

R&S®NRP18S-xx

High-Power Three-Path Diode Power Sensors

User Manual



This manual describes the following high-power three-path diode power sensors with firmware version FW 02.00 and later:

- R&S®NRP18S-10 (1424.6721.02)
- R&S®NRP18S-20 (1424.6738.02)
- R&S®NRP18S-25 (1424.6744.02)

© 2020 Rohde & Schwarz GmbH & Co. KG

Mühldorfstr. 15, 81671 München, Germany

Phone: +49 89 41 29 - 0

Fax: +49 89 41 29 12 164

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.

Trade names are trademarks of the owners.

1178.3686.02 | Version 05 | R&S®NRP18S-xx

Throughout this manual, products from Rohde & Schwarz are indicated without the ® symbol, for example R&S®NRP18S-10 is abbreviated as R&S NRP18S-10.

Contents

1	Preface.....	5
1.1	Documentation Overview.....	5
1.2	Key Features.....	6
2	Safety Information.....	8
3	Preparing for Use.....	9
3.1	Unpacking and Checking the Power Sensor.....	9
3.2	Operating Conditions.....	9
3.3	Considerations for Test Setup.....	10
3.4	Connecting to a DUT.....	10
3.5	Connecting a Cable to the Host Interface.....	11
3.6	Connecting to a Controlling Host.....	12
4	Power Sensor Tour.....	16
4.1	RF Connector.....	16
4.2	Trigger I/O Connector.....	17
4.3	Host Interface.....	17
4.4	Status LED.....	17
5	Operating Concepts.....	18
5.1	R&S NRP Toolkit.....	18
5.2	Remote Control.....	20
5.3	R&S NRPV.....	21
5.4	R&S Power Viewer.....	25
5.5	R&S Power Viewer Mobile.....	29
5.6	R&S NRX.....	29
5.7	R&S NRP2.....	31
6	Firmware Update.....	35
6.1	Hardware and Software Requirements.....	35
6.2	Updating the Firmware.....	35
7	Replacing an R&S NRP-Zxx with an R&S NRP18S-xx.....	39
7.1	Most Important Differences.....	39

7.2	Prerequisites.....	39
8	Remote Control Commands.....	41
8.1	Conventions Used in SCPI Command Descriptions.....	41
8.2	Notations.....	41
8.3	Common Commands.....	43
8.4	Preparing for the Measurement.....	47
8.5	Controlling the Measurement.....	49
8.6	Configuring and Retrieving Results.....	64
8.7	Configuring the Measurement Modes.....	69
8.8	Configuring Basic Measurement Parameters.....	81
8.9	Calibrating/Zeroing the Power Sensor	101
8.10	Testing the Power Sensor.....	103
8.11	Configuring the System.....	104
8.12	Using the Status Register.....	116
9	Performing Measurement Tasks - Programming Examples.....	120
9.1	Performing the Simplest Measurement.....	120
9.2	Performing the Fastest Measurement in Continuous Average Mode.....	120
9.3	Performing a Buffered Continuous Average Measurement.....	123
9.4	Performing Trace Measurements.....	125
9.5	Trace Measurement with Synchronization to Measurement Complete.....	126
10	Remote Control Basics.....	128
10.1	Remote Control Interfaces and Protocols.....	128
10.2	SCPI Command Structure.....	129
10.3	Status Reporting System.....	135
11	Troubleshooting.....	153
11.1	Displaying Status Information.....	153
11.2	Performing a Selftest.....	153
11.3	Problems during a Firmware Update.....	153
11.4	Contacting Customer Support.....	154
	List of Commands.....	155
	Index.....	160

1 Preface

This chapter provides an overview of the user documentation and an introduction to the R&S NRP18S-xx.

1.1 Documentation Overview

This section provides an overview of the R&S NRP18S-xx user documentation. Unless specified otherwise, you find the documents on the R&S NRP18S-xx product page at:

www.rohde-schwarz.com/product/nrp18s

1.1.1 Getting Started Manual

Introduces the R&S NRP18S-xx and describes how to set up and start working with the product. Includes basic operations and general information, e.g. safety instructions, etc. A printed version is delivered with the power sensor.

1.1.2 User Manuals

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

1.1.3 Tutorials

Tutorials offer guided examples and demonstrations on operating the R&S NRP18S-xx. They are provided on the product page of the internet.

1.1.4 Instrument Security Procedures

Deals with security issues when working with the R&S NRP18S-xx in secure areas. It is available for download on the Internet.

1.1.5 Basic Safety Instructions

Contains safety instructions, operating conditions and further important information. The printed document is delivered with the instrument.

1.1.6 Data Sheets and Brochures

The data sheet contains the technical specifications of the R&S NRP18S-xx. 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.

www.rohde-schwarz.com/brochure-datasheet/nrp18s

1.1.7 Release Notes and Open Source Acknowledgment (OSA)

The release notes list new features, improvements and known issues of the current firmware version, and describe the firmware installation.

The "Open Source Acknowledgment" is provided on the user documentation CD-ROM, included in the delivery. It contains verbatim license texts of the used open source software.

www.rohde-schwarz.com/firmware/nrp18s

1.1.8 Application Notes, Application Cards, White Papers, etc.

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

www.rohde-schwarz.com/application/nrp18s

1.2 Key Features

The 3-path diode power sensors are members of the R&S NRP series power sensors from Rohde & Schwarz.

They provide a high-speed USB interface that constitutes both the communication port and the power supply connection.

Special features of power sensors with precision attenuators

Each R&S NRP18S-xx power sensor is delivered with a high-quality attenuator. It enables measuring at higher power levels.



Figure 1-1: R&S NRP18S-10, R&S NRP18S-20 and R&S NRP18S-25 (from left)

To provide physically correct results, the sensor itself numerically compensates the influence of the attenuator. Hence, the measurement output is not the power level that is measured at the input port of the sensor. It represents the power level at the input of the attenuator.

This numerical compensation of a two-port ("S-parameter device") connected upstream is also implemented in R&S NRP power sensors that are delivered without an attenuator. You can select any third-party attenuator, measure its S-parameters and connect it upstream of the sensor in place of the supplied attenuator.

However, a precise measurement of the S-parameters is not trivial, and not all attenuators are sufficiently stable. To enable high-precision power measurements with R&S NRP18S-xx power sensors, each included attenuator is highly stable and accurately calibrated on site. The factory-provided calibration data is stored in the factory calibration data set of the sensor and includes the precisely determined S-parameter values.

At start-up of the power sensor, it automatically activates S-parameter correction based on the stored S-parameters. You can change this default behavior using the S-parameters tool included in the R&S NRP Toolkit, see [Chapter 8.8.4.3, "S-Parameter Correction"](#), on page 87.

- If you use the power sensor without an attenuator to increase sensitivity, disable S-parameter correction.
- If you use the power sensor with a third-party attenuator, load its S-parameters and define it as the active device.

For a detailed specification, refer to the data sheet.

2 Safety Information

The product documentation helps you use the R&S NRP18S-xx safely and efficiently. Follow the instructions provided here and in the printed "Basic Safety Instructions". Keep the product documentation nearby and offer it to other users.

Intended use

The R&S NRP18S-xx is intended for the development, production and verification of electronic components and devices in industrial, administrative, and laboratory environments. Use the R&S NRP18S-xx 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 about the potential dangers and gives instructions how to prevent personal injuries or damage caused by dangerous situations. Safety information is provided as follows:

- The printed "Basic Safety Instructions" provide safety information in many languages and are delivered with the R&S NRP18S-xx.
- Throughout the documentation, safety instructions are provided when you need to take care during setup or operation.

3 Preparing for Use

For information on safety, see:

- [Chapter 2, "Safety Information"](#), on page 8
- [Chapter 3.2, "Operating Conditions"](#), on page 9

3.1 Unpacking and Checking the Power Sensor

Check the equipment for completeness using the delivery note and the accessory lists for the various items. Check the power sensor for any damage. If there is damage, immediately contact the carrier who delivered the power sensor. Make sure not to discard the box and packing material.



Packing material

Retain the original packing material. If the instrument needs to be transported or shipped later, you can use the material to protect the control elements and connectors.

3.2 Operating Conditions

⚠ CAUTION

Risk of injury through hot surfaces

Under certain conditions, the maximum surface temperatures of the power sensor can exceed the limits defined in EN 61010-1.

Provide protection against unintentional contact and wear appropriate personal protective equipment when touching the sensor.

Specific operating conditions are required to ensure accurate measurements and to avoid damage to the power sensor and connected devices. Before switching on the power sensor, observe the information on appropriate operating conditions provided in the basic safety instructions and the data sheet of the power sensor.

In particular, ensure the following:

- The power sensor is dry and shows no sign of condensation.
- The ambient temperature does not exceed the range specified in the data sheet.
- Signal levels at the input connectors are all within the specified ranges.
- Signal outputs are connected correctly and are not overloaded.

3.3 Considerations for Test Setup

Preventing electrostatic discharge (ESD)

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

- **NOTICE!** Risk of electrostatic discharge (ESD). Electrostatic discharge (ESD) can damage the electronic components of the power sensor and the device under test (DUT).

Ground yourself to avoid electrostatic discharge (ESD) damage:

- Using a wrist strap and cord, connect yourself to the ground.
- Use a conductive floor mat and heel strap combination.

EMI impact on measurement results

Electromagnetic interference (EMI) may affect the measurement results.

To suppress generated electromagnetic interference (EMI):

- Use suitable shielded cables of high quality. For example, use double-shielded RF and LAN cables.
- Always terminate open cable ends.
- Note the EMC classification in the data sheet.

3.4 Connecting to a DUT

For connecting the power sensor to a DUT, use the RF connector. See [Chapter 4.1, "RF Connector"](#), on page 16.

NOTICE

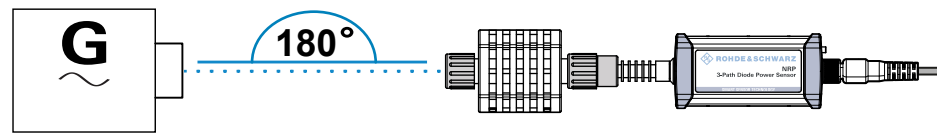
Risk of overloading the sensor

Using a power sensor at a level above its upper measuring limit can damage the sensor head. To avoid this risk, make sure not to exceed the test limit.

The test limits specified on the type label are valid only for the supplied attenuator. For operation without attenuator, lower test limits apply, as specified in the data sheet.

To connect to the DUT

1. Ensure that the RF connector of your DUT is compatible with the RF connector of the power sensor or attenuator.
2. Insert the RF connector straight into the RF output of your DUT. Take care not to tilt it.



3. **NOTICE!** Risk of damaging the center pin of the RF connector. Always rotate only the hex nut of the RF connector. Never rotate the power sensor itself.
Tighten the RF connector manually.

To disconnect from the DUT

- **NOTICE!** Risk of damaging the center pin of the RF connector. Always rotate only the union nut of the RF connector. Never rotate the power sensor itself.
If the power sensor is operated with the supplied attenuator:
 - a) Carefully loosen the union nut of the RF connector at the input of the attenuator.
 - b) Remove the sensor with the attenuator.
 Normally, it is not necessary or advisable to separate the attenuator from the sensor if both are operated together.
- If the sensor is operated without the supplied attenuator, carefully loosen the union nut at the RF connector of the sensor and then remove the sensor.

3.5 Connecting a Cable to the Host Interface

For connecting the power sensor to a USB host, use the host interface. See [Chapter 4.3, "Host Interface"](#), on page 17.

Depending on the USB host, use one of the following cables:

- Computer or R&S NRP-Z5 sensor hub:
R&S NRP-ZKU cable with a USB connector, R&S order number 1419.0658.xx
See [Chapter 3.6.1, "Computer"](#), on page 12.
- Base units, R&S NRX or R&S NRP2, or other supported Rohde & Schwarz instruments:
R&S NRP-ZK6 cable with a push-pull type connector, R&S order number 1419.0664.xx
See [Chapter 3.6.2, "Base Unit"](#), on page 15.

These cables can be obtained in different lengths up to 5 meters.

To connect a cable to the host interface of the power sensor

1. Insert the screw-lock cable connector into the host interface connector of the power sensor.
2. Tighten the union nut manually.

To disconnect the host interface of the power sensor

- Loosen the union nut of the screw-lock cable connector and remove the cable.

3.6 Connecting to a Controlling Host

As a controlling host, you can use:

- [Computer](#)
- [Base Unit](#)

For operating the power sensor, you can choose from various possibilities. For details, see [Chapter 5, "Operating Concepts"](#), on page 18.

3.6.1 Computer

If the controlling host is a computer, you can operate the power sensor using a supported software, the web user interface or remote control. For details, see [Chapter 5, "Operating Concepts"](#), on page 18.

- Establish the connection using:
 - Host interface
See [Chapter 3.6.1.1, "Simple USB Connection"](#), on page 12.
See [Chapter 3.6.1.2, "R&S NRP-Z5 Sensor Hub Setup"](#), on page 13.

3.6.1.1 Simple USB Connection

All R&S NRP18S-xx power sensors can be connected to the USB interface of a computer.

Required equipment

- R&S NRP18S-xx power sensor
- R&S NRP-ZKU cable

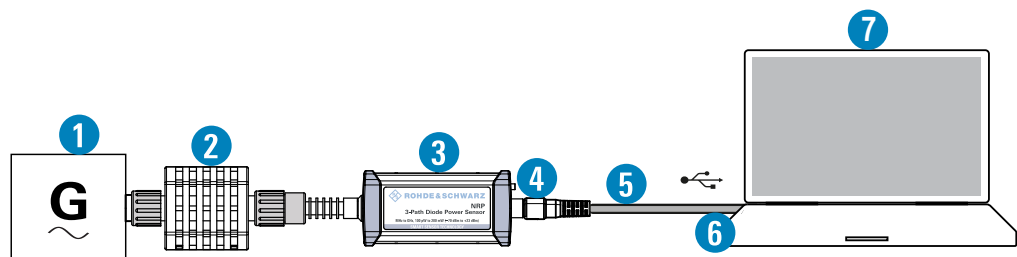
Setup

Figure 3-1: Setup with an R&S NRP-ZKU cable

- 1 = Signal source
- 2 = Attenuator
- 3 = R&S NRP18S-xx power sensor
- 4 = Host interface connector
- 5 = R&S NRP-ZKU cable
- 6 = USB connector
- 7 = Computer with installed VISA driver or R&S NRP Toolkit



Incorrectly connecting/disconnecting the R&S NRP18S-xx power sensors can damage the power sensors or lead to erroneous results.

Ensure that you connect/disconnect your power sensor as described in [Chapter 3, "Preparing for Use"](#), on page 9.

1. Connect the cables as shown in [Figure 3-1](#) :
 - a) Connect the R&S NRP-ZKU cable to the power sensor.
See ["To connect a cable to the host interface of the power sensor"](#) on page 11.
 - b) Connect the R&S NRP-ZKU cable to the computer.
 - c) Connect the power sensor to the signal source.
2. On the computer, start a software application to view the measurement results.
See [Chapter 5, "Operating Concepts"](#), on page 18.

3.6.1.2 R&S NRP-Z5 Sensor Hub Setup

The R&S NRP-Z5 sensor hub (high-speed USB 2.0) can host up to four R&S NRP18S-xx power sensors and provides simultaneous external triggering to all connected sensors.

Required equipment

- 1 to 4 R&S NRP18S-xx power sensors
- 1 R&S NRP-ZK6 cable per sensor
- R&S NRP-Z5 sensor hub with external power supply unit and USB cable
- BNC cables to connect the trigger input and trigger output signals (optional)

Setup

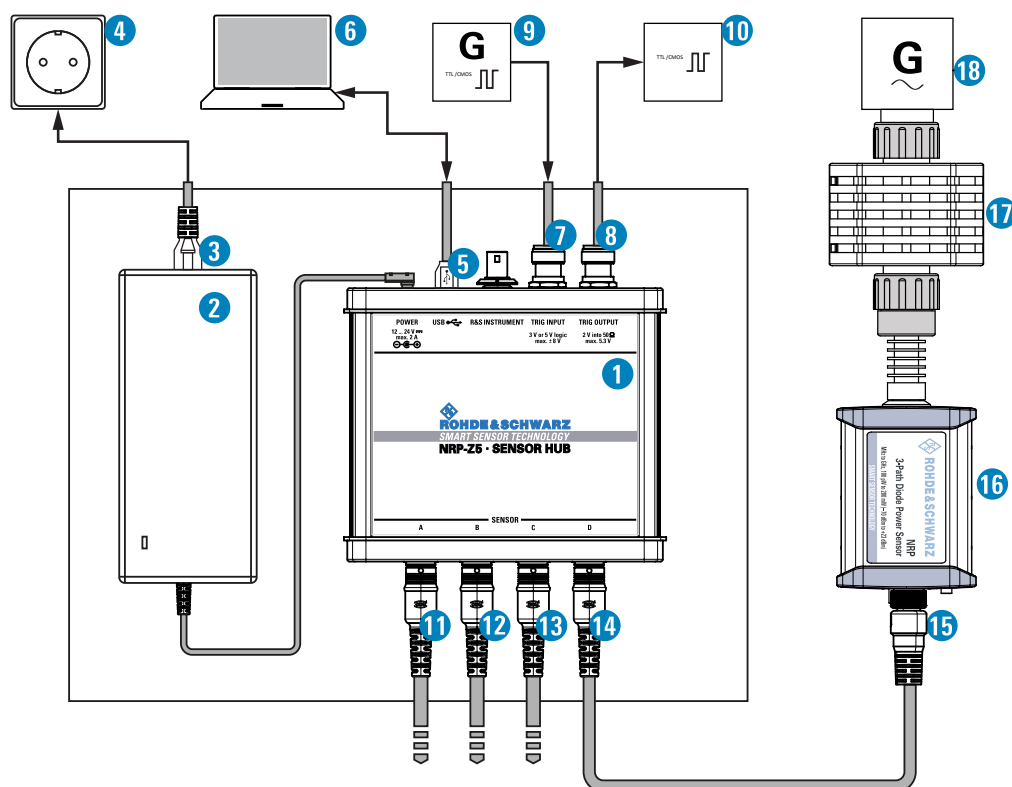


Figure 3-2: Setup with an R&S NRP-Z5 sensor hub

- 1 = R&S NRP-Z5 sensor hub
- 2 = External power supply unit (supplied)
- 3 = Power cable (supplied)
- 4 = AC power supply
- 5 = USB cable (supplied)
- 6 = Computer with USB host interface
- 7, 8 = BNC cable (optional, not supplied)
- 9 = Trigger source (optional)
- 10 = Triggered device (optional)
- 11-14 = R&S NRP-ZK6 cable
- 15 = Host interface connector
- 16 = R&S NRP18S-xx power sensor
- 17 = Attenuator
- 18 = Signal source



Incorrectly connecting/disconnecting the R&S NRP18S-xx power sensors can damage the power sensors or lead to erroneous results.

