment ... 20
Operating system ... 1132
Access ... 1136
service packs ... 1134
Settings ... 1136
Operation
Concepts ... 43
Manual ... 44
Operation complete
Remote ... 1198
Operator
Type ... 282
Operators
Formula editor mathematics ... 284
Optical
Average power ... 377
Modulation amplitude ... 377
Options
hardware installation ... 32
Identification (remote) ... 1198
R&S RTP-B1 (MSO) ... 33
R&S RTP-B1E (for R&S RT-ZVC) ... 33
R&S RTP-B6 (waveform and function generator) ... 33
Output buffer ... 1155
Overshoot measurement ... 342

## P

Parallel buses ... 938
Clock setup ... 942
Configuration settings ... 938
Configuring ... 946
Decode table ... 948
Export ... 963
Triggering ... 962

Parallel poll register enable
Remote ... 1199
Passive probe
Settings ... 151
Password ... 1133
Remote Desktop ... 1148
Pattern
Qualification, state trigger ... 214, 232
Pattern generator
Setup ... 985
Pattern search condition ... 213
Pattern trigger ... 213
MSO ... 958
PCIe
Trigger settings ... 892
PCIe Gen 1/2
Search settings ... 904
Searching ... 915
Triggering ... 902
Peak detect ... 132
Peak excursion
Cursor measurement ... 319, 367
Search ... 319, 367
Spectrum measurement ... 319, 367
Peak search
Cursor measurement ... 309, 310
Spectrum measurement ... 309, 317
Peak to peak measurement ... 342
Performance ... 97
Considerations for FFT ... 404
FFT parameters ... 402
Performance test ... 128
Ping ... 1145
Position ... 40
Horizontal ... 38, 118
Vertical ... 40, 112
Position mode ... 85
Power
Connector ... 34
On/Off ... 27
PPE ... 1155
Preset ... 101
Preset key ... 43
Printing
Settings ... 508
Probability markers
Histograms ... 376
ProbeMeter ... 120
Measurement ... 342
Setting ... 150
Probes
Active ... 120
AutoZero ... 147
Compensation ... 32
Differential ... 121
Passive, compensation ... 189
Setup ... 146
Progressive scanning ... 220
Projector ... 30
Protocols
8b/10b trigger settings ... 787
100BASE-T1 configuration settings ... 646
100BASE-T1 trigger settings ... 650
1000BASE-T1 configuration settings ... 663
1000BASE-T1 trigger settings ... 667
CAN configuration settings ... 559
Custom: filter settings ... 775
Custom: Manchester / NRZ configuration settings ... 750
Custom: Manchester / NRZ trigger settings ... 770
D-PHY configuration settings ... 703
Data format ... 518
DDR configuration settings ... 916
DDR decode results ... 923
Display settings ... 518
Ethernet configuration settings ... 629
Ethernet trigger settings ... 635
I²C configuration settings ... 527
I²C decode results ... 535
I²C trigger settings ... 530
LIN configuration settings ... 587
LIN trigger settings ... 591
M-PHY configuration settings ... 720
M-PHY decode results ... 738
M-PHY trigger settings ... 723
MDIO configuration settings ... 796
MDIO trigger settings ... 800
PCIe Gen 1/2 search settings ... 904
PCIe trigger settings ... 892
SpaceWire configuration settings ... 871
SPI configuration settings ... 540
SPI decode results ... 547
SPI trigger settings ... 545
UART decode results ... 557
USB 2.0 configuration settings ... 813
USB 2.0 trigger settings ... 818
USB 3.1 configuration settings ... 844
USB 3.1 trigger settings ... 848
USBPD configuration settings ... 858
USBPD trigger settings ... 861
Pulse count ... 346
Pulse train count ... 351

## Q

Q-factor
Eye ... 357
Quick measurements ... 60

## R

R&S RT-ZVC
multi-channel power probe ... 174
Rack mounting ... 26
Recall instrument settings ... 1199
Recall intermediate ... 1199
Record length ... 115, 117
Rectangular window (FFT) ... 410
Redo ... 58
Ref In ... 35
Ref key ... 39
Ref Out ... 35
Reference levels
Basics ... 332
Configuring ... 333
Defining automatically ... 333
Defining manually ... 333
displaying ... 328
Displaying ... 334
Hysteresis ... 333
Level settings ... 335
Settings ... 335
Tubes ... 333
Reference point ... 38, 47, 118