Ensure that you connect/disconnect your power sensor as described in [Chapter 3, "Preparing for Use"](#), on page 9.

1. Connect the cables as shown in [Figure 3-2](#):

- a) Connect the R&S NRP-ZK6 cable to the power sensor.
See ["To connect a cable to the host interface of the power sensor"](#) on page 11
 - b) Connect the power sensors to the R&S NRP-Z5 sensor hub. You can connect up to four sensors.
 - c) Connect the R&S NRP-Z5 to the computer.
 - d) Connect the power sensors to the signal source.
 - e) Connect the delivered external power supply unit to the R&S NRP-Z5 and to an AC supply connector.
 - f) Connect the trigger input of the R&S NRP-Z5 with a BNC cable to the trigger source (optional).
 - g) Connect the trigger output of the R&S NRP-Z5 with a BNC cable to the trigger device (optional).
2. On the computer, start a software application to view the measurement results.
See [Chapter 5, "Operating Concepts"](#), on page 18.

3.6.2 Base Unit

As a controlling host, you can use an R&S NRX or R&S NRP2 base unit. You can also operate the power sensor using other supported Rohde & Schwarz instruments with a sensor connector. For details, see also the user manual of the instrument.

- Establish the connection with the base unit using:
 - Host interface
See [Chapter 5.6, "R&S NRX"](#), on page 29
See [Chapter 5.7, "R&S NRP2"](#), on page 31

4 Power Sensor Tour

This chapter provides an overview of the available connectors and LEDs of the power sensor.

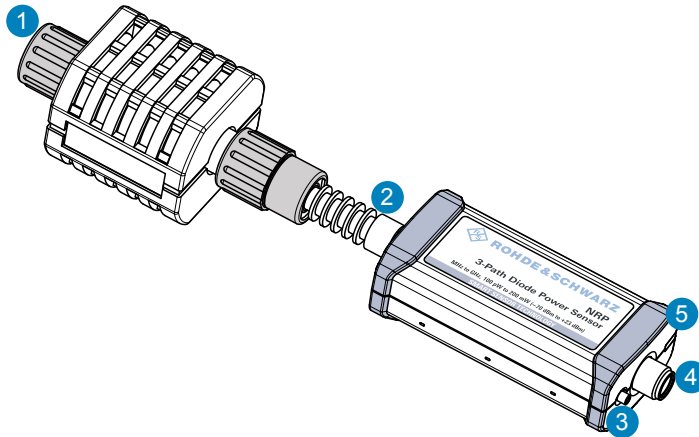


Figure 4-1: R&S NRP18S-xx power sensor

- 1 = RF connector (attenuator), see [Chapter 4.1, "RF Connector"](#), on page 16
- 2 = RF connector (sensor), see [Chapter 4.1, "RF Connector"](#), on page 16
- 3 = Trigger I/O connector, see [Chapter 4.2, "Trigger I/O Connector"](#), on page 17
- 4 = Host interface connector, see [Chapter 4.3, "Host Interface"](#), on page 17
- 5 = Status LED, see [Chapter 4.4, "Status LED"](#), on page 17

4.1 RF Connector

The RF connector is used for connecting the power sensor to a device under test (DUT) or a signal generator. See [Chapter 3.4, "Connecting to a DUT"](#), on page 10.

The RF connector type is an N connector. Use it to connect to the following:

- Included attenuator, which is in turn connected to a device under test (DUT) or a signal generator.
- DUT or signal generator, directly connected.
- Two-port device ("S-parameter device"), which in turn is connected to a DUT or a signal generator.

For maximum measurement accuracy, tighten the RF connector using a torque wrench with a nominal torque as specified in the following table.

Table 4-1: R&S NRP18S-xx RF connector characteristics

Power sensor	Male connector	Matching female connector	Tightening torque
R&S NRP18S	N	N	1.36 Nm (12" lbs)

4.2 Trigger I/O Connector

The trigger I/O is a connector of SMB type.

It is used as an input for signals if the trigger source parameter is set to `EXtErnal2`. It is used as an output for trigger signals if the sensor is operated in the trigger master mode.

Further information:









- [Chapter 8.5.2, "Triggering"](#), on page 50

4.3 Host Interface

The host interface is used for establishing a connection between the power sensor and a USB host. For this purpose, an external cable is needed. See [Chapter 3.5, "Connecting a Cable to the Host Interface"](#), on page 11.

4.4 Status LED

The status LED gives information about the state of the power sensor. The following states are defined:

Indication		State
	White	Idle state. The sensor performs no measurement and is ready for use.
	Flashing white	Firmware update is in progress
	Slow flashing white	Sanitizing in progress
	Yellow	Wait for trigger state
	Green	Measuring state
	Turquoise blue	Zeroing is in progress
	Slow flashing red	Static error You can query the error type with <code>SYSTem:SERRor?</code> .
	Fast flashing red	Critical static error You can query the error type with <code>SYSTem:SERRor?</code> . Note: If this state occurs after a firmware update, the update was not successful. Perform the firmware update again. See also Chapter 11.3, "Problems during a Firmware Update" , on page 153.

5 Operating Concepts

For operating the power sensor, you can choose from various possibilities:

- [Chapter 5.2, "Remote Control"](#), on page 20
- [Chapter 5.3, "R&S NRPV"](#), on page 21
- [Chapter 5.4, "R&S Power Viewer"](#), on page 25
- [Chapter 5.5, "R&S Power Viewer Mobile"](#), on page 29
- [Chapter 5.6, "R&S NRX"](#), on page 29
- [Chapter 5.7, "R&S NRP2"](#), on page 31

5.1 R&S NRP Toolkit



Before you start using the power sensor, it is recommended to install the R&S NRP Toolkit.

The R&S NRP Toolkit is the basic software package that supplies low-level drivers and tools for all power sensors. The components of the R&S NRP Toolkit depend on the operating system.

5.1.1 Versions and Downloads

The R&S NRP Toolkit is available for the Microsoft Windows operating systems listed under [Chapter 5.1.2, "System Requirements"](#), on page 18, Linux distributions and MacOSX. Several R&S NRP Toolkit versions are available on your documentation CD-ROM.

The latest version for Windows is available at www.rohde-schwarz.com/software/nrp-toolkit.

To obtain an R&S NRP Toolkit for an operating system other than Microsoft Windows, contact the Rohde & Schwarz customer support: customersupport@rohde-schwarz.com

5.1.2 System Requirements

Hardware requirements:

- Desktop computer or laptop, or an Intel-based Apple Mac

Supported operating systems:

- Microsoft Windows versions
 - Microsoft Windows Vista 32/64-bit
 - Microsoft Windows 7 32/64-bit

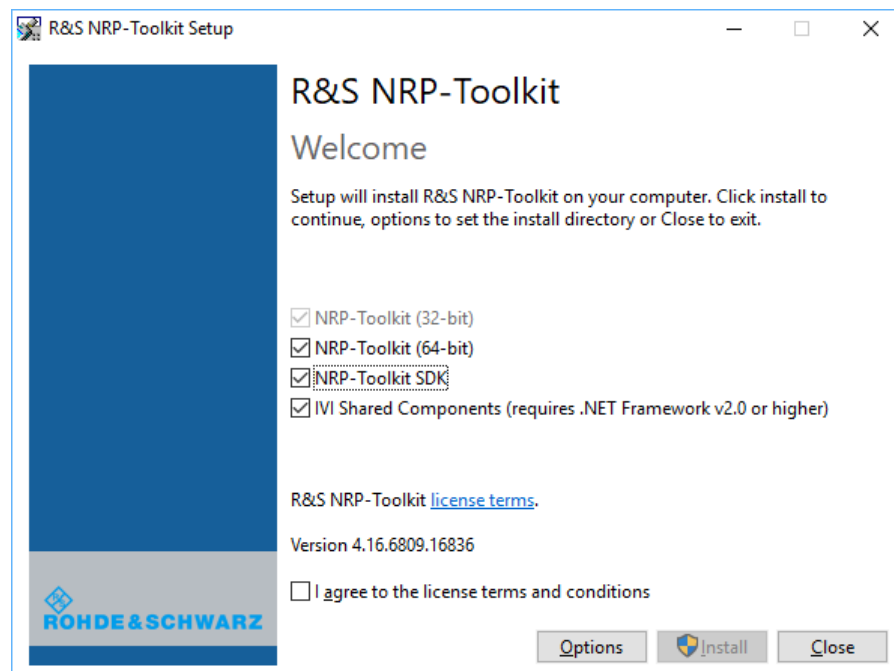
- Microsoft Windows 8/ 8.1 32/64-bit
- Microsoft Windows 10 32/64-bit
- For information on other operating systems, see [Chapter 5.1.1, "Versions and Downloads"](#), on page 18.

5.1.3 R&S NRP Toolkit for Windows

The R&S NRP Toolkit installer for Windows-based systems contains the components described in the release notes available at www.rohde-schwarz.com/software/nrp-toolkit.

Installing on a computer

1. Start the R&S NRP Toolkit installer on the Windows-based computer.
In the "NRP-Toolkit Setup" dialog, the correct R&S NRP Toolkit version for your operating system, 32-bit or 64-bit, is already selected.
2. Enable the components you want to install.
 - "NRP-Toolkit (SDK)"
The software development kit (SDK) provides programming examples for the R&S power sensors.
See [Chapter 9, "Performing Measurement Tasks - Programming Examples"](#), on page 120.
 - "IVI Shared Components"
Installs the USBTMC driver. Enabled by default because the installation is recommended.
See also [Table 10-1](#).



3. Accept the license terms to continue with the installation.
4. Click "Next" and complete the installation process.

5.1.3.1 Components of the R&S NRP Toolkit

Access: "Start" > "NRP-Toolkit"

The following tools are part of the R&S NRP Toolkit for Windows.

Configure Network Sensor

Useful if you have troubles establishing a LAN connection with an R&S NRP LAN power sensor. The tool provides the following functions:

- Configuring the network settings by (temporary) connecting the selected sensor to the computer using USB.
- Discovering the sensors that have been configured via the Zeroconf (APIA) protocol.

The tool comes with a guide (PDF) that is also available in the "Start" menu. The guide explains the network setup.

Firmware Update

You can use the Firmware Update for NRP Family program to load new firmware for the power sensors.

See [Chapter 6, "Firmware Update"](#), on page 35.

NRP Version Display

Displays version information of all installed, power measurement-relevant software packages.

R&S NRP-Z Uncertainty Calculator

Determines the expanded measurement uncertainty. The tool comes with a manual (PDF) that is also available in the "Start" menu.

S-Parameter Update Multi

Helps loading an S-parameter table into the power sensor.

See [Chapter 8.8.4.5, "Using the S-Parameters Tool"](#), on page 91.

Terminal

Low-level communication program for sending commands to the power sensor.

5.2 Remote Control

You can remote control the R&S NRP18S-xx easily.

Further information:

- [Chapter 8, "Remote Control Commands"](#), on page 41
- [Chapter 10, "Remote Control Basics"](#), on page 128
- [Chapter 10.1, "Remote Control Interfaces and Protocols"](#), on page 128
- [Chapter 3.6.1, "Computer"](#), on page 12

5.3 R&S NRPV

The R&S NRPV enables you to measure power in all available measurement modes. Also, you can use up to four power sensors simultaneously.

The R&S NRPV is provided on your documentation CD-ROM and on the Rohde & Schwarz website as a separate standalone installation package.

Required equipment

- R&S NRP18S-xx power sensor
- R&S NRP-ZKU cable or an R&S NRP-Z5 sensor hub and an R&S NRP-ZK6 cable to connect the sensor to the computer
- Windows computer with installed:
 - R&S NRP Toolkit V 4.17 or higher
 - R&S NRPV version 3.2 or higher (refer to the operating manual of the R&S NRPV for a description of the installation process)

Table 5-1: Operating an R&S NRP18S-xx with or without attenuator


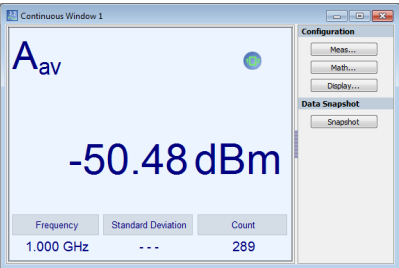

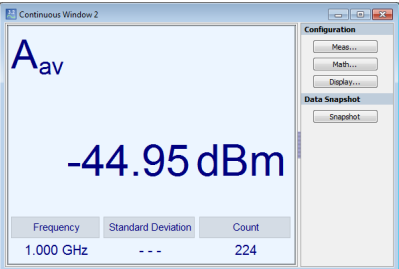
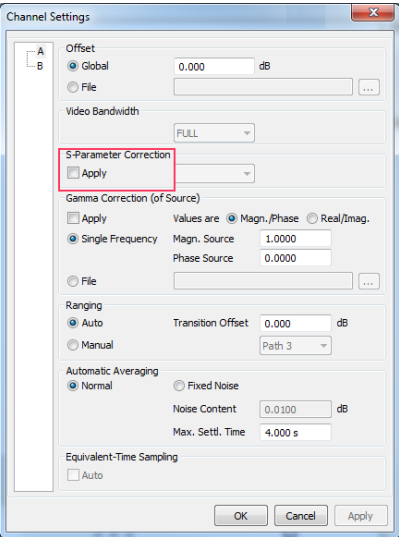

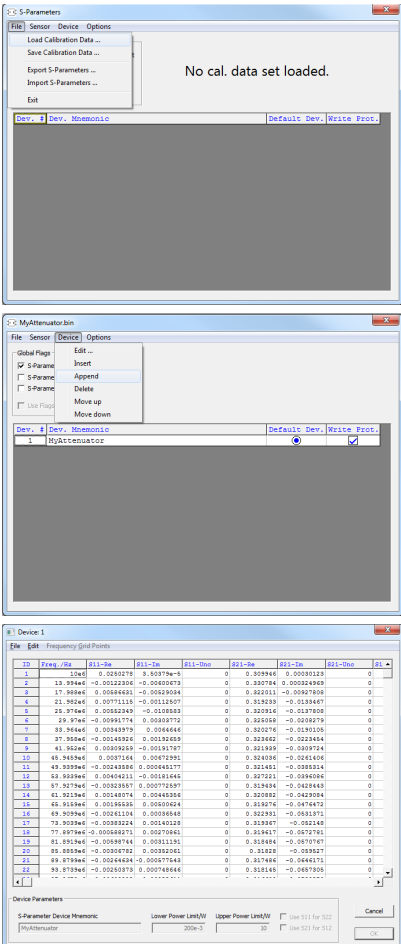
Sensor operation mode...	R&S NRPV dialogs...
<div><p>... with the attenuator supplied in the delivery</p></div>	<div><p>The application automatically recognizes the activated S-Parameter device in the sensor and activates the "S-Parameter Correction".</p></div>
<div><p>... without the supplied attenuator</p></div>	<div><p>If the supplied attenuator is not connected, you must disable "S-Parameter Correction" in the "Channel Settings" dialog.</p><p>To disable or enable the S-Parameter correction, select "Configure > Channel Settings".</p><p>Alternatively, you can change the default setting of the sensor by disabling "S-Parameter Correction" via the S-Parameters tool included in the R&S NRP Toolkit.</p></div>

Table 5-2: Operating an R&S NRP18S-xx with a different attenuator

Sensor operation mode...	R&S S-Parameters tool dialogs...
 <p>... with a different S-parameter device not supplied in the delivery</p>	<p>Load the S-Parameter device describing the attenuator into the sensor using the S-Parameters tool included in the R&S NRP Toolkit, see Chapter 8.8.4.5, "Using the S-Parameters Tool", on page 91.</p>  <p>Activate the "S-Parameter Correction" in the R&S NRPV "Channel Settings" dialog.</p>

Setup

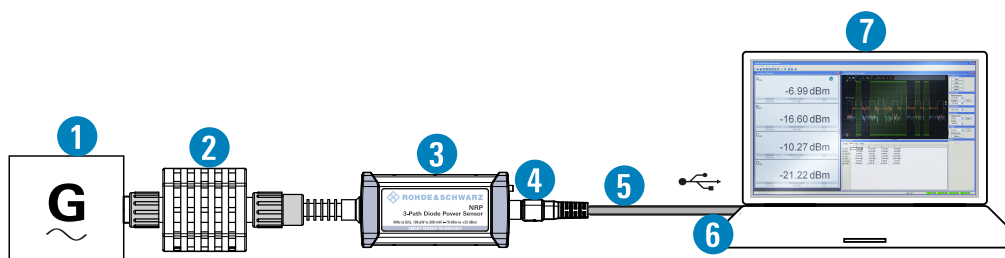


Figure 5-1: Setup with an R&S NRPV

- 1 = Signal source
- 2 = Attenuator
- 3 = R&S NRP18S-xx power sensor
- 4 = Host interface connector
- 5 = R&S NRP-ZKU cable
- 6 = USB connector
- 7 = Computer with installed R&S NRPV

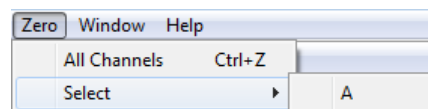


Incorrectly connecting/disconnecting the R&S NRP18S-xx power sensors can damage the power sensors or lead to erroneous results.

Ensure that you connect/disconnect your power sensor as described in [Chapter 3, "Preparing for Use"](#), on page 9.

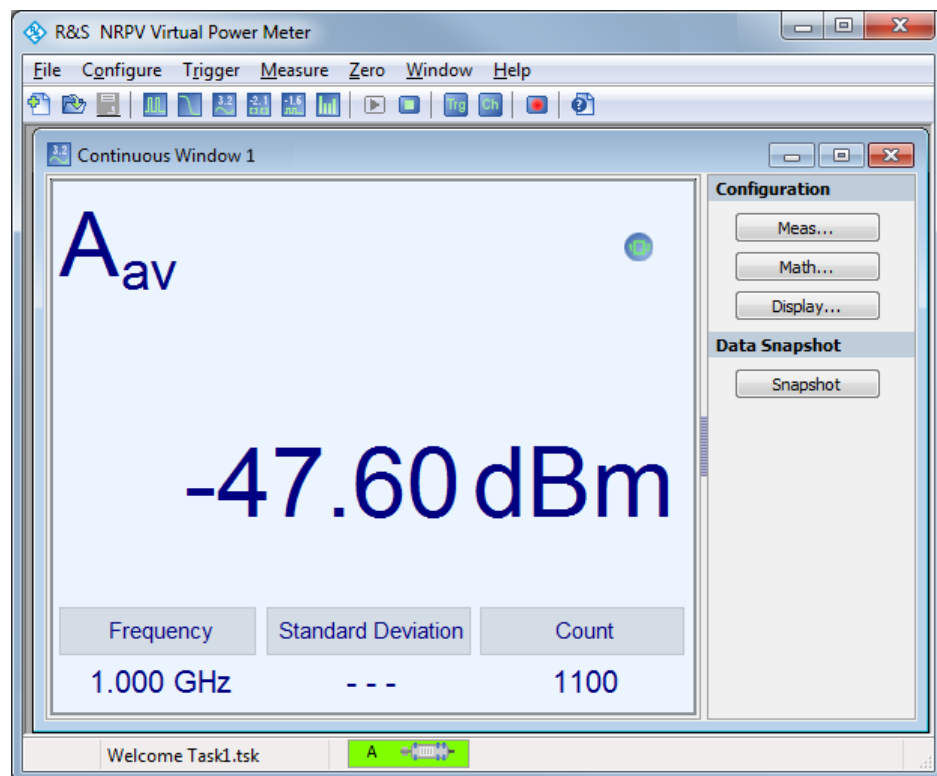
Starting a measurement

1. Connect the power sensor to the computer as shown in [Figure 5-1](#).
For a detailed description, refer to [Chapter 3.6.1.1, "Simple USB Connection"](#), on page 12.
2. Start the R&S NRPV.
3. Execute zeroing:
Note: Turn off all measurement signals before zeroing. An active measurement signal during zeroing causes an error.
 - a) Switch off the measurement signal.
 - b) Select "Zero > Select > A" (channel short name).



Zeroing takes several seconds. During zeroing, a message shows the progress. After completion, the message reports either success or an error ("Success" / "Failed").

4. Switch on the test signal of the signal source.
5. To start a continuous measurement, select "Measure > Continuous".
The "Continuous" measurement window appears. It shows the measurement results numerically, and the control panel for accessing further dialogs with parameters for measurement, evaluation and display.



For a detailed description on how to measure in this setup, refer to the operating manual of the R&S NRPV.

5.4 R&S Power Viewer

The R&S Power Viewer is software that simplifies many measurement tasks. It is provided on your documentation CD-ROM and on the Rohde & Schwarz website as a separate standalone installation package.

Required equipment

- R&S NRP18S-xx power sensor
- R&S NRP-ZKU cable or an R&S NRP-Z5 sensor hub and an R&S NRP-ZK6 cable to connect the sensor to the computer
- Computer with installed:
 - R&S NRP Toolkit V 4.17 or higher
 - R&S Power Viewer version 9.2 or higher (refer to the operating manual of the R&S Power Viewer for a description of the installation process)



If you want to use an android device like a tablet or a smartphone, use the R&S Power Viewer Mobile. For details, see [Chapter 5.5, "R&S Power Viewer Mobile"](#), on page 29.

Table 5-3: Operating an R&S NRP18S-xx with or without attenuator


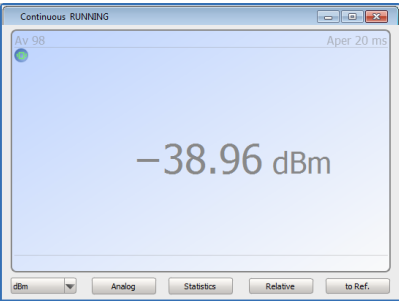

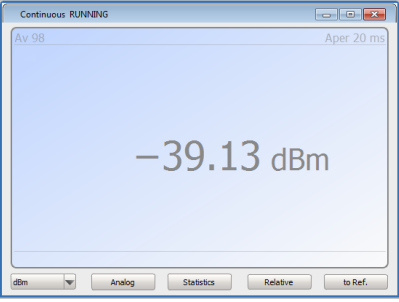
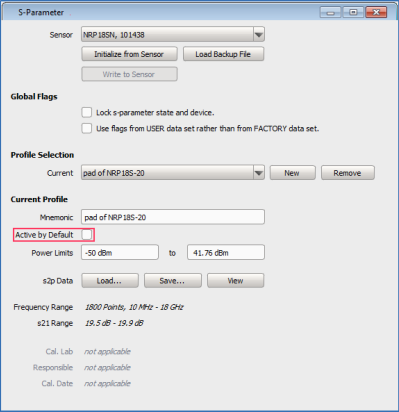

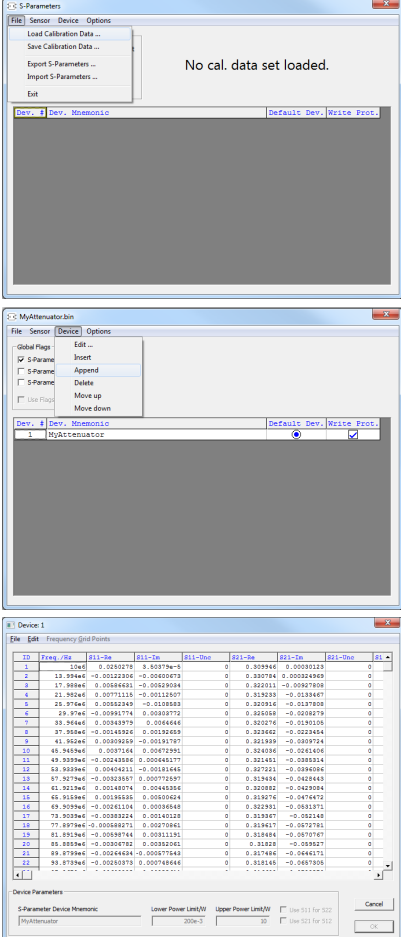
Sensor operation mode...	R&S Power Viewer dialogs...
<div><p>... with the attenuator supplied in the delivery</p></div>	<div><p>The application automatically recognizes the activated S-parameter device in the sensor and activates the "S-Parameter" correction.</p></div>
<div><p>... without the supplied attenuator</p></div>	<div><p>If the supplied attenuator is not connected, you must disable S-Parameter correction in the "S-Parameter" settings dialog.</p><p>To disable or enable the S-Parameter correction, select "Sensor > S-Parameter".</p><p>Alternatively, you can change the default setting of the sensor by disabling "S-Parameter Correction" via the S-parameters tool included in the R&S NRP Toolkit.</p></div>

Table 5-4: Operating an R&S NRP18S-xx with a different attenuator

Sensor operation mode...	S-Parameters tool dialogs...
 <p>... with a different S-parameter device not supplied in the delivery</p>	<p>Load the S-Parameter device describing the attenuator into the sensor using the S-Parameters tool included in the R&S NRP Toolkit, see Chapter 8.8.4.5, "Using the S-Parameters Tool", on page 91.</p>  <p>Activate the "S-Parameter Correction" in the R&S Power Viewer "S-Parameters" dialog.</p>

Setup

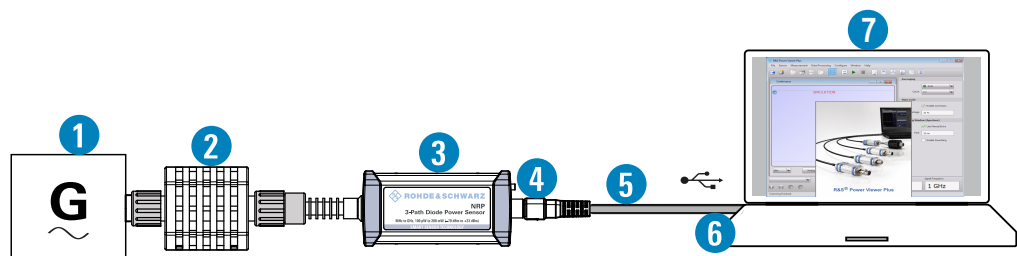


Figure 5-2: Setup with the R&S Power Viewer

- 1 = Signal source
- 2 = Attenuator
- 3 = R&S NRP18S-xx power sensor
- 4 = Host interface connector
- 5 = R&S NRP-ZKU cable
- 6 = USB connector
- 7 = Computer with installed R&S Power Viewer

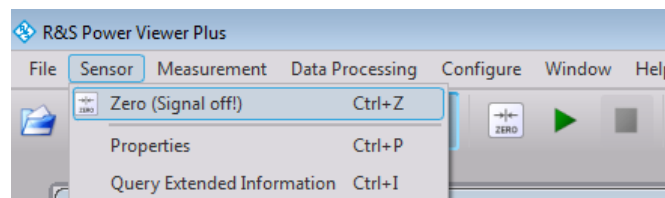


Incorrectly connecting/disconnecting the R&S NRP18S-xx power sensors can damage the power sensors or lead to erroneous results.

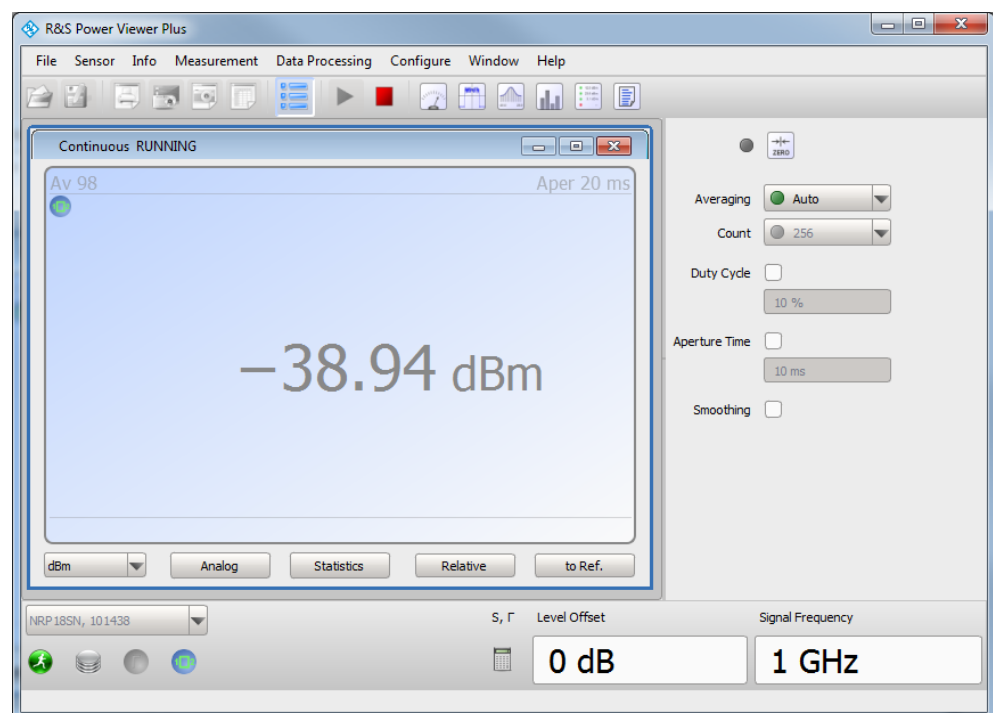
Ensure that you connect/disconnect your power sensor as described in [Chapter 3, "Preparing for Use"](#), on page 9.

Starting a measurement

1. Connect the cables as shown in [Figure 5-2](#).
For a detailed description, refer to [Chapter 3.6.1.1, "Simple USB Connection"](#), on page 12.
2. Start the R&S Power Viewer.
3. Execute zeroing:
Note: Turn off all measurement power signals before zeroing. An active measurement signal during zeroing causes an error.
 - a) Switch off the measurement signal.
 - b) Select "Sensor > Zero (Signal off)".



4. Switch on the test signal of the signal source.
5. For a continuous average measurement, select "Measurement > Continuous".
The "Continuous" measurement window appears. It shows the measurement results numerically and some parameters that can be configured.
6. To start the measurement press "Measurement > Start".
The measurement result is shown in the "Continuous" measurement window.



For a detailed description of how to measure in this setup, refer to the operating manual of your R&S Power Viewer. The manual is installed automatically during the installation of the R&S Power Viewer.

5.5 R&S Power Viewer Mobile

The R&S Power Viewer Mobile extends the functionality of the R&S Power Viewer to Android-based devices, such as a smartphone and tablets.

You can download the R&S Power Viewer Mobile free of charge from the Google Play Store.

The 1MA215 "Using R&S®NRP Series Power Sensors with Android™ Handheld Devices" application note gives a detailed description on installation and features of the R&S Power Viewer Mobile. The application note is provided on the documentation CD-ROM.

5.6 R&S NRX

In a measurement, the R&S NRX uses all sensor-dependent measurement functions and displays the results. Thus, you can configure both the measurement and the power sensor.

Required equipment

- R&S NRP18S-xx power sensor
- R&S NRP-ZK8 to connect the sensor to the R&S NRX
- R&S NRX

Setup

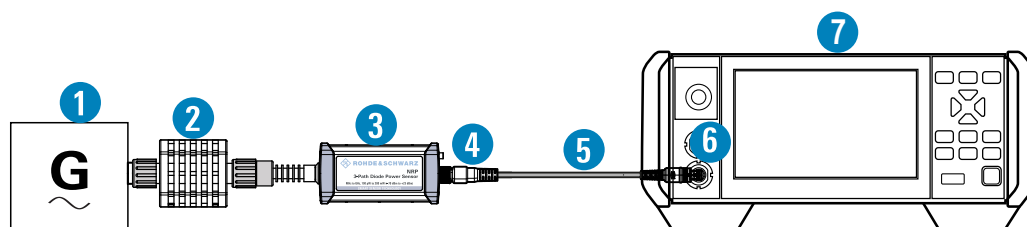


Figure 5-3: Setup with an R&S NRX base unit

- 1 = Signal source
- 2 = Attenuator
- 3 = R&S NRP18S-xx power sensor
- 4 = Host interface connector
- 5 = R&S NRP-ZK8
- 6 = Sensor input connector of the R&S NRX
- 7 = R&S NRX base unit



Incorrectly connecting/disconnecting the R&S NRP18S-xx power sensors can damage the power sensors or lead to erroneous results.

Ensure that you connect/disconnect your power sensor as described in [Chapter 3, "Preparing for Use"](#), on page 9.

Starting a measurement

1. Preset the R&S NRX and the connected R&S power sensors.
 - a) Press the [Preset] key.
 - b) Tap "Preset".
All parameters are set to their defaults.
2. If measuring in zero-IF mode (RBW > 40 MHz), consider to zero the power sensor:

Note: Turn off all measurement signals before zeroing. An active measurement signal during zeroing causes an error.

 - a) Switch off the power of the signal source.
 - b) Press the [Zero] key of the R&S NRX.
 - c) Tap "Zero All Sensors".
3. Configure the measurement.
 - a) In the "Measurement Settings" dialog, select the "Measurement Type", for example "Continuous Average".
 - b) Tap "Quick Setup" > "Auto Set".
4. Switch on the signal source.

The measurement starts, and the result is displayed in dBm.

5. If necessary, perform further settings.

For a detailed description of how to measure in this setup, refer to the user manual of the R&S NRX.

5.7 R&S NRP2

With the R&S NRP18S-xx power sensors and an R&S NRP2, you can measure power with up to four power sensors simultaneously. All sensor-dependent measurement functions can be used and the results can be displayed in parallel.

Required equipment

- R&S NRP18S-xx power sensor
- R&S NRP-ZK6 cable to connect the sensor to the R&S NRP2
- R&S NRP2 base unit with FW version 7.11 or higher

Table 5-5: Offset settings for operating an R&S NRP18S-xx with or without attenuator


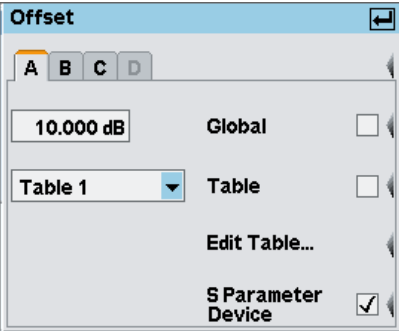

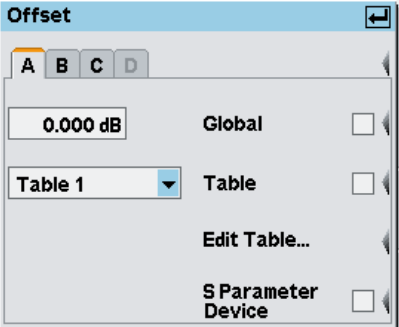

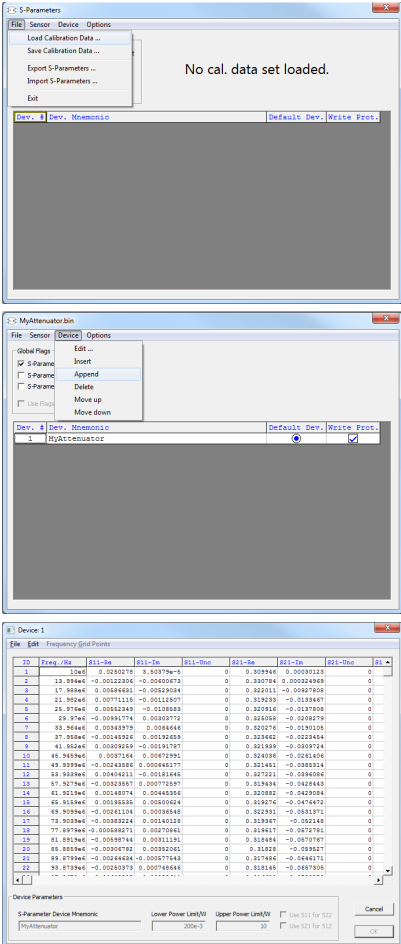
Sensor operation mode...	R&S NRP2 dialogs...
 <p>... with the attenuator supplied in the delivery</p>	 <p>"S Parameter Device" is automatically activated for correction, when you switch on the R&S NRP2 or plug in the sensor.</p>
 <p>... without the supplied attenuator</p>	 <p>Disable "S Parameter Device" each time when switching on the R&S NRP2 or you plug in the sensor.</p> <p>Alternatively, you can change the default setting of the sensor by disabling "S-Parameter Correction" via the S-Parameters tool included in the R&S NRP Toolkit.</p>

Table 5-6: Offset settings for operating an R&S NRP18S-xx with a different attenuator

Sensor operation mode...	S-Parameters tool dialogs...
 <p>... with a different S-parameter device not supplied in the delivery</p>	<p>Since the R&S NRP2 does not support selecting one out of several S-Parameter devices, you must activate the correct device via the S-Parameters tool included in the R&S NRP Toolkit, see Chapter 8.8.4.5, "Using the S-Parameters Tool", on page 91.</p>  <p>Activate the "S-Parameter Device" in the R&S NRP2 "Offset" dialog.</p>

Setup

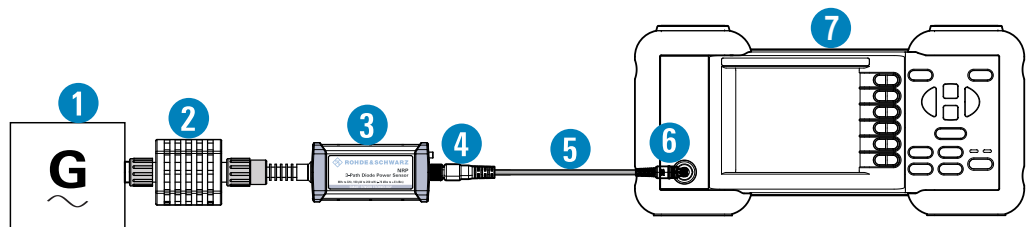


Figure 5-4: Setup with an R&S NRP2 base unit

- 1 = Signal source
- 2 = Attenuator
- 3 = R&S NRP18S-xx power sensor

- 4 = Host interface connector
- 5 = R&S NRP-ZK6 cable
- 6 = Sensor input connector of the R&S NRP2
- 7 = R&S NRP2 base unit



Incorrectly connecting/disconnecting the R&S NRP18S-xx power sensors can damage the power sensors or lead to erroneous results.