Reference signal
1 GHz, internal ... 128
External ... 127
OCXO ... 127
Reference waveforms ... 39, 51
Displaying ... 271
Loading ... 271
Saving ... 271
Registers ... 1155
Release notes ... 20
Remote commands
Finding a command ... 1172
Remote control ... 43
Find command using help ... 67
GPIB address ... 1151
Interfaces ... 1150
Protocols ... 1150
Starting ... 1153
Remote Desktop ... 43, 1147
Remote operation ... 43, 1145, 1147
Repetitive, trigger mode ... 226
Reset
Histogram, long-term meas., statistics ... 390
Reset values
Remote ... 1199
Resolution ... 112, 115, 117
16 bit ... 136
Resolution enhancement ... 116
Resource string
LAN ... 1150
Restoring
Settings ... 102
Result boxes
Displaying ... 61
Result lines
Displaying ... 328
Result tables ... 48
Results
Clear ... 327
Configuring display ... 61
Lines in display ... 334
Mask test ... 423
Measurements ... 325
Reset long-term measurement ... 390
Saving ... 499
RFFE
Advanced ... 685
Trigger settings ... 687
Triggering ... 694
Rise time ... 343
RMS
Acquisition setting ... 132
RMS measurement ... 342
Roll mode ... 124
Strart roll time ... 125
RS232 ... 550
Basics ... 550
Run / Stop ... 58
Run / Stop key ... 37
Runt
Search conditions ... 455
Trigger ... 202

## S

Safety instructions ... 20
Sample rate ... 115
Samples ... 115
History ... 298
Save instrument settings ... 1200
Save intermediate ... 1200
Save Reference ... 61
Save Waveform ... 61
Save/Recall
Remote ... 1604
User preferences ... 478
User settings ... 475, 476
Waveforms ... 490
Saveset ... 475, 480
Saving
File name generation ... 107
History ... 490, 505
Instrument settings ... 475, 480
Long-term measurements ... 500
Preset ... 101
Results ... 499
Screenshots ... 511
User preferences ... 478
Waveform histograms ... 497
Waveform segments ... 490, 505
Waveforms ... 490, 505
Scale
Horizontal, rotary knob ... 38
Math waveforms ... 295
Vertical ... 112, 141
Vertical, rotary knob ... 40
Scaling
Horizontal ... 123
Long-term measurements ... 390
Long-term measurements, horizontal ... 391
Measurement histograms ... 390
Scanning system (TV signals) ... 220
SCPI
Finding a command ... 1172
Screen resolution ... 30
Screenshot
Saving ... 511
Web interface ... 1146
Screenshots ... 508
Meta information ... 507
Print and save ... 507
Search ... 59
Auto clear ... 469
Clearing results ... 469
Conditions ... 449
Control ... 449
Copy trigger settings ... 452
Data2Clock ... 460
Definition ... 449
Edge ... 453
Enable ... 451
FFT ... 449
Gate ... 465
Glitch ... 454
Interval ... 458
On digital signals ... 964
PCIe Gen 1/2 settings ... 904
Peak excursion ... 319, 367
Results ... 450
Runt ... 455
Scope ... 449
Setup & Hold ... 460
Show results ... 469
Slew rate ... 459