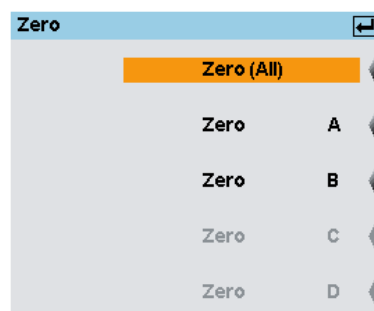
Ensure that you connect/disconnect your power sensor as described in [Chapter 3, "Preparing for Use"](#), on page 9.

Starting a measurement

1. Connect the cables as shown in [Figure 5-4](#):
 - a) Connect the R&S NRP-ZK6 cable to the host interface connector of the sensor.
 - b) Connect the R&S NRP-ZK6 cable to a sensor input connector of the R&S NRP2.
 - c) Connect the [RF] connector of the power sensor to the signal source.
2. Preset the R&S NRP2.
 - a) Press the [(PRE)SET] hardkey.
The "File" menu appears.
 - b) Press the [(PRE)SET] hardkey again or press the "Preset" softkey.
All parameters are set to their defaults, even when in inactive operating modes.
3. Execute zeroing:

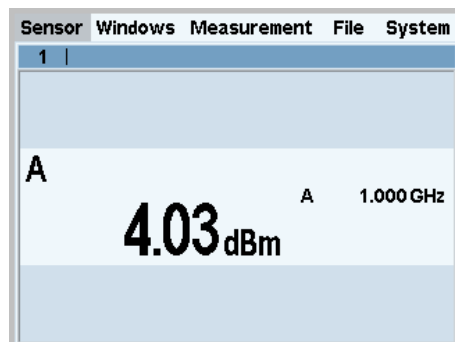
Note: Turn off all measurement signals before zeroing. An active measurement signal during zeroing causes an error.

 - a) Switch off the power of the signal source.
 - b) Press the [ZERO] hardkey of the R&S NRP2.
The "Zero" dialog box is displayed.



- c) Press the [ZERO] hardkey again to perform zeroing of all connected sensor channels ("Zero (All)") or press the appropriate softkey to select a specific sensor for zeroing.
4. Press the [FREQ] hardkey and enter the carrier frequency of the applied signal if the specified measurement accuracy is to be reached.
5. Switch on the signal source.

The result window indicates the result (in dBm) obtained with sensor A.



6. If necessary, perform further settings.

For a detailed description of how to measure in this setup, refer to the operating manual of your R&S NRP2.

6 Firmware Update

- [Hardware and Software Requirements](#).....35
- [Updating the Firmware](#).....35

6.1 Hardware and Software Requirements

For performing a firmware update, the system requirements are as follows:

- Connectors and cables for establishing a connection to the computer
See [Chapter 3.6.1, "Computer"](#), on page 12.
- Rohde & Schwarz update file (*.rsu) for the power sensor
Download the most recent firmware version from the Rohde & Schwarz homepage on the Internet, since the CD-ROM accompanying the power sensor contains the firmware dating from the time of delivery. The latest firmware update files are available at:
www.rohde-schwarz.com/en/firmware/nrp_s_sn/
If the *.rsu file is packed in a *.zip archive, extract it before updating.
- If you use the Firmware Update for NRP Family program, further requirements are essential. See ["Checking the prerequisites"](#) on page 36.

6.2 Updating the Firmware

NOTICE

Risk of faulty firmware

Disconnecting the power supply while an update is in progress can lead to missing or faulty firmware.

Take special care not to disconnect the power supply while the update is in progress. Interrupting the power supply during the firmware update most likely leads to an unusable power sensor that needs to be sent in for maintenance.

You can use the following methods to update the firmware installed on the power sensor:

- [Using the Firmware Update for NRP Family Program](#).....35
- [Using Remote Control](#).....37

6.2.1 Using the Firmware Update for NRP Family Program

Firmware Update for NRP Family is part of the R&S NRP Toolkit. See also [Chapter 5.1, "R&S NRP Toolkit"](#), on page 18.

Checking the prerequisites

1. Ensure that a recent VISA software is installed on the computer. You can perform a firmware update with Firmware Update for NRP Family only if the power sensor is recognized as a VISA device.
2. Ensure that the R&S NRP Toolkit for Windows is installed on the computer. See [Chapter 5.1, "R&S NRP Toolkit"](#), on page 18.

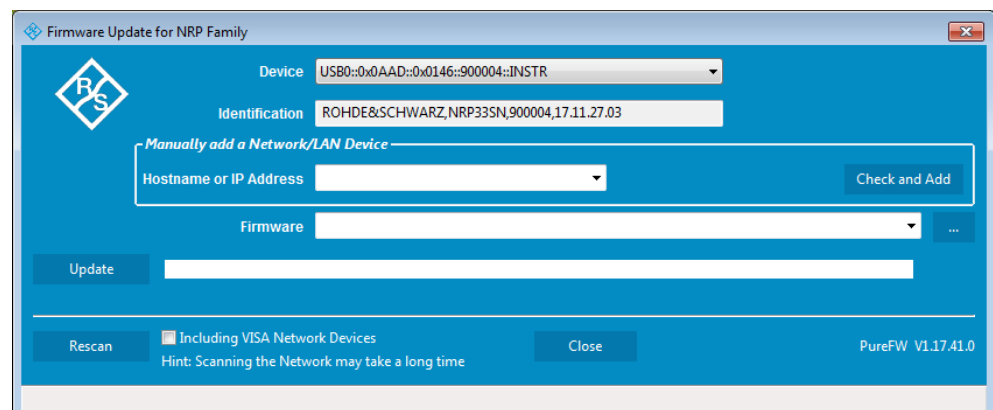
Firmware Update for NRP Family

A firmware update can take up to 5 minutes. Ensure that the update is not interrupted.

1. Check the prerequisites, see ["Checking the prerequisites"](#) on page 36.
2. Connect the power sensor to the computer as described in [Chapter 3.6.1, "Computer"](#), on page 12.
3. Start the Firmware Update for NRP Family program:
"Start" menu > "NRP-Toolkit" > "Firmware Update".

The program automatically starts scanning for Rohde & Schwarz power sensors connected via USB.

When the scan is completed, all recognized power sensors are listed under "Device".

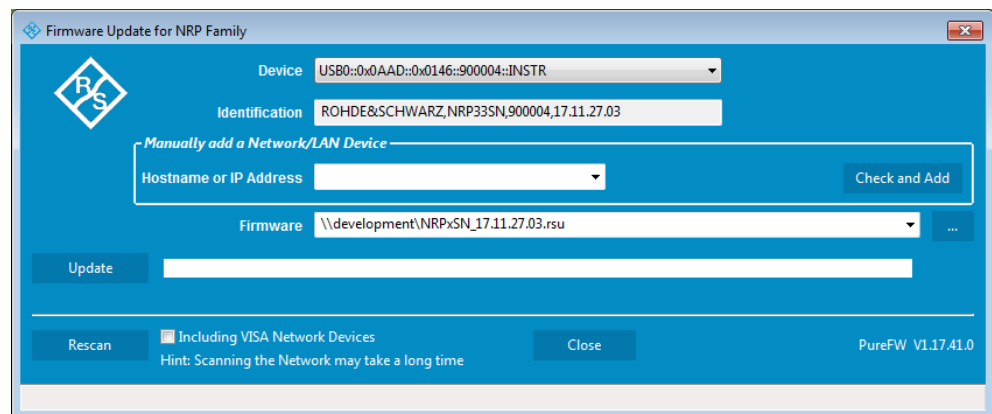


4. If the sensor you want to update is not listed, perform one of the following actions:
 - a) Click "Rescan" to search for attached sensors.
 - b) Check whether all necessary drivers are installed on the computer.
For example, if the VISA library is not installed on the computer, no VISA power sensor is accessible.
See also ["Troubleshooting"](#) on page 37.
5. Under "Device", select the sensor you want to update.
Note: The "Hostname or IP Address" field is not used during this procedure. Therefore, leave it empty.
6. Under "Firmware", enter the full path and filename of the update file, or press the browse button next to the field. New firmware for the Rohde & Schwarz power sensors generally has an *.rsu (Rohde & Schwarz update) extension.

7. Click "Update".

During the update process, a progress bar is displayed. The update sequence can take a couple of minutes, depending on the sensor model and the size of the selected file.

8. Check if the update was successful. The firmware version in the "Identification" field must match the version you selected in the "Firmware" field.



Troubleshooting

You do not find the sensor in the list of sensors provided by Firmware Update for NRP Family.

The driver assigned to the sensor is the legacy driver.

- Install a recent VISA software.

The power sensor is highlighted by a yellow exclamation mark in the Windows device manager.

Windows tries in vain to find a USB driver for the power sensor.

- Install a recent VISA software.

For further information, see [Chapter 11.3, "Problems during a Firmware Update"](#), on page 153.

6.2.2 Using Remote Control

If you want to integrate a firmware update function in an application, use `SYSTem:FWUPdate` on page 109.

Example:

You want to update your R&S NRP18S-10 with the `nrp18s_10_FW_15.02.12.01.rsu` file. This file has a size of 10242884 bytes.

To send the file to the sensor for updating the firmware, your application has to assemble a memory block containing:

```
SYST:FWUP <block_data>
```

The `<block_data>` are definite length arbitrary block data as described in [SYSTem:FWUPdate](#) on page 109.

The size of the file is 10242884. This number has 8 digits. Thus, the `<block_data>` consist of the following:

- `#`
- `8`
How many digits follow to specify the file size.
- `10242884`
Number that specifies the file size.
- `<file_contents>`
Contents of the `*.rsu` file, byte-by-byte
- `0x0a`
Delimiter

In this example, you write exactly 10242905 bytes to the power sensor, for example by using a `viWrite()` function.

The 10242905 bytes result from the values of the list above:

$$9 + 1 + 1 + 1 + 8 + 10242884 + 1$$

In a (pseudo) string notation, the memory block looks as follows:

```
SYST:FWUP #810242884<file_contents>0x0a,
```

7 Replacing an R&S NRP-Zxx with an R&S NRP18S-xx

The new R&S NRP18S-xx sensors are compatible with the R&S NRP-Zxx series of power sensors.

New power sensor	Replaces this sensor
R&S NRP18S-10	R&S NRP-Z22
R&S NRP18S-20	R&S NRP-Z23
R&S NRP18S-25	R&S NRP-Z24



To use the new power sensors, it can be required to update the drivers. For computer-based software applications (R&S NRPV and R&S Power Viewer), install latest R&S NRP Toolkit (V 4.17 or higher).

For using the sensors with R&S NRP2, signal generators, spectrum analyzers or other Rohde & Schwarz instruments, install the latest firmware version.

7.1 Most Important Differences

The new and the old sensors are compatible as far as possible. However, there are some differences:

- The state of the sensors is indicated by an LED, see [Chapter 4.4, "Status LED"](#), on page 17.
- After connecting the R&S NRP18S-xx sensors, the first measurement can be available after 7 seconds (R&S NRP-Zxx: 4 seconds).

7.2 Prerequisites

R&S NRP Toolkit

Install the R&S NRP Toolkit V 4.17 or higher, see [Chapter 5.1, "R&S NRP Toolkit"](#), on page 18.

The new version of the R&S NRP Toolkit is compatible with both the R&S NRP-Zxx and the R&S NRP18S-xx so that its installation does not affect the usage of the R&S NRP-Z power sensors.

After the new version of the R&S NRP Toolkit is installed, you can connect the R&S NRP18S-xx power sensor to the computer and use it with Rohde & Schwarz software applications or your own programs.

Software applications and firmware

Software/firmware	Prerequisites
R&S NRPV	See the release notes and the manual of the R&S NRPV.
R&S Power Viewer	See the release notes and the manual of the R&S Power Viewer.
R&S NRP2	Install firmware version 7.11 or higher.
R&S signal generators, spectrum analyzers or other instruments	Install the latest firmware version (released December 2014 or later).

8 Remote Control Commands

In the following sections, all commands implemented in the sensor are listed according to the command system and then described in detail. For the most part, the notation used complies with SCPI specifications.

8.1 Conventions Used in SCPI Command Descriptions

Note the following conventions 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 NRP18S-xx 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.

For further information on units, see also ["Units"](#) on page 132.

8.2 Notations

For a detailed description of SCPI notations, see [Chapter 10, "Remote Control Basics"](#), on page 128.

Numeric suffixes <n>

If a command can be applied to multiple instances of an object, e.g. specific sensors, the required instances can be specified by a suffix added to the command. Numeric suffixes are indicated by angular brackets (<1...4>, <n>, <l>) and are replaced by a single value in the command. Entries without a suffix are interpreted as having the suffix 1.

Optional keywords []

Some command systems permit certain keywords to be inserted into the header or omitted. These keywords are marked by square brackets in the description. The instrument must recognize the long command to comply with the SCPI standard. Some commands are considerably shortened by these optional mnemonics.

Therefore, not only is there a short and a long form for the commands (distinguished here by uppercase and lowercase letters) but also a short form which is created by omitting optional keywords.

Example:

Command [SENSe<Sensor>:] [POWer:] [AVG:] SMOothing:STATe 1 can be written as:

```
SENSe1:POWer:AVG:SMOothing:STATe 1
SENS:POW:AVG:SMO:STAT 1
SENSe:POWer:SMOothing:STATe 1
SENSe:SMOothing:STATe 1
SMOothing:STATe 1
SMO:STAT 1
```

Parameters

Parameters must be separated from the header by a "white space". If several parameters are specified in a command, they are separated by a comma (,). For a description of the parameter types, refer to [Chapter 10.2.3, "SCPI Parameters"](#), on page 131.

Example:

Definition: [SENSe<Sensor>:] AVERage:COUNT:AUtO:NSRatio <n timer>

Command: AVER:COUN:AUtO:NSR 0.01

Special characters | and { }

	<p>A vertical bar in parameter definitions indicates alternative possibilities in the sense of "or". The effect of the command differs, depending on which parameter is used.</p> <p>Example:</p> <p>Definition: INITiate:CONTinuous ON OFF</p> <p>Command INITiate:CONTinuous ON starts the measurements</p> <p>Command INITiate:CONTinuous OFF stops the measurements</p>
{ }	Parameters in braces may be included in the command once, several times or not at all.

8.3 Common Commands

The common commands are taken from the IEEE 488.2 (IEC 625–2) standard. The headers of these commands consist of an asterisk * followed by three letters.

*CLS.....	43
*ESE.....	44
*ESR?.....	44
*IDN?.....	44
*IST?.....	44
*OPC.....	44
*OPT?.....	45
*PRE.....	45
*RCL.....	45
*RST.....	45
*SAV.....	45
*SRE.....	46
*STB?.....	46
*TRG.....	46
*TST?.....	46
*WAI.....	47

*CLS

CLear Status

Resets the:

- Status byte (STB)
- Standard event register (ESR)
- EVENT part of the QUESTionable and the OPERation register
- Error/event queue

The command does not alter the ENABle and TRANsition parts of the registers.

Usage: Event

***ESE <register>**

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:

<register>	Range:	0 to 255
	*RST:	0

***ESR?**

Event Status Read query

Returns the contents of the event status register in decimal form (0 to 255) and subsequently sets the register to zero.

Usage: Query only

***IDN?**

IDeNtification query

Returns a string with information on the sensor's identity (device identification code). In addition, the version number of the installed firmware is indicated.

Usage: Query only

***IST?**

Individual SStatus query

Returns the current value of the IST flag in decimal form. The IST flag is the status bit which is sent during a parallel poll.

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. *OPC must be sent at the end of a program message.

The query form returns a "1" when all previous commands have been processed. It is important that the read timeout is set sufficiently long.

Since *OPC? waits until all previous commands are executed, "1" is returned in all cases.

*OPC? basically functions like the *WAI command, but *WAI does not return a response.

*OPC? is preferred to *WAI because with *OPC?, the execution of commands can be queried from a controller program before new commands are sent. This prevents overflow of the input queue when too many commands are sent that cannot be executed.

Unlike *WAI, *OPC? must be sent at the end of a program message.

*OPT?

OPTion identification query

Returns a comma-separated list of installed options.

Usage: Query only

*PRE <register>

Parallel poll Register Enable

Sets the parallel poll enable register to the specified value or queries the current value.

Parameters:

<register>	Range:	0 to 255
	*RST:	0

*RCL <number>

ReCaLI

Calls the device state which has been stored with the *SAV command under the specified number.

Setting parameters:

<number>	Range:	0 to 9
	*RST:	0

Usage: Setting only

*RST

Reset

Sets the instrument to a defined default status. The default settings are indicated in the description of commands.

The command corresponds to the `SYSTem:PRESet` command.

Usage: Event

*SAV <number>

SAVe

Stores the current device state under the specified number. The storage numbers 0 to 9 are available.

Setting parameters:

<number> Range: 0 to 9
 *RST: 0

Usage: Setting only

***SRE <register>**

Service Request Enable

Sets the service request enable register to the specified value. This command determines under which conditions a service request is triggered.

Parameters:

<register> Range: 0 to 255
 *RST: 0

***STB?**

STatus Byte query

Returns the contents of the status byte in decimal form.

Usage: Query only

***TRG**

TRiGger

Triggers a measurement. This command is only valid if the power sensor is in the waiting for trigger state and the trigger source is set to `BUS`

See `TRIGger:SOURce` `BUS`.

Usage: Event

***TST?**

Selftest query

Triggers a self test of the instrument and outputs an error code in decimal form. 0 indicates that no errors have occurred.

Example: *TST?
 Query
 0
 Response: Passed

Example: *TST?
 Query
 1
 Response: Failed

Usage: Query only

***WAI**

WAI to continue

Prevents the execution of the subsequent commands until all preceding commands have been executed and all signals have settled.

Usage: Event

8.4 Preparing for the Measurement

Before starting a measurement, you need to do the following:

- [Selecting the Reference Source](#)..... 47
- [Selecting a Measurement Path](#)..... 48
- [Selecting a Measurement Mode](#)..... 48

8.4.1 Selecting the Reference Source

The `ROSCillator` subsystem contains commands for configuring the reference source.

[SENSe<Sensor>:]ROSCillator:SOURce <source>

Sets the source of the reference oscillator. Refers typically to a precision, stabilized time base.

Parameters:

<source>

INTernal | EXTernal | HOST

INTernal

Internal precision oscillator

EXTernal | HOST

External signal supplied at the host interface connector.

***RST:** If the sensor boots or reboots, the source is set to INTernal. If the sensor is reset, the source setting is kept unchanged.

Example: ROSC:SOUR INT

8.4.2 Selecting a Measurement Path

The **RANGe** subsystem contains commands for selection of a measurement path.

Remote commands:

[SENSe<Sensor>:]RANGe.....	48
[SENSe<Sensor>:]RANGe:AUTO.....	48
[SENSe<Sensor>:]RANGe:CLeVel.....	48

[SENSe<Sensor>:]RANGe <range>

Selects the active measurement path manually.

Parameters:

<range> The sensitivity of the paths differs.
 0 is the most sensitive path.
 2 is the most insensitive path.
 1 is the path with medium sensitivity.
 Range: 0 to 2
 *RST: 2

[SENSe<Sensor>:]RANGe:AUTO <state>

Enables automatic measurement path selection.

Parameters:

<state> *RST: ON

[SENSe<Sensor>:]RANGe:CLeVel <level>

Reduces the transition range between the measurement paths, 0 -> 1 and 1 -> 2, by the set value. Thus, you can improve the measurement accuracy for signals with a high peak-to-average ratio, since the headroom for modulation peaks becomes larger. However, the S/N ratio is reduced at the lower limits of the transition ranges.

Parameters:

<level> Range: -20.00 to 0.00
 *RST: 0.00
 Default unit: dB

8.4.3 Selecting a Measurement Mode

- Before starting a measurement, select the measurement mode using:

[SENSe<Sensor>:]FUNCTION

The following modes are available:

- Continuous average ("POWer:AVG"): After a trigger event, the power is integrated over a time interval.

See also [Chapter 8.7.1, "Continuous Average Measurement"](#), on page 69.

- Burst average ("POWer:BURSt:AVG"): The integration time of a measurement is not predefined but determined by the sensor with the aid of a burst detector. The start of a burst is detected when the measurement signal rises above a set trigger level. The measurement ends when the signal drops below a trigger threshold. See also [Chapter 8.7.2, "Burst Average Measurement"](#), on page 72.
- Timeslot average ("POWer:TSLot:AVG"): The power is simultaneously measured in up to 32 time slots. The measurement result is represented by a vector that can contain up to 32 indices and contains the power of a time slot at each index. See also [Chapter 8.7.3, "Timeslot Average Measurement"](#), on page 74.
- Trace ("XTIME:POWer"): A sequence of measurements is performed. See also [Chapter 8.7.4, "Trace Measurement"](#), on page 76.

8.5 Controlling the Measurement

The power sensor offers a bunch of possibilities to control the measurement:

- Do you want to start the measurement immediately after the initiate command or do you want to wait for a trigger event?
- Do you want to start a single measurement cycle or a sequence of measurement cycles?
- Do you want to output each new average value as a measurement result or do you want to bundle more measured values into one result?

8.5.1 Starting and Ending a Measurement

ABORT.....	49
INITiate:ALL.....	50
INITiate[:IMMEDIATE].....	50
INITiate:CONTinuous.....	50

ABORT

Immediately interrupts the current measurement. If the measurement has been started as a single measurement (INITiate[:IMMEDIATE]), the sensor goes into the idle state. However, if a continuous measurement is in progress (INITiate:CONTinuous ON), the trigger system of the sensor enters the waiting for trigger state, and if the trigger condition is met, a new measurement is immediately started.

See also [Chapter 8.5, "Controlling the Measurement"](#), on page 49.

Usage: Event

INITiate:ALL**INITiate[:IMMediate]**

Starts a single measurement cycle. The sensor changes from the idle state to the waiting for trigger state. As soon as the trigger condition is fulfilled, the sensor begins the measurement. Depending on the number of trigger events that are required, e.g. for averaging, the sensor enters the waiting for trigger state several times. Once the entire measurement is completed, a measurement result is available, and the sensor enters the idle state again.

Use the command only after the continuous measurement mode has been switched off (**INITiate:CONTinuous OFF**).

See also [Chapter 8.5, "Controlling the Measurement"](#), on page 49.

Example: See [Chapter 9.3, "Performing a Buffered Continuous Average Measurement"](#), on page 123.

Usage: Event

INITiate:CONTinuous <state>

Activates/deactivates the continuous measurement mode. In continuous measurement mode, the power sensor does not go into the idle state after a measurement has been completed, but immediately executes another measurement cycle.

See also [Chapter 8.5.2, "Triggering"](#), on page 50.

Parameters:

<state>

ON

Measurements are performed continuously. If a measurement is completed, the sensor does not return to the idle state but enters the waiting for trigger state again.

OFF

Ends the continuous measurement mode, and sets the sensor to the idle state.

*RST: OFF

Example: See [Chapter 9.3, "Performing a Buffered Continuous Average Measurement"](#), on page 123.

8.5.2 Triggering

In a basic continuous measurement, the measurement is started immediately after the initiate command, see also [Chapter 8.5.2.2, "Waiting for a Trigger Event"](#), on page 51. However, sometimes you want that the measurement starts only if a specific condition is fulfilled. For example, if a signal level is exceeded, or in certain time intervals. For these cases, you can define a trigger for the measurement.

Further information:

- [Chapter 8.5.5, "Configuring the Trigger"](#), on page 58

8.5.2.1 Trigger States

The power sensor has trigger states to define the exact start and stop time of a measurement and the sequence of a measurement cycle. The following states are defined:

- **Idle**
The power sensor performs no measurement. After powered on, the power sensor is in the idle state.
- **Waiting for trigger**
The power sensor waits for a trigger event that is defined by the trigger source. When the trigger event occurs, the power sensor enters the measuring state.
- **Measuring**
The power sensor is measuring data. It remains in this state during the measurement. When the measurement is completed, it exits this state immediately.

8.5.2.2 Waiting for a Trigger Event

Before a trigger can be executed, the power sensor has to be set to the waiting for trigger state. Depending on the required number of measurement cycles, you use one of the following commands:

- `INITiate:CONTInuous`
A new measurement cycle is started automatically after the previous one has been terminated.
- `INITiate[:IMMediate]`
The number of measurement cycles is restricted.
If `TRIGger:COUNt 1` is set, the command starts a single measurement cycle that renders one result. Every time you send this command, a new measurement cycle is started.
Otherwise, as many measurement cycles are performed as determined by the trigger count.

8.5.2.3 Trigger Sources

The possible trigger conditions and the execution of a trigger depend on the selected trigger mode and trigger source.

If the signal power exceeds or falls below a reference level set by the trigger level, the measurement is started after the defined delay time. Waiting for a trigger event can be skipped.

Trigger source	Description	Remote commands to initiate the measurement
"Hold"	Triggered by the remote command.	<code>TRIGger:IMMediate</code>
"Immediate"	Measures immediately, does not wait for trigger condition.	-
"Internal"	Uses the input signal as trigger signal.	<code>TRIGger:IMMediate</code>
"External 1"	Uses the digital input signal supplied using a differential pair in the 8-pin sensor cable.	<code>TRIGger:IMMediate</code>

Trigger source	Description	Remote commands to initiate the measurement
"External 2"	Uses the digital input signal supplied at the SMB connector.	TRIGger:IMMediate
"Bus"	Triggered by the remote command.	*TRG TRIGger:IMMediate

8.5.2.4 Dropout Time

The dropout time is useful when dealing with signals with several active slots, for example GSM signals, see [Figure 8-1](#). When measuring in sync with the signal, a trigger event is to be produced at A, but not at B or C.

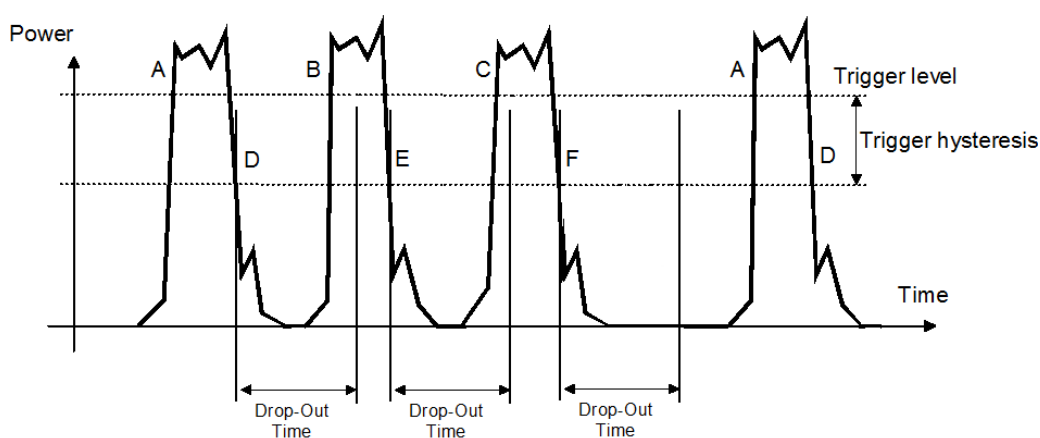


Figure 8-1: Significance of the dropout time

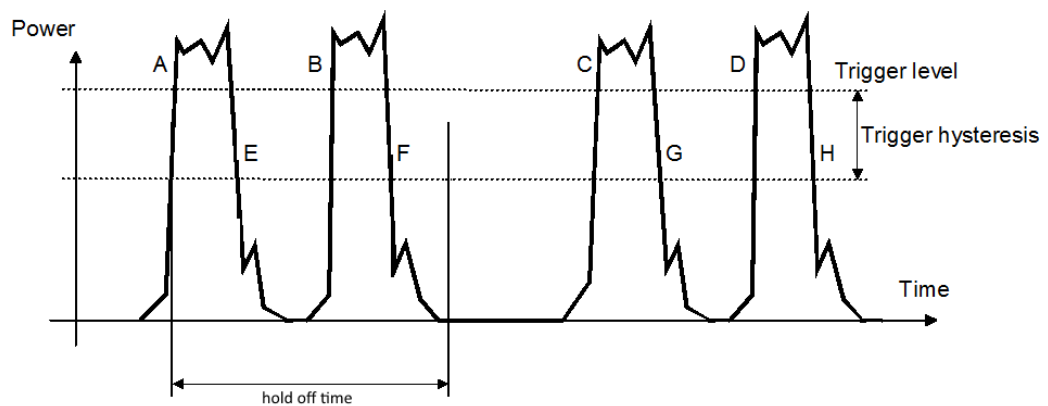
The RF power between the slots is below the threshold defined by the trigger level and the trigger hysteresis. Therefore, the trigger hysteresis alone cannot prevent triggering at B or at C. Therefore, set the dropout time greater than the time elapsed between points D and B and between E and C, but smaller than the time elapsed between F and A. Thus, you ensure that triggering takes place at A.

Because the mechanism associated with the dropout time is reactivated whenever the trigger threshold is crossed, you can obtain also unambiguous triggering for many complex signals.

If you use a hold-off time instead of a dropout time, you can obtain stable triggering conditions - regular triggering at the same point. But you cannot achieve exclusive triggering at A.

8.5.2.5 Hold-Off Time

During the hold-off time, a period after a trigger event, all trigger events are ignored.



8.5.3 Controlling the Measurement Results

The R&S NRP18S-xx can cope with the wide range of measurement scenarios with the help of the so-called "termination control". Depending on how fast your measurement results change, you can define, how the measurement results are output.

In continuous average mode, use `[SENSe<Sensor>:]AVERage:TCONtrol`.

In trace mode, use `[SENSe<Sensor>:]TRACe:AVERage:TCONtrol`.

Repeating termination control

Outputs a measurement result when the entire measurement has been completed. This means that the number of measurement cycle repetitions is equal to the set average count. If the average count is large, the measurement time can be very long.

Useful if you expect slow changes in the results, and you want to avoid outputting redundant data.

Moving termination control

Outputs intermediate values to facilitate early detection of changes in the measured quantity. This means that for each partial measurement, a new average value is output as a measurement result. Thus, the measurement result is a moving average of the last partial measurements. How many of the partial measurements are averaged is defined by the average count.

Useful if you want to detect trends in the result during the measurement.

8.5.4 Interplay of the Controlling Mechanisms

In the following examples, continuous measurement scenarios are used. But these scenarios apply also to single measurements. The only difference is that a single measurement is not repeated.

8.5.4.1 Continuous Average Mode

General settings for these examples:

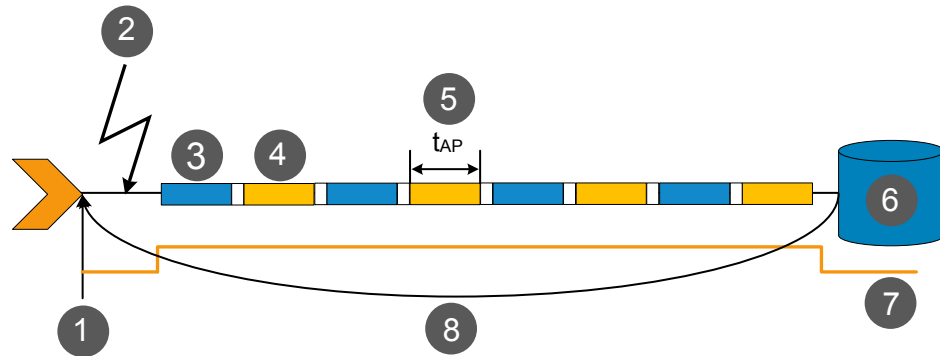
- `INITiate:CONTinuous ON`
- `[SENSe<Sensor>:]AVERage:COUNT 4`
- `[SENSe<Sensor>:]AVERage:COUNT:AUTO OFF`

Example: Repeating termination control

Further settings for this example:

- `[SENSe<Sensor>:]AVERage:TCONTROL REPEAT`

The measurement is started by the trigger event. Due to the chopper phases, one measurement lasts twice the defined aperture time. As defined by the average count, after 4 measurements, the result is averaged and available. During the whole measurement cycle, the trigger synchronization is high (`TRIGger:SYNC:STATe ON`).



- 1 = Start of the measurement cycle
- 2 = Trigger event
- 3 = Noninverted chopper phase
- 4 = Inverted chopper phase
- 5 = Duration of one aperture time ($1 \times t_{AP}$) \triangleq length of one chopper phase
- 6 = Measurement result
- 7 = Trigger synchronization
- 8 = Return to the start of the measurement cycle

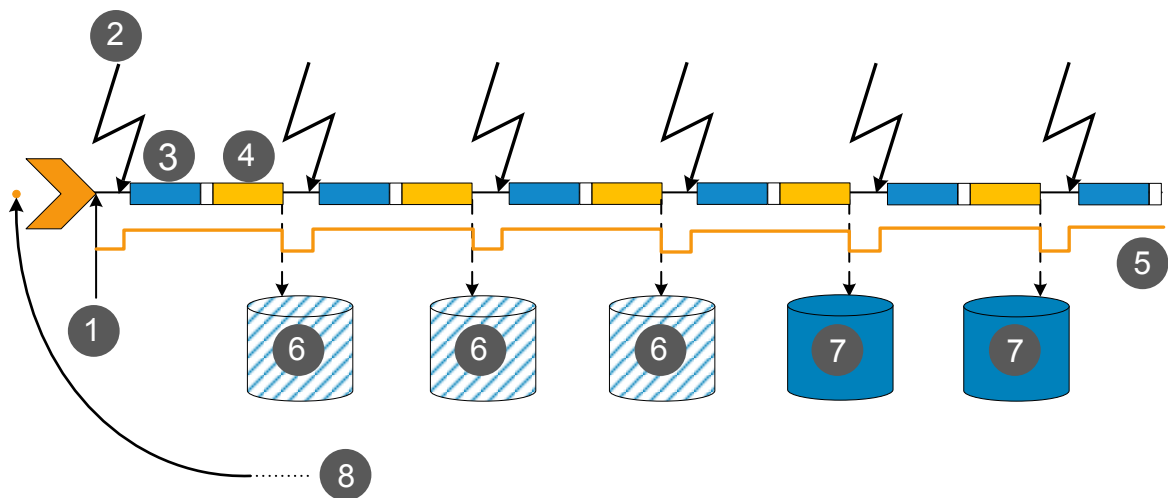
Example: Moving termination control

Further settings for this example:

- `[SENSe<Sensor>:]AVERage:TCONtrol MOVing`
- `TRIGger:COUNt 16`

Every measurement is started by a trigger event. Due to the chopper phases, one measurement lasts twice the defined aperture time. During each measurement, the trigger synchronization is high (`TRIGger:SYNC:STATE ON`). Every measurement provides a result. During the settling phase, the amount of the result is already correct, but the noise is higher. After 4 measurements, when the average count is reached, settled data are available.

When the trigger count is reached (`TRIGger:COUNt` on page 59), the power sensor returns to the idle state.



- 1 = Start of the measurement cycle
- 2 = Trigger event
- 3 = Noninverted chopper phase
- 4 = Inverted chopper phase
- 5 = Trigger synchronization
- 6 = Measurement result before average count is reached
- 7 = Averaged measurement result after average count is reached
- 8 = Return to idle state after trigger count (= 16 in this example) is reached

8.5.4.2 Trace Mode

General settings for the first two examples:

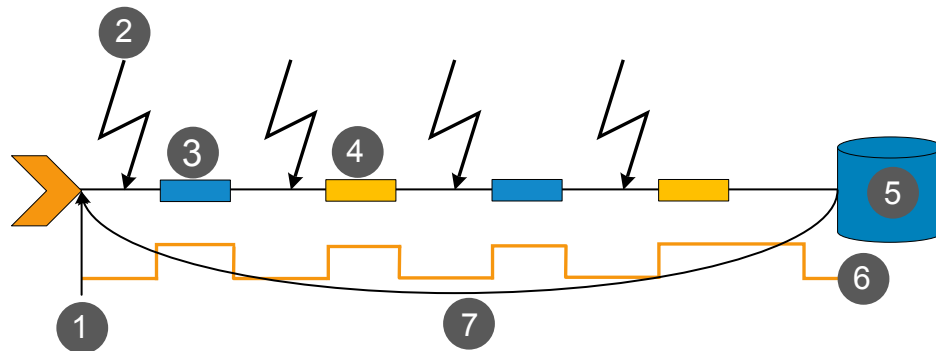
- `INITiate:CONTinuous ON`
- `[SENSe<Sensor>:]AVERage:COUNt 2`

Example: Repeating termination control

Further settings for this example:

- `[SENSe<Sensor>:]AVERage:TCONtrol REPEAT`

Every chopper phase is started by a trigger event and lasts the defined aperture time. During a chopper phase, the trigger synchronization is high (`TRIGger:SYNC:STATe ON`). After 2 chopper phases, 1 measurement is completed. As defined by the average count, after 2 measurements, the result is averaged and available.