Sorting results ... 469
Source ... 451
Timeout ... 457
Trigger level ... 453, 454, 455, 458, 459, 461
Trigger search ... 452
Width ... 454
Window ... 456
Zoom window, configuring ... 472
Search conditions
Pattern ... 213
Searching
In help ... 68
Secured environment ... 2477
Select
Waveform ... 53
Selected waveform ... 52
Self-alignment ... 103, 105, 1197
Self-test
Remote ... 1200
Selftest ... 106
Serial pattern trigger ... 241
MSO ... 960
Service manual ... 20
Service packs ... 1134
Service request enable register (SRE)
Remote ... 1200
Settling time ... 343
Setup & Hold
Search ... 460
Trigger ... 210
Trigger (MSO) ... 955
Setup time ... 349
Setup/Hold measurement ... 349
Setup/Hold ratio ... 349
Show labels ... 85
Signal bar ... 47
Signal icons ... 47
Signal label ... 47
Signal levels
Configuring ... 333
Signal preview ... 52
Signal threshold
Amplitude/time measurements ... 347
Signal view ... 47
Signals
Color settings ... 78
Single ... 58
Single key ... 37
Skew ... 125
Slew rate
Trigger ... 208
Slew Rate
Search conditions ... 459
Slope key ... 37
SmartGrid ... 53
Source key ... 37
SpaceWire
Advanced ... 872
Basics ... 870
Configuration settings ... 871
Configuring ... 874
Spectrogram ... 61
Spectrum
Key ... 41
Spectrum measurements
Channel BW ... 366
Channel CF ... 366
Cursor measurement ... 309, 310
N dB down ... 366
Occupied bandwidth ... 366
Peak detection ... 310
Peak excursion ... 319, 367
Peak search ... 309
Results ... 363
Settings ... 365
SPI
Configuration settings ... 540
Configuring ... 544
Decode results ... 547
Trigger settings ... 545
Triggering ... 547
SPI protocol
Basics ... 539
SRE ... 1155
Standard
TV trigger ... 217
Standard deviation
Histograms ... 368
Standard deviation measurement ... 342
State trigger ... 212
MSO ... 957
Statistics
Enable ... 387
Measurement results ... 384
Status
Measurements ... 326
Status byte
Remote ... 1197, 1200
Status registers
Overview ... 1154
STATus:OPERation ... 1155
Status reporting system
Common commands ... 1196
Status reports ... 1154
STB ... 1155
Style (waveforms) ... 95
Switch off
Waveform ... 53
Switch on
Waveform ... 52
Switching of transitions ... 346


## T


Technical support ... 2476
Temperature
Changes ... 105
Termination ... 112
Text entry ... 64
Thresholds
MSO, configuring ... 947
MSO, settings ... 939
Time base ... 38, 117
Time measurements ... 343
Time qualification
MSO ... 958
Time scale ... 38, 117, 123
Time stamp ... 302
Timebase ... 123
Timeout
Search conditions ... 457
Trigger ... 206
Timeout trigger
MSO ... 954

Toolbar ... 55
Annotate ... 60
Coupled zoom ... 59, 263
Cursor ... 59
Delete ... 61
FFT ... 60
Find level ... 59
Hardware zoom ... 59, 263
Hide/show icons ... 56
Histogram ... 60
Histogram icons ... 372
Load saveset ... 58
Masks ... 60
Measure ... 60
Measurement ... 321
Overview ... 57
Quick measurement ... 60
Redo ... 58
Run / Stop ... 58
Save Reference ... 61
Save Waveform ... 61
Search ... 59
Show tooltips ... 58
Single ... 58
Undo ... 58
Zone trigger ... 61
Zoom ... 59, 263
Tooltips ... 58
Show ... 67
Total harmonic distortion
Spectrum measurement ... 363
Touch Lock key ... 42
Touchscreen
Adjusting ... 100
Compared with mouse ... 44
Control elements ... 48
Lock, unlock ... 42
Usage ... 44
Track
Enable ... 387
Track waveform
Cursor measurement ... 310, 315
Transfer order
CAN ... 572
LIN ... 594
Transition switching count ... 346
Transition trigger
See Slew rate trigger ... 208
Transparency ... 42
Trigger
8b/10b settings ... 787
100BASE-T1 settings ... 650
1000BASE-T1 settings ... 667
CAN settings ... 567
Custom: Manchester / NRZ settings ... 770
Data2Clock ... 210
Data2Clock (MSO) ... 955
Delay (Holdoff) ... 227
Edge ... 199
Edge (MSO) ... 951
Ethernet settings ... 635
Event (definition) ... 193
Event (remote) ... 1200
External input ... 256
External output ... 35
Force ... 227
Glitch ... 200
Holdoff ... 227
I²C settings ... 530
Information ... 194
Interval ... 207
Label ... 47
Level ... 36, 46
Level, search ... 453, 454, 455, 458, 459, 461
LIN settings ... 591
M-PHY settings ... 723
MDIO settings ... 800
Mode ... 37, 226
MSO settings ... 950
Pattern ... 213
Pattern (MSO) ... 958
PCIe settings ... 892
Position ... 46
Run ... 226
Runt ... 202
Search ... 452
Sequence (definition) ... 193
Serial pattern ... 241
Serial pattern (MSO) ... 960
Settings, copy to search ... 452
Setup & Hold ... 210
Setup & Hold (MSO) ... 955
Slew rate ... 208
Slope ... 37
Source ... 37
Sources, MSO ... 950
SPI settings ... 545
State ... 194, 212
State (MSO) ... 957
Timeout ... 206
Timeout (MSO) ... 954
TV (video) ... 216
USB 2.0 settings ... 818
USB 3.1 settings ... 848
USBPD settings ... 861
Width ... 201
Width (MSO) ... 952
Window ... 204
Trigger level
Find ... 59
Trigger offset ... 118
Trigger out
Pulse on limit violation ... 396
Settings ... 230
Trigger sources
MSO ... 950
Trigger types
CAN ... 568
Edge ... 199
Glitch ... 200
Interval ... 207
LIN ... 592
Pattern ... 213
Runt ... 202
Serial pattern ... 241
Setup & Hold ... 210
Slew rate ... 208
Timeout ... 206
Width ... 201
Window ... 204
Trigger Types ... 198
State ... 212
TV/Video ... 216