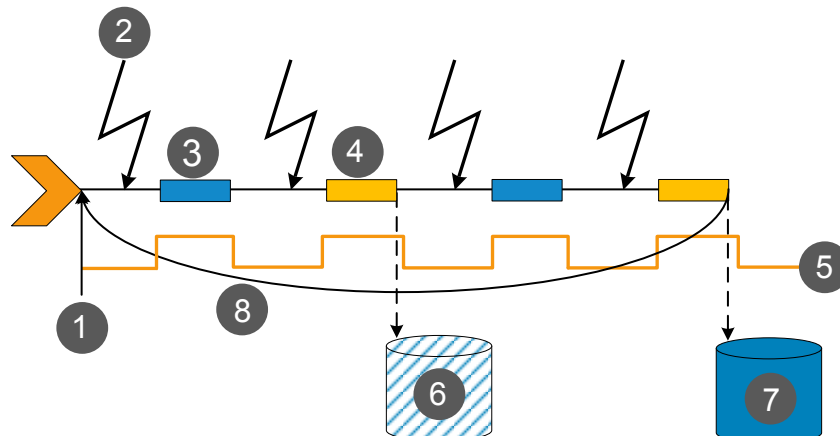
- 1 = Start of the measurement cycle
- 2 = Trigger event
- 3 = Noninverted chopper phase
- 4 = Inverted chopper phase
- 5 = Measurement result
- 6 = Trigger synchronization
- 7 = Return to the start of the measurement cycle

Example: Moving termination control

Further settings for this example:

- `[SENSe<Sensor>:]AVERage:TCONtrol MOVing`

Every chopper phase is started by a trigger event and lasts the defined aperture time. During a chopper phase, the trigger synchronization is high (`TRIGger:SYNC:STATe ON`). Every measurement provides a result. After 2 measurements, when the average count is reached, settled data are available.

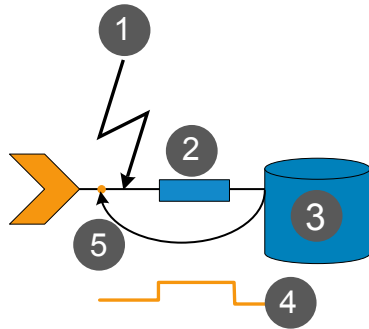


- 1 = Start of the measurement cycle
- 2 = Trigger event
- 3 = Noninverted chopper phase
- 4 = Inverted chopper phase
- 5 = Trigger synchronization
- 6 = Measurement result before average count is reached
- 7 = Averaged measurement result after average count is reached
- 8 = Return to the start of the measurement cycle

Example: Average count = 1

```
[SENSe<Sensor>:]AVERAge:COUNT 1
```

For average count 1, the setting of the termination control has no impact. In both cases, the measurement runs for the duration of one aperture time. Then, settled data are available, and the power sensor returns to the idle state.



- 1 = Trigger event
- 2 = Noninverted chopper phase
- 3 = Measurement result
- 4 = Trigger synchronization
- 5 = Return to idle state

8.5.5 Configuring the Trigger

Further information:

- [Chapter 8.5, "Controlling the Measurement"](#), on page 49

Remote commands:

TRIGger:ATRigger:DELay	59
TRIGger:ATRigger:EXECuted?	59
TRIGger:ATRigger[:STATe]	59
TRIGger:COUNt	59
TRIGger:DELay	60
TRIGger:DELay:AUTO	60
TRIGger:DTIME	60
TRIGger:EXTErnal<2...2>:IMPedance	60
TRIGger:HOLDoff	61
TRIGger:HYSTeresis	61
TRIGger:IMMediate	61
TRIGger:LEVel	61
TRIGger:LEVel:UNIT	62
TRIGger:MASTer:PORT	62
TRIGger:MASTer:STATe	62
TRIGger:SLOPe	63
TRIGger:SOURce	63
TRIGger:SYNC:PORT	63
TRIGger:SYNC:STATe	63

TRIGger:ATRigger:DElay <delay>

Effective only if **TRIGger:ATRigger[:STATe]** is set to ON.

Sets the delay between the artificial trigger event and the beginning of the actual measurement

Parameters:

<delay>	Range:	0.1 to 5.0
	*RST:	0.3
	Default unit:	Seconds

TRIGger:ATRigger:EXECuted?

Queries the number of measurements that were triggered automatically since **TRIGger:ATRigger[:STATe]** was set to ON.

In normal scalar measurements, this number can only be 0 or 1. If a buffered measurement was executed, this number indicates how many results in the returned array of measurement data were executed without a real trigger event.

Usage: Query only

TRIGger:ATRigger[:STATe] <state>

Controls the automatic trigger function. If enabled, an artificial trigger is generated if the delay time has elapsed after the measurement start and no trigger event has occurred.

The delay time is set using **TRIGger:ATRigger:DElay**.

Parameters:

<state>	*RST:	OFF
---------	-------	-----

TRIGger:COUNt <count>

Sets the number of measurement cycles to be performed when the measurement is started using **INITiate[:IMMediate]**.

This number equals the number of results that can be obtained from the sensor after a single measurement. As long as the defined number of measurements is not executed, the sensor automatically initiates another measurement internally when the current result is available.

This command is particularly useful in conjunction with buffered measurements. For example, to fill a buffer with a predefined size with measurements that have been triggered externally or by *TRG without having to start the measurement multiple times.

Parameters:

<count>	Range:	1 to 8192
	*RST:	1

Example: See [Chapter 9.3, "Performing a Buffered Continuous Average Measurement"](#), on page 123.

TRIGger:DElay <delay>

Sets the delay between the trigger event and the beginning of the actual measurement (integration).

Parameters:

<delay> Range: -5.0 to 10.0
 *RST: 0.0
 Default unit: s

TRIGger:DElay:AUTO <state>

If **TRIGger:DElay:AUTO** ON is set, no measurement is started until the power sensor has settled. For this purpose, the delay value is automatically determined.

If a longer period is set using **TRIGger:DElay**, the automatically determined delay is ignored.

Parameters:

<state> *RST: OFF

TRIGger:DTIME <dropout_time>

Sets the dropout time for the internal trigger source. During this time, the signal power must exceed (negative trigger slope) or undercut (positive trigger slope) the level defined by the trigger level and trigger hysteresis. At least, this time must elapse before triggering can occur again.

See [Chapter 8.5.2.4, "Dropout Time"](#), on page 52.

Parameters:

<dropout_time> Range: 0.00 to 10.00
 *RST: 0.00
 Default unit: s

TRIGger:EXternal<2...2>:IMPedance <impedance>

Effective only if **TRIGger:SOURce** EXternal2 is set.

Sets termination resistance of the second external trigger input. Choose the setting that fits the impedance of the trigger source to minimize reflections on the trigger signals.

Suffix:

<2...2> 2

Parameters:

<impedance> HIGH | LOW
 HIGH
 ~10 kΩ
 LOW
 50 kΩ

*RST: HIGH

TRIGger:HOLDOff <holdoff>

Sets the hold-off time, see [Chapter 8.5.2.5, "Hold-Off Time"](#), on page 52 .

Parameters:

<holdoff>	Range:	0.00 to 10.00
	*RST:	0.00
	Default unit:	Seconds

TRIGger:HYSTeresis <hysteresis>

Sets the hysteresis. A trigger event occurs, if the trigger level:

- Falls below the set value on a rising slope.
- Rises above the set value on a falling slope

Thus, you can use this setting to eliminate the effects of noise in the signal for the edge detector of the trigger system.

Parameters:

<hysteresis>	Range:	0.00 to 10.00
	*RST:	0.00
	Default unit:	DB

TRIGger:IMMediate

Causes a generic trigger event. The power sensor leaves the waiting for trigger state immediately, irrespective of the trigger source and the trigger delay, and starts the measurement.

This command is the only way to start a measurement if the trigger source is set to hold ([TRIGger:SOURce](#) HOLD). Only one measurement cycle is executed, irrespective of the averaging factor.

Usage: Event

TRIGger:LEVel <level>

Effective only if [TRIGger:SOURce](#) INTERNAL.

Sets the trigger threshold for internal triggering derived from the test signal.

If you enter a value without unit, the unit is defined by [TRIGger:LEVel:UNIT](#).

If an S-parameter device is active and/or if a mounted component with a global offset in front of the sensor is considered, the currently effective trigger threshold and a trigger threshold to be input are referenced to the appropriately shifted sensor data. If the S-parameter device and/or the offset correction are disabled, the trigger threshold and its input limits are adjusted as necessary.

Parameters:

<level> Range: 1.0e-7 to 200.0e-3
 *RST: 1.0e-6
 Default unit: Watts

TRIGger:LEVel:UNIT <unit>

Sets the unit of the trigger level if this value is entered without a unit.

See also [TRIGger:LEVel](#) on page 61.

Parameters:

<unit> DBM | W | DBUV
 *RST: W

TRIGger:MASTer:PORT <master_port>

Selects the port where the sensor outputs its own trigger event in case it is trigger master. See [TRIGger:MASTer:STATe](#) for more information.

If the sensor is the trigger master, it can output its trigger event either on the EXTernal<1> or EXTernal2.

If the sensor triggers itself, the trigger source of the sensor must be assigned to the other external port, as shown in the examples.

Parameters:

<master_port> EXT1 | EXTernal1 | EXT2 | EXTernal2
 *RST: EXT1

Example:

```
TRIG:MAST:PORT EXT1
TRIG:SOUR EXT2
TRIG:MAST:STAT ON
```

Example:

```
TRIG:MAST:PORT EXT2
TRIG:SOUR EXT1
TRIG:MAST:STAT ON
```

TRIGger:MASTer:STATe <state>

Enables or disables the trigger master mode of the sensor. In this state, the power sensor can output a digital trigger signal in sync with its own trigger event.

If enabled, select the output port for the trigger signal using [TRIGger:MASTer:PORT](#).

Typically, the trigger master uses its internal trigger source. But you can also trigger the trigger master externally, because the power sensor has got two external trigger connectors. If you trigger the trigger master externally, use EXTernal1 as external trigger input port (trigger source) and EXTernal2 as trigger master output port or vice versa.

Parameters:

<state> *RST: OFF

TRIGger:SLOPe <slope>

Effective only if **TRIGger:SOURce** is set to **INTernal** or **EXTernal**.

Determines which edge of the envelope power, with internal triggering, or increasing voltage, with external triggering, is used for triggering.

Parameters:

<slope> POSitive | NEGative
POSitive
Rising edge
NEGative
Falling edge
*RST: POSitive

TRIGger:SOURce <source>

Selects the source for the trigger event detector.

Parameters:

<source> HOLD | IMMEDIATE | INTernal | BUS | EXTernal | EXT1 |
EXTernal1 | EXT2 | EXTernal2
See [Chapter 8.5.2.3, "Trigger Sources"](#), on page 51.
*RST: IMMEDIATE

TRIGger:SYNC:PORT <sync_port>

Selects the external connection for the sync output of the sensor. For more information, see **TRIGger:SYNC:STATe**.

Parameters:

<sync_port> EXT1 | EXTernal1 | EXT2 | EXTernal2
*RST: EXT1

TRIGger:SYNC:STATe <state>

Usually used in combination with **TRIGger:MASTer:STATe** ON.

If enabled, blocks the external trigger bus as long as the sensor remains in the measurement state. Thus, ensures that a new measurement is only started after all sensors have completed their measurements.

Make sure that the number of repetitions is the same for all sensors involved in the measurement. Otherwise, the trigger bus is blocked by any sensor that has completed its measurements before the others and has returned to the idle state.

Parameters:

<state> *RST: OFF

8.6 Configuring and Retrieving Results

- [Setting the Power Unit](#)..... 64
- [Setting the Result Format](#)..... 64
- [Retrieving Results](#)..... 65

8.6.1 Setting the Power Unit

The `UNIT` subsystem contains commands for setting up the power unit.

UNIT:POWER <unit>

Sets the output unit for the measured power values.

Parameters:

<unit> DBM | W | DBUV
 *RST: W

Example: UNIT:POW DBM

8.6.2 Setting the Result Format

The `FORMat` subsystem sets the format of numeric data (measured values) that is exchanged between the remote control computer and the power sensors if high-level measurement commands are used.

Remote commands:

- [FORMat:BORDER](#)..... 64
- [FORMat:SREGister](#)..... 65
- [FORMat\[:DATA\]](#)..... 65

FORMat:BORDER <border>

Selects the order of bytes in 32-bit or 64-bit binary data.

Parameters:

<border> NORMal | SWAPped

NORMal

The 1st byte is the least significant byte, the 4th/8th byte the most significant byte.

Fulfills the Little Endian (little end comes first) convention, used by x86/x64 CPUs, for example.

SWAPped

The 1st byte is the most significant byte, the 4th/8th byte the least significant byte.

Fulfills the Big Endian (big end comes first) convention.

*RST: NORMal

Example: FORM:BORD NORM

FORMat:SREGister <register>

Specifies which format is used for the return value of *STB?.

Parameters:

<register> ASCII | HEXadecimal | OCTal | BINary
 *RST: ASCII

Example: FORM:SREG ASC

FORMat[:DATA] [<data,length>, <length>]

Specifies how the R&S NRP18S-xx sends the numeric data to the controlling host/ computer.

Parameters:

<data,length> <REAL,32 | 64>

REAL

Block of binary values, 32-bit or 64-bit each; so-called "SCPI definite length block"

32 | 64

32-bit or 64-bit

If you omit the length, the R&S NRP18S-xx sets the last used length.

Example for REAL, 32 format:

#6768000....<binary float values>....

Example for REAL, 64 format:

#71536000....<binary float values>....

<data[,length]> <ASCII[,0 to 12]>

ASCII

List of comma separated, readable values.

[,0 to 12]

Defines the number of decimal places.

The reset value 0 does not restrict the number of decimal places.

Example for ASCII, 4 format:

1.2938e-06, -4.7269e-11, ...

*RST: ASCII,0

8.6.3 Retrieving Results

After performing the measurement, you can query the measurement results with a command from the FETCh subsystem.

Remote commands:

[SENSe<Sensor>:]FUNCTION.....	66
FETCh<Sensor>:ARRay[:POWer][:AVG]?.....	67
FETCh<Sensor>[:SCALar][:POWer][:AVG]?.....	67

FETCh<Sensor>[:SCALar][:POWer]:BURSt?	67
FETCh<Sensor>[:SCALar][:POWer]:TSLot?	67
CALCulate:FEED	67
[SENSe<Sensor>:]AUXiliary	68

[SENSe<Sensor>:]FUNCTION <function>

Sets the measurement mode.

Parameters:

<function>

"POWer:AVG"

Continuous Average

After a trigger event, the power is integrated over a time interval (averaging) set with [SENSe<Sensor>:] [POWer:] [AVG:] [APERture].

"POWer:BURSt:AVG"

Burst

In remote control, this measurement mode is very similar to the ContAv mode. The integration time is, however, not predefined but determined by the sensor with the aid of a burst detector. The start of a burst is detected when the measurement signal rises above the set trigger level. The end is set when the signal drops below the trigger threshold.

[SENSe<Sensor>:] [POWer:] BURSt:DTOLerance defines the time interval during which the signal level must be below the trigger level so that the end of the burst can be detected. In the Burst mode, the set trigger source is ignored and TRIGger:SOURce on page 63INT is implicitly assumed.

"POWer:TSLot:AVG"

Timeslot

The power is simultaneously measured in up to 32 time windows.

The number of time windows is set with [SENSe<Sensor>:] [POWer:] TSLot[:AVG]:COUNT.

The length of a time window is determined via [SENSe<Sensor>:] [POWer:] TSLot[:AVG]:WIDTH. The measurement result is represented by a vector that can contain up to 32 indices and contains the power of a time window at each index.

"XTIME:POWer"

Trace

In this mode, power over time is measured. Therefore a number of measurement points are defined ([SENSe<Sensor>:] TRACe:POINts) where the length of an individual measurement is determined from the ratio of total time ([SENSe<Sensor>:] TRACe:TIME) and the defined number of measurement points.

*RST: "POWer:AVG"

FETCh<Sensor>:ARRay[:POWer][:AVG]?

Queries the last valid measurement result of a measurement with enabled data buffer mode.

Usage: Query only

FETCh<Sensor>[:SCALar][:POWer][:AVG]?

Queries the last valid measurement result.

Usage: Query only

FETCh<Sensor>[:SCALar][:POWer]:BURSt?

Queries the last valid measurement result for the burst average measurement mode.

Usage: Query only

FETCh<Sensor>[:SCALar][:POWer]:TSLot?

Queries the last valid measurement result of the timeslot average measurement mode.

Usage: Query only

CALCulate:FEED <mode>

If you query measurement data using **FETCh<Sensor>[:SCALar][:POWer][:AVG]?**, the power sensor returns data of the measurand that was configured before. Generally, this measurand is the average power. However, the power sensor can also output data of other measurands.

To configure which measurand the **FETCh<Sensor>[:SCALar][:POWer][:AVG]?** command sends, use the **CALCulate:FEED** command before the measurement is initiated. Depending on the measurement mode, the following settings are possible:

SENS:FUNC	Possible CALC:FEED	Meaning
"POWer:AVG"	"POWer:AVERAge"	Average value
	"POWer:PEAK"	Peak value
	"POWer:RANDom"	Randomly selected value from the measurement interval
"POWer:BURSt:AVG"	"POWer:AVERAge"	Average value
	"POWer:PEAK"	Peak value
	"POWer:RANDom"	Randomly selected value from the measurement interval
"POWer:TSLot:AVG"	"POWer:AVERAge"	Average value
	"POWer:PEAK"	Peak value

SENS:FUNC	Possible CALC:FEED	Meaning
	"POWer:RANDom"	Randomly selected value from the measurement interval
"XTIME:POWer"	"POWer:TRACe"	Measurement sequence
	"POWer:PEAK:TRACe"	Peak value of the samples per trace point
	"POWer:RANDom:TRACe"	Randomly selected value of the samples per trace point

Parameters:

<mode>

*RST: "POWer:AVERAge"

Example:

The following sequence of commands configures a peak trace measurement:

```
*RST
SENSe:FUNCTion "XTIME:POWer"
SENSe:FREQuency 1.0e9
SENSe:TRACe:POINts 500
SENS:TRAC:TIME 20e-3
TRIGGer:SOURce INTernal
TRIGGer:SLOPe POSitive
TRIGGer:DTIME 0.001
TRIGGer:HYSTeresis 0.1
TRIGGer:LEVel 30e-6
SENSe:TRACe:AVERAge:COUNt 8
SENSe:TRACe:AVERAge:STATe ON
CALCulate:FEED "POWer:PEAK:TRACe"
INITiate
FETCh?
```

[SENSe<Sensor>:]AUXiliary <mode>

Enables the measurement of additional measured values that are determined together with the main measured value.

Parameters:

<mode>

NONE | MINMax | RNDMax

NONE

No additional values are measured.

MINMax

Minima and maxima of the trace are transmitted together with the measured value.

Usually, extreme values are lost due to averaging the measured values.

RNDMax

Randomly selected samples are transmitted. All evaluations use these values instead of the average values.

*RST: NONE

8.7 Configuring the Measurement Modes

In the following chapter the settings needed for configuring a measurement mode are described.

Further information:

- [Chapter 8.8, "Configuring Basic Measurement Parameters"](#), on page 81
- [Chapter 8.5, "Controlling the Measurement"](#), on page 49

Contents:

• Continuous Average Measurement	69
• Burst Average Measurement	72
• Timeslot Average Measurement	74
• Trace Measurement	76

8.7.1 Continuous Average Measurement

The Continuous Average mode measures the signal average power asynchronously within definable time intervals (sampling windows). The aperture (width of the sampling windows) can be defined.

Reducing noise and zero offset

The smoothing filter can reduce result fluctuations caused by modulation. But activating it increases the inherent noise of the sensor by approx. 20 %, so do not activate it unless required.

Configuring continuous average measurements of modulated signals

When measuring modulated signals in continuous average mode, the measurement can show fluctuation due to the modulation. If that is the case, adapt the size of the sampling window exactly to the modulation period to get an optimally stable display. If the modulation period varies or is not precisely known, you can also activate the smoothing function.

With smoothing activated, the selected sampling window has to be 5 to 9 times larger than the modulation period so that the fluctuations caused by modulation are sufficiently reduced. The sampling values are subjected to weighting (raised-von-Hann window), which corresponds to video filtering.

If you deactivate the smoothing filter, 300 to 3000 periods are required to obtain the same effect. The sampling values are considered equivalent and are averaged in a sampling window, which yields an integrating behavior of the measuring instrument. To obtain optimum suppression of variations in the result, exactly adapt the modulation period to the size of the sampling window. Otherwise, the modulation can have a considerable influence, even if the sampling window is much larger than the modulation period.

Calculating the measurement time

Normally, the measurement time is calculated as follows:

$$MT = 2 * AC * APER + (2 * AC - 1) * 100 \mu s$$

with:

MT: overall measurement time

AC: average count

APER: aperture time

100 μs is the time for switching the chopper phase.

Using `[SENSe<Sensor>:] [POWer:] [AVG:] FAST ON`, you can accelerate the measurement as follows:

- Chopper is disabled.
- Average count is set to 1, no matter which average count you have set.

Thus, the overall measurement time is only defined by the aperture time, and the measurement time for a fast measurement is calculated as follows:

$$MT = APER$$

The fast measurement setting delivers up to 100 000 measurements per second without any blind time over randomly long time periods. Programming examples are given in [Chapter 9.2, "Performing the Fastest Measurement in Continuous Average Mode"](#), on page 120.

Remote commands:

<code>[SENSe<Sensor>:][POWer:][AVG:]APERture</code>	70
<code>[SENSe<Sensor>:][POWer:][AVG:]BUFFer:CLEar</code>	71
<code>[SENSe<Sensor>:][POWer:][AVG:]BUFFer:COUNT?</code>	71
<code>[SENSe<Sensor>:][POWer:][AVG:]BUFFer:DATA?</code>	71
<code>[SENSe<Sensor>:][POWer:][AVG:]BUFFer:SIZE</code>	71
<code>[SENSe<Sensor>:][POWer:][AVG:]BUFFer:STATe</code>	71
<code>[SENSe<Sensor>:][POWer:][AVG:]FAST</code>	72
<code>[SENSe<Sensor>:][POWer:][AVG:]SMOothing:STATe</code>	72

`[SENSe<Sensor>:][POWer:][AVG:]APERture` <integration_time>

Sets the duration of the sampling window in the continuous average mode. During this time interval, the average signal power is measured.

Parameters:

<integration_time> The minimum value is implemented for fast unchopped continuous average measurements. See also [Chapter 9.2.2, "Triggered Fast Unchopped Continuous Average Measurement"](#), on page 122.

Range: 8.0e-6 to 2.00

*RST: 0.02

Default unit: Seconds

Example: `APER 0.02`

[SENSe<Sensor>:][POWER:][AVG:]BUFFEr:CLEAr

Clears the contents of the result buffer in continuous average mode.

Example: `BUFF:CLE`

Usage: Event

[SENSe<Sensor>:][POWER:][AVG:]BUFFEr:COUNT?

Queries the number of results that are currently stored in the result buffer. Available in continuous average mode.

Example: `BUFF:COUN?`

Usage: Query only

[SENSe<Sensor>:][POWER:][AVG:]BUFFEr:DATA?

Queries the results of the continuous average result buffer and returns them even if the buffer is not full.

In contrast, `FETCH<Sensor>[:SCALar][:POWER][:AVG]?` returns a result only if the buffer is full.

Usage: Query only

[SENSe<Sensor>:][POWER:][AVG:]BUFFEr:SIZE <count>

Sets the size of the result buffer in continuous average mode.

You can enable the buffer using `[SENSe<Sensor>:][POWER:][AVG:]BUFFEr:STATe`.

Parameters:

<count> Range: 1 to 8192
 *RST: 1

Example: `BUFF:SIZE 1`
 See [Chapter 9.3, "Performing a Buffered Continuous Average Measurement"](#), on page 123.

[SENSe<Sensor>:][POWER:][AVG:]BUFFEr:STATe <state>

Enables or disables the buffered continuous average mode. If the buffer mode is enabled, all results generated by trigger events are collected in the sensor until the buffer is filled.

You can set the size of the buffer using `[SENSe<Sensor>:][POWER:][AVG:]BUFFEr:SIZE`.

Parameters:

<state> ON | OFF
 *RST: OFF

Example: BUFF:STAT OFF

[SENSe<Sensor>:][POWer:][AVG:]FAST <state>

Enables or disables a fast unchopped continuous average measurement. If enabled, the average count is enforced to 1, and any setting for average count is silently ignored.

You can increase the measurement accuracy by increasing the duration of the sampling window using [SENSe<Sensor>:] [POWer:] [AVG:] APERTure.

The fast measurement setting delivers up to 100 000 measurements per second without any blind time over randomly long time periods.

See also "Calculating the measurement time" on page 69.

Parameters:

<state> *RST: OFF

Example: FAST ON
 See [Chapter 9.2, "Performing the Fastest Measurement in Continuous Average Mode"](#), on page 120.

[SENSe<Sensor>:][POWer:][AVG:]SMOothing:STATe <state>

Enables or disables the smoothing filter, a steep-edge digital lowpass filter. If you cannot adjust the aperture time exactly to the modulation period, the filter reduces result fluctuations caused by modulation.

Parameters:

<state> ON | OFF
 *RST: OFF

Example: SMO:STAT OFF

8.7.2 Burst Average Measurement

The Burst Average mode is used to measure the average power of bursts. The time interval in which the average power is measured starts when the power exceeds the trigger level and stops when the trigger logic detects the end of the pulse.

[Fig. 8-2](#) shows a graphical representation of a burst average measurement and the meaning of the parameters that can be configured.

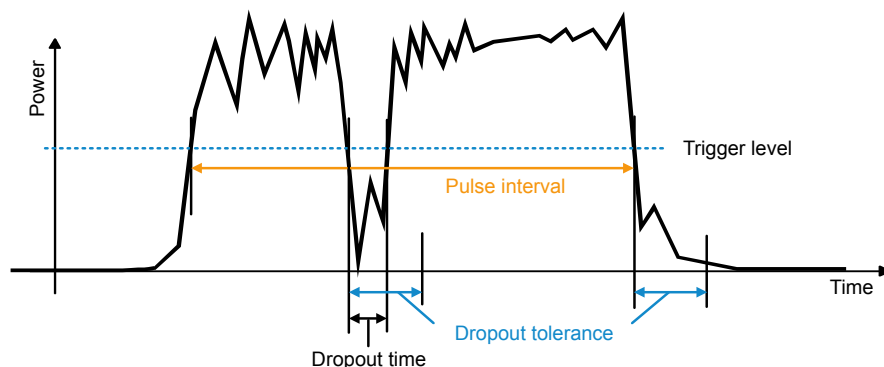


Figure 8-2: Burst average measurement parameters

To prevent power drops due to modulation from being erroneously interpreted as the end of a pulse, you must define the dropout tolerance. This is a time interval in which the pulse end is only recognized if the signal level no longer exceeds the trigger level.

Triggering a burst average measurement

In the burst average mode, only internal trigger events from the signal are evaluated, irrespective of the setting of the `TRIGger:SOURce` parameter. The `TRIGger:DElay` parameter is also ignored, so that the measurement interval begins exactly when the signal exceeds the trigger level.

Defining a time interval for the measurement

Time intervals that are to be excluded from the measurement can be set at the beginning and at the end of the measurement interval with the commands `[SENSe<Sensor>:]TIMing:EXCLude:START` and `[SENSe<Sensor>:]TIMing:EXCLude:STOP` on page 86, see [Chapter 8.8.3, "Setting Exclusion Time"](#), on page 85.

Remote commands:

<code>[SENSe<Sensor>:][POWER:]BURSt:DTOLerance</code>	73
<code>[SENSe<Sensor>:][POWER:]BURSt:LENGth?</code>	74

`[SENSe<Sensor>:][POWER:]BURSt:DTOLerance <tolerance>`

Sets the drop-out tolerance, a time interval in which the pulse end is only recognized if the signal level no longer exceeds the trigger level (see [Figure 8-2](#)).

Parameters:

<tolerance>	Range:	0.00 to 0.30
	*RST:	1.000e-6
	Default unit:	Seconds

[SENSe<Sensor>:][POWER:]BURSt:LENGth?

Queries the length of a burst (pulse interval), the time between the trigger point of the measurement and the time the trigger logic detects the end of the pulse (see [Figure 8-2](#)).

Usage: Query only

8.7.3 Timeslot Average Measurement

The Timeslot Average mode is used to measure the average power of a definable number of successive timeslots within a frame structure with equal spacing. The measurement result is an array with the same number of elements as timeslots. Each element represents the average power in a particular timeslot.

[Fig.8-3](#) shows a graphical representation of a timeslot average measurement and the meaning of the parameters that can be configured.

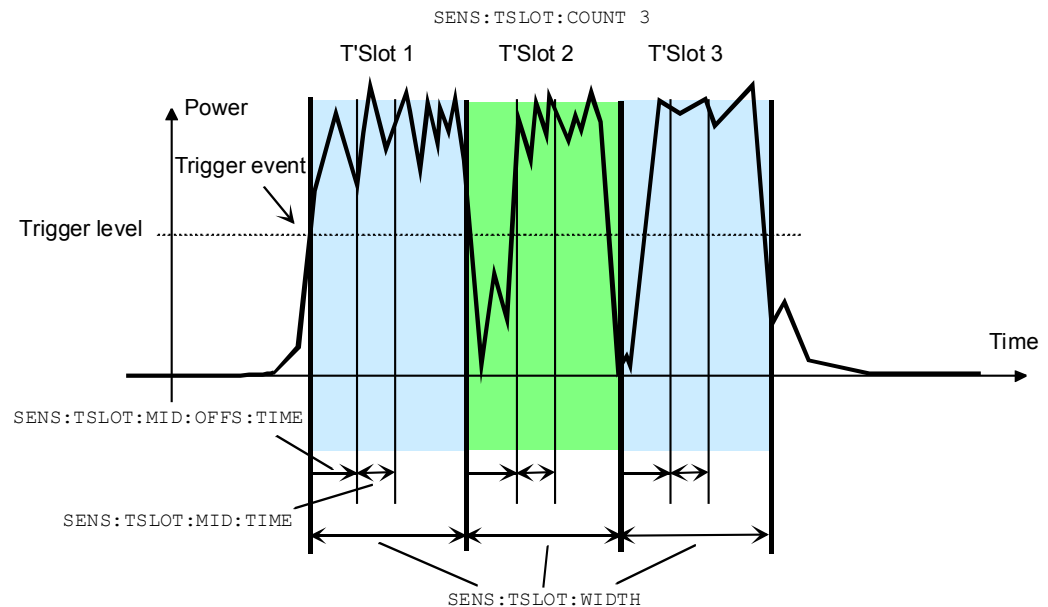


Figure 8-3: Timeslot parameters

Triggering a timeslot average measurement

In the timeslot mode, internal and external trigger events from the signal are evaluated depending on the settings of the **TRIGger:SOURce** parameter. It is essential to define the **TRIGger:DElay** parameter to ensure that the beginning of the first slot to be measured coincides with the delayed trigger point.

Defining a time interval for the measurement

Additionally, time intervals that are to be excluded from the measurement can be set at the beginning and at the end of the measurement interval, with the commands

[SENSe<Sensor>:]TIMing:EXCLude:START and [SENSe<Sensor>:]TIMing:EXCLude:STOP, see [Chapter 8.8.3, "Setting Exclusion Time"](#), on page 85.

Remote commands:

[SENSe<Sensor>:][POWER:]TSLot[:AVG]:COUNT.....	75
[SENSe<Sensor>:][POWER:]TSLot[:AVG]:WIDTH.....	75
[SENSe<Sensor>:][POWER:]TSLot[:AVG][:EXCLude]:MID:OFFSet[:TIME].....	75
[SENSe<Sensor>:][POWER:]TSLot[:AVG][:EXCLude]:MID:TIME.....	75
[SENSe<Sensor>:][POWER:]TSLot[:AVG][:EXCLude]:MID[:STATe].....	76

[SENSe<Sensor>:][POWER:]TSLot[:AVG]:COUNT <count>

Sets the number of simultaneously measured timeslots in the timeslot mode (see [Figure 8-3](#)).

Parameters:

<count>	Range:	1 to 128
	*RST:	8

[SENSe<Sensor>:][POWER:]TSLot[:AVG]:WIDTH <width>

Sets the length of the timeslot (see [Figure 8-3](#)).

Parameters:

<width>	Range:	10.0e-6 to 0.10
	*RST:	1.000e-3
	Default unit:	Seconds

[SENSe<Sensor>:][POWER:]TSLot[:AVG][:EXCLude]:MID:OFFSet[:TIME] <time>

Determines the distance from the start of the timeslots to the start of the interval to be blanked out (see [Figure 8-3](#)).

Parameters:

<time>	Range:	0.00 to 0.10
	*RST:	0.00
	Default unit:	Seconds

[SENSe<Sensor>:][POWER:]TSLot[:AVG][:EXCLude]:MID:TIME <time>

Sets the length of the time interval in the timeslots to be excluded from the measurement (see [Figure 8-3](#)). The parameter applies to each individual timeslot.

Note: Even if the exclusion interval exceeds the timeslot because, for example, its right limit is outside the timeslot, correct results are obtained. In the extreme case, where the interval length has been set to a value greater than the timeslot length, 0 W is output as the measured power. No error message is output.

Parameters:

<time> Range: 0.00 to 0.10
 *RST: 0.00
 Default unit: Seconds

[SENSe<Sensor>:] [POWER:] TSLot[:AVG][:EXCLude]:MID[:STATe] <state>

Enables or disables the blanking out of time intervals in the timeslots.

Parameters:

<state> *RST: OFF

8.7.4 Trace Measurement

The trace measurement is used to acquire the course of power over a certain time. During the measurement time ([SENSe<Sensor>:] TRACe:TIME) a large number of measurements are made and the result is returned to the user as an array of values with a predefined size [SENSe<Sensor>:] TRACe:POINts. The length of an individual measurement(-point) is determined from the ratio of measurement time and measurement points. The entire result is called a "trace". Each trace must be triggered separately.

Remote commands:

[SENSe<Sensor>:]TRACe:AVERage:COUNT.....	76
[SENSe<Sensor>:]TRACe:AVERage:TCONtrol.....	76
[SENSe<Sensor>:]TRACe:AVERage[:STATe].....	77
[SENSe<Sensor>:]TRACe:DATA?.....	77
[SENSe<Sensor>:]TRACe:MPWidth?.....	80
[SENSe<Sensor>:]TRACe:OFFSet:TIME.....	80
[SENSe<Sensor>:]TRACe:POINts.....	80
[SENSe<Sensor>:]TRACe:REALtime.....	80
[SENSe<Sensor>:]TRACe:TIME.....	81

[SENSe<Sensor>:]TRACe:AVERage:COUNT <count>

Sets the number of readings that are averaged for one measured value. The higher the count, the lower the noise, and the longer it takes to obtain a measured value.

Averaging is only effective, if [SENSe<Sensor>:] TRACe:AVERage [:STATe] ON is set.

Parameters:

<count> Range: 1 to 65536
 *RST: 4

[SENSe<Sensor>:]TRACe:AVERage:TCONtrol <mode>

Available in trace mode.

Defines how the measurement results are output. This is called termination control.

See also [Chapter 8.5, "Controlling the Measurement"](#), on page 49.

Parameters:

<mode> MOVing | REPeat

MOVing

Outputs intermediate values to facilitate early detection of changes in the measured quantity. In the settled state, that means when the number of measurements specified by the average count has been performed, a moving average is output.

REPeat

Specifies that a measurement result is not output until the entire measurement has been completed. This means that the number of measurement cycle repetitions is equal to the set average count. If the average count is large, the measurement time can be very long.

The average count is set using [\[SENSe<Sensor>:\]TRACe:AVERage:COUNT](#).

*RST: REPeat

Example:

TRAC:AVER:TCON REP

[SENSe<Sensor>:]TRACe:AVERage[:STATe] <state>

Switches the averaging filter on and off for the Trace mode.

Parameters:

<state> *RST: ON

[SENSe<Sensor>:]TRACe:DATA?

Returns the measured trace data in a well-defined format.

Unlike the `FETCh?` command, [\[SENSe<Sensor>:\]TRACe:DATA](#) takes the settings of [\[SENSe<Sensor>:\]AUXiliary](#) into account, as explained below.

Command response

To describe the format of the command response, it is important to know some additional information.

Besides the average power, the sensor can measure additional measurands like minimum, maximum or random. These additional measurands are denoted as auxiliary measurands and can be selected through the [\[SENSe<Sensor>:\]AUXiliary](#) command.

A trace measurement can be configured to deliver up to three measurands. Therefore, the resulting block of data which is provided by a [\[SENSe<Sensor>:\]TRACe:DATA](#) query can contain 1 to 3 blocks of user data.