Tubes
Configuring ... 339
Histograms ... 339
Reference level ... 333
TV trigger ... 216
Type
Trigger ... 198

## U

UART ... 550
Basics ... 550
Configuring ... 554
Decode results ... 557
Triggering ... 557
Unbiased correlation ... 288
Undersampling ... 115
Undo ... 58
Unlock touchscreen ... 42
USB
Cable ... 32
Configuring ... 817
Connecting devices ... 29
Connector ... 32
Device ... 35
Triggering ... 851
USB 2.0
Configuration settings ... 813
Trigger settings ... 818
Triggering ... 825
USB 3.1
Configuration settings ... 844
Configuring ... 847
Trigger settings ... 848
USB 3.1 protocol
Basics ... 834
USB protocol
Basics ... 808
USBPD
Configuration settings ... 858
Configuring ... 861
Trigger settings ... 861
Triggering ... 864
User ID
Remote Desktop ... 1148
User manual ... 19
User mask
Settings ... 427
User name ... 1133
User settings
Saving ... 475

## V

Vertical
Controls ... 38
Position ... 112
Resolution ... 112
Scale ... 112, 141
Vertical system ... 112
Video (TV) trigger ... 216
Violation
Limit checks ... 395
Violation of limits
Actions ... 395
Violation tolerance, mask test ... 434
Virus protection ... 1134
VISA
Resource string ... 1150
VNC ... 43, 1145

## W

Wait
Remote ... 1201
Waveform
Arithmetic ... 116
Data query ... 1252
Loading ... 482
Saving ... 482
Waveform arithmetic ... 133
Waveform count
Histograms ... 368
Waveform export
CSV ... 488
Data query ... 1252
Files and formats ... 483
Header files ... 483
Remote control ... 1613
Value files ... 488
XML ... 488
Waveform generator
AM modulation ... 975
Arbitrary ... 980
Coupling ... 991
DC offset alignment ... 990
FM modulation ... 977
FSK modulation ... 979
Function generator ... 968
Modulation ... 973
Noise ... 989
Output ... 988
PWM modulation ... 976
Settings ... 966
Sweep ... 979
Waveform generator option ... 33
Waveform mask
Settings ... 430
Waveform samples ... 115
Waveform Value files ... 488
Waveforms ... 51
Arrange ... 53
Clipping ... 1159
Overview and usage ... 51
Preview ... 53
Saving ... 490, 505
Select ... 53
States ... 52
Style ... 95
Switch off ... 53
Switch on ... 52
Zooming ... 261
Web browser ... 1142
Web control ... 1145
Web interface ... 43, 1142
Browser ... 1142
LAN configuration ... 1143
Width
Search conditions ... 454
Trigger ... 201
Width trigger
MSO ... 952

Window
Search conditions ..... 456
Trigger ..... 204
Windows ..... 1132
Access ..... 1136
Settings ..... 1136

## X

XML
Waveform export ..... 488
xx
Advanced ..... 648
XY-diagram
Displaying ..... 306
settings ..... 304
XY-waveforms ..... 51

## Z

Zone key ..... 37
Zone trigger ..... 37, 61
Zoom ..... 261
Area ..... 47
Area, coupling ..... 331, 414, 467
Coupled ..... 270
Diagram ..... 47
Key ..... 38
Methods ..... 261
MSO ..... 947
Multiple ..... 269
On the touchscreen ..... 267
Position/Range ..... 269
Procedures ..... 267
Quick access ..... 59, 263
Saving data ..... 505
Settings ..... 263
Settings for data export ..... 490
Start-stop values ..... 268
Start/Stop settings ..... 264
Used for gating ..... 329
Waveforms ..... 51