Basically, the [\[SENSe<Sensor>:\]AUXiliary](#) response represents a Definite Length Arbitrary Block Response Data as defined in IEEE488.2. This object consists of a header and some content.

In principle, the response has the format as shown in [Figure 8-4](#):

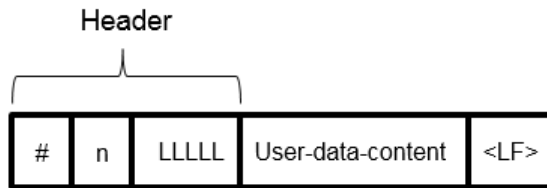


Figure 8-4: Response format

- Header consisting of:
 - Character #
 - Single digit ('n') which tells the number of following digits that is taken as the size of the content.
 - Number consisting of as many digits as 'n' specified ('LLLLL'). This number gives the size of the content
- Content ('user data content'), see also [Figure 8-5](#). As many bytes as 'LLLLL' specified.
- Single linefeed character (symbolically shown as <LF>, [Response format](#))

Example

The arbitrary block response data for a user data that contains 45182 bytes is:

```
#545182xxxxxx.....xxxxxx <LF>
```

The arbitrary block response data for a user data content 'THIS IS A TEST' is:

```
#214THIS IS A TEST<LF>
```

Explanation: 'THIS IS A TEST' has 14 bytes, and '14' has 2 digits, hence the #214

User data content

The previous paragraphs described how to separate the "user data content" from the header. We keep the designator "user data content" in the further description for denoting the totality of the contained measurement results.

The further description deals with the user data content and shows what is embedded in it. There are similar mechanisms as with arbitrary block response data in the user data content. As indicated above, the user data content can have 1 or more sections with trace measurement results, depending on the selection of auxiliary measurands. Each section is composed of:

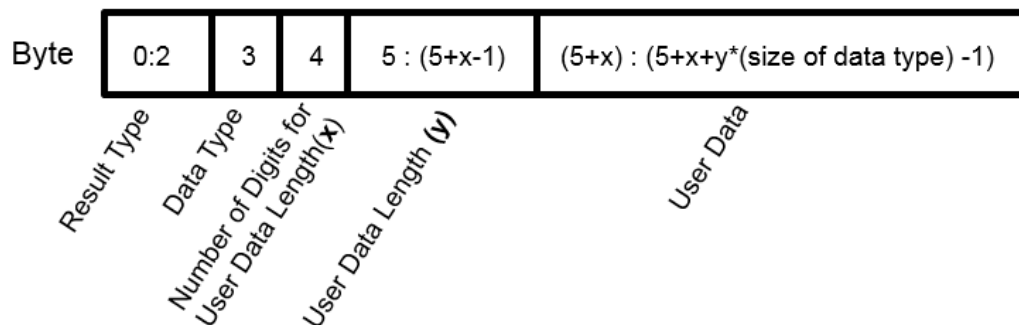


Figure 8-5: User data content format

y = Number of values which follow the header

x = Number of digits of y

- Result type (always 3 bytes, one of 'AVG', 'MIN', 'MAX' or 'RND')
- Designator for the contained data type with the size of 1 byte
Currently, the only possible designator is 'f' for 4-byte IEEE754 float data type, little endian
- Single digit which tells the number of following digits that are taken as the number of contained float values
- As many digits as the number of digits for user data specified ('x'). These digits are interpreted as the number of values ('y') (not number of bytes) which follow the header
- Measurement result values in the format that is described by the data type (currently IEEE754 float only)

If no `[SENSe<Sensor>:]AUXiliary` measurands have been activated before executing the measurement, the user data content is finished here. In case that auxiliary measurands have been selected, the above section is repeated for every auxiliary measurand. The user data content looks like:

```
AVGf3100...(400byte AVG values)...MINf3100...(400byte min. values)...MAXf3100...(400byte max. values)...
```

Where each of

```
...(400byte AVG values)...
...(400byte min. values)...
...(400byte max. values)...
```

Stands for 400 bytes as the equivalent of 100 float values.

Keep in mind that this is only the user data content. It is embedded in the arbitrary block response data response of the `[SENSe<Sensor>:]TRACe:DATA` query.

Example: `TRAC:DATA?`

Usage: Query only

[SENSe<Sensor>:]TRACe:MPWidth?

Queries the attainable time resolution for the Trace mode. The result is the smallest possible distance between two pixels, i.e. it is the smallest time interval that can be assigned to a pixel.

Usage: Query only

[SENSe<Sensor>:]TRACe:OFFSet:TIME <time>

Adds an offset to the beginning of the trace sequence. Thus, the trace in the result display is moved in positive or negative x-direction. If you measure with more than one sensor, you can use this offset to arrange the traces to each other. The start of recording relative to the trigger event is set using [TRIGger:DELaY](#).

Parameters:

<time> Range: - (500 ms + TRGger:DELaY) to 100 s
 *RST: 0.0
 Default unit: Seconds

Example: TRAC:OFFS:TIME 1.0

[SENSe<Sensor>:]TRACe:POINts <points>

Sets the number of required values per trace sequence.

Parameters:

<points> Range: 1 to 100000
 *RST: 260

[SENSe<Sensor>:]TRACe:REALtime <state>

Available in trace mode.

If disabled, each measurement from the power sensor is averaged. If enabled, only one sampling sequence per measurement is recorded, thus increasing the measurement speed. With a higher measurement speed, the measured values of an individual measurement are immediately delivered.

The averaging filter is not used, so the following settings are ignored:

- [\[SENSe<Sensor>:\] TRACe:AVErAge \[:STATe\]](#)
- [\[SENSe<Sensor>:\] TRACe:AVErAge:COUNT](#)

Parameters:

<state> *RST: OFF

[SENSe<Sensor>:]TRACe:TIME <time>

Sets the trace length, time to be covered by the trace sequence. This time period is divided into a number of equal intervals, in which the average power is determined. The number of intervals equals the number of trace points, which is set with the command `[SENSe<Sensor>:]TRACe:POINts`.

Parameters:

<time> Range: 10.0e-6 to 3.0
 *RST: 0.01
 Default unit: Seconds

8.8 Configuring Basic Measurement Parameters

The following section describes the settings common for several measurement modes.

8.8.1 Configuring Auto Averaging

Describes the commands for automatic averaging in the continuous average and burst measurements.

Remote commands:

<code>[SENSe<Sensor>:]AVERage:COUNT</code>	81
<code>[SENSe<Sensor>:]AVERage:COUNT:AUTO</code>	82
<code>[SENSe<Sensor>:]AVERage:COUNT:AUTO:MTIME</code>	82
<code>[SENSe<Sensor>:]AVERage:COUNT:AUTO:NSRatio</code>	82
<code>[SENSe<Sensor>:]AVERage:COUNT:AUTO:RESolution</code>	83
<code>[SENSe<Sensor>:]AVERage:COUNT:AUTO:SLOT</code>	83
<code>[SENSe<Sensor>:]AVERage:RESet</code>	83
<code>[SENSe<Sensor>:]AVERage:COUNT:AUTO:TYPE</code>	84
<code>[SENSe<Sensor>:]AVERage:TCONtrol</code>	84
<code>[SENSe<Sensor>:]AVERage[:STATe]</code>	84

[SENSe<Sensor>:]AVERage:COUNT <count>

Sets the number of readings that are averaged for one measured value. The higher the count, the lower the noise, and the longer it takes to obtain a measured value.

Average count is often also called averaging factor, but it designates the same thing, i.e. the number of measured values that have to be averaged for forming the measurement result.

Averaging is only effective, if `[SENSe<Sensor>:]AVERage[:STATe] ON` is set.

Parameters:

<count> Range: 1 to 65536
 *RST: 4

Example: AVER:COUNT 1

[SENSe<Sensor>:]AVERage:COUNT:AUTO <state>

Sets the mode for determining the average count.

Parameters:

<state>

ON

Auto averaging: the averaging factor is continuously determined and set depending on the level of power and other parameters.

OFF

Fixed filter: the previous, automatically determined averaging factor is used.

ONCE

An averaging factor is determined by the filter automatic function under the current measurement conditions and is then used in the fixed filter mode.

*RST: ON

[SENSe<Sensor>:]AVERage:COUNT:AUTO:MTIME <maximum_time>

Sets an upper limit for the settling time of the auto-averaging filter if `[SENSe<Sensor>:]AVERage:COUNT:AUTO:TYPE` is set to `NSRatio`. Thus it limits the length of the filter.

Parameters:

<maximum_time>

Range: 0.01 to 999.99

*RST: 4.00

Default unit: Seconds

[SENSe<Sensor>:]AVERage:COUNT:AUTO:NSRatio <nsr>

Determines the relative noise component in the measurement result for the measurement modes with scalar results. These measurement modes are Continuous Average, Burst Average and Timeslot Average, provided the particular sensor supports them.

This command is only effective if the auto average calculation is enabled:

- `[SENSe<Sensor>:]AVERage:COUNT:AUTO` ON
- `[SENSe<Sensor>:]AVERage:COUNT:AUTO:TYPE` NSR

The noise component is defined as the magnitude of the level variation in dB caused by the inherent noise of the sensor (two standard deviations).

The query returns the relative noise component in the measured value.

Parameters:

<nsr>

Range: 100.000e-6 to 1.00

*RST: 0.01

Default unit: dB

[SENSe<Sensor>:]AVERage:COUNT:AUTO:RESolution <resolution>

Defines the number of significant places for linear units and the number of decimal places for logarithmic units which should be free of noise in the measurement result.

The setting is only taken into account, if [SENSe<Sensor>:]AVERage:COUNT:AUTO ON and [SENSe<Sensor>:]AVERage:COUNT:AUTO:TYPE RES are set.

Parameters:

<resolution>	Range:	1 to 4
	*RST:	3

[SENSe<Sensor>:]AVERage:COUNT:AUTO:SLOT <slot>

Available only if [SENSe<Sensor>:]FUNCTION is set to POWER:TSLOT:AVG.

Sets a timeslot from which the measured value is used to determine the filter length automatically. The timeslot number must not exceed the number of the currently set timeslots.

Parameters:

<slot>	Range:	1 to 128
	*RST:	1

[SENSe<Sensor>:]AVERage:RESet

Deletes all previous measurement results that the averaging filter contains and initializes the averaging filter. The filter length gradually increases from 1 to the set averaging factor. Thus, trends in the measurement result become quickly apparent. Note that the measurement time required for the averaging filter to settle completely remains unchanged.

Use this command if:

- High averaging factor is set.
[SENSe<Sensor>:]AVERage:COUNT
- Intermediate values are output as measurement results.
[SENSe<Sensor>:]AVERage:TCONTROL MOVing
- Power has significantly decreased since the previous measurement, for example by several powers of 10.

In this situation, previous measurement results, which are still contained in the averaging filter, strongly affect the settling of the display. As a result, the advantage of detecting trends in the measurement result while the measurement is still in progress is lost.

Example: AVER:RES

Usage: Event

[SENSe<Sensor>:]AVERage:COUNT:AUTO:TYPE <type>

Sets the automatic averaging filter mode.

Parameters:

<type> RESolution | NSRatio

RESolution

The usual mode for the power sensors.

NSRatio

Predefines the compliance to an exactly defined noise component.

*RST: RESolution

[SENSe<Sensor>:]AVERage:TCONtrol <mode>

Defines how the measurement results are output. This is called termination control.

See also [Chapter 8.5, "Controlling the Measurement"](#), on page 49.

Parameters:

<mode> MOVing | REPeat

MOVing

Outputs intermediate values to facilitate early detection of changes in the measured quantity. In the settled state, that means when the number of measurements specified by the average count has been performed, a moving average is output.

REPeat

Specifies that a measurement result is not output until the entire measurement has been completed. This means that the number of measurement cycle repetitions is equal to the set average count. If the average count is large, the measurement time can be very long.

The average count is set using [\[SENSe<Sensor>:\]AVERage:COUNT](#) on page 81.

*RST: REPeat

Example: AVER:TCON REP

[SENSe<Sensor>:]AVERage[:STATe] <state>

Enables or disables the averaging filter.

Parameters:

<state> *RST: ON

8.8.2 Setting the Frequency

The frequency of the signal to be measured is not automatically determined. For achieving better accuracy, the carrier frequency of the applied signal must be set.

[SENSe<Sensor>:]FREQuency <frequency>

Transfers the carrier frequency of the RF signal to be measured. This frequency is used for the frequency-response correction of the measurement result.

The center frequency is set for broadband signals, e.g. spread-spectrum signals, multi-carrier signals.

Parameters:

<frequency> Range: 0.0 to 110.0e9
 *RST: 50.0e6
 Default unit: Frequency

Example: `FREQ 10000`

8.8.3 Setting Exclusion Time

In the burst average and the timeslot average modes, you can define a time period at the beginning or at the end of an integration period, which is excluded from the measurement result, as shown in [Figure 8-6](#).

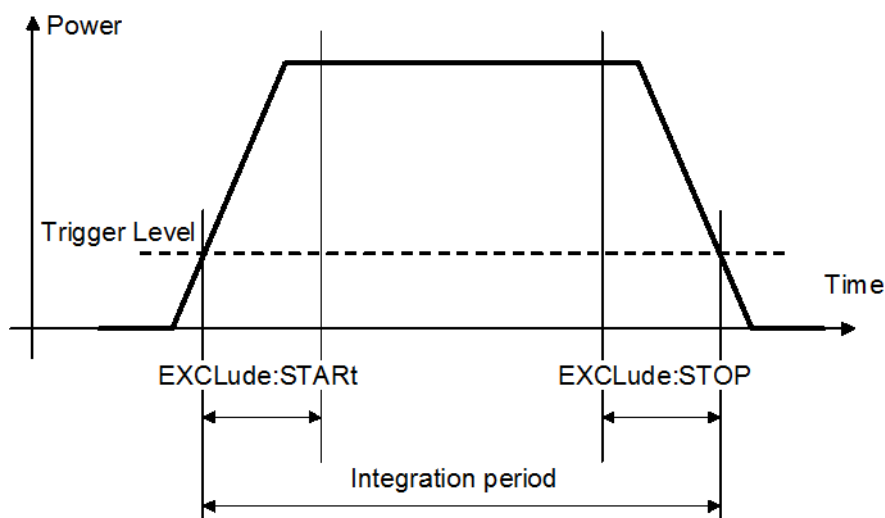


Figure 8-6: Effect of commands `SENS:TIM:EXCL:STAR` and `SENS:TIM:EXCL:STOP`

Remote commands:

`[SENSe<Sensor>:]TIMing:EXCLude:STARt`.....86
`[SENSe<Sensor>:]TIMing:EXCLude:STOP`.....86

[SENSe<Sensor>:]TIMing:EXCLude:STARt <exclude_start>

Sets a time that is to be excluded at the beginning of the integration period, see [Figure 8-6](#).

Parameters:

<exclude_start> Range: 0.0 to 1.0
 *RST: 0.0
 Default unit: Seconds

[SENSe<Sensor>:]TIMing:EXCLude:STOP <exclude_stop>

Sets a time that is to be excluded at the end of the integration period, see [Figure 8-6](#).

Parameters:

<exclude_stop> Range: 0.0 to 1.0
 *RST: 0.0
 Default unit: Seconds

8.8.4 Configuring Corrections

It is possible to set some parameters that compensate for a change of the measured signal due to fixed external influences.

- [Duty Cycle Corrections](#)..... 86
- [Offset Corrections](#)..... 87
- [S-Parameter Correction](#)..... 87
- [S-Gamma Corrections](#)..... 90
- [Using the S-Parameters Tool](#)..... 91

8.8.4.1 Duty Cycle Corrections

The duty cycle is the percentage of one period during which the signal is active, when pulse-modulated signals are corrected. The duty cycle is only evaluated in the Continuous Average mode.

Remote commands:

[\[SENSe<Sensor>:\]CORRection:DCYCLE](#)..... 86
[\[SENSe<Sensor>:\]CORRection:DCYCLE:STATe](#)..... 87

[SENSe<Sensor>:]CORRection:DCYCLE <duty_cycle>

Available in continuous average mode.

Sets the duty cycle for measuring pulse-modulated signals. The duty cycle defines the percentage of one period during which the signal is active. If the duty cycle is enabled, the R&S NRP18S-xx takes this percentage into account when calculating the signal pulse power from the average power.

Parameters:

<duty_cycle> Range: 0.001 to 100.00
 *RST: 1.00
 Default unit: Percent

[SENSe<Sensor>:]CORRection:DCYClE:STATe <state>

Enables or disables the duty cycle correction for the measured value.

Parameters:

<state> *RST: OFF

8.8.4.2 Offset Corrections

The offset accounts for external losses by adding a fixed level offset in dB.

The attenuation of an attenuator located ahead of the sensor (or the coupling attenuation of a directional coupler) is taken into account with a positive offset, i.e. the sensor calculates the power at the input of the attenuator or the directional coupler. A negative offset can be used to correct the influence of an amplifier connected ahead.

Using S-parameters instead of a fixed offset allows more precise measurements, because the interaction between the sensor and the component can be taken into account. See also [Chapter 8.8.4.3, "S-Parameter Correction"](#), on page 87.

Remote commands:

[SENSe<Sensor>:]CORRection:OFFSet..... 87
 [SENSe<Sensor>:]CORRection:OFFSet:STATe..... 87

[SENSe<Sensor>:]CORRection:OFFSet <offset>

Sets a fixed offset that is added to correct the measured value.

Parameters:

<offset> Range: -200.00 to 200.00
 *RST: 0
 Default unit: dB

[SENSe<Sensor>:]CORRection:OFFSet:STATe <state>

Enables or disables the offset correction.

Parameters:

<state> *RST: OFF

Example: CORR:OFFS:STAT ON

8.8.4.3 S-Parameter Correction

S-parameter correction compensates for the losses and reflections introduced by a component – such as an attenuator, directional coupler, or matching pad – that is

attached to a power sensor. Using S-parameters instead of a fixed offset increases measurement accuracy by accounting for the interaction between the sensor and the component. It shifts the reference plane of the sensor from its RF connector to the input of the device that is being applied externally. For more information on the fundamentals of the S-parameters and for application example, see also [1GP70: Using S-Parameters with R&S®NRP-Z Power Sensors](#).

All R&S NRP18S-xx power sensors allow compensating for the influence of any 2-port device between the signal source and the sensor input. As a result, the firmware can calculate the power that the signal source actually delivers. Examples of such 2-port devices include attenuators, matching pads, minimum-loss pads and waveguide adapters. One precondition for such compensation is that you provide a complete set of S-parameter data for the 2-port device in the frequency range required by the application.

The S-parameters of the attenuator delivered with the R&S NRP18S-xx have been measured by Rohde & Schwarz. The results of the factory calibration, including an S-parameter table that matches the delivered attenuator, are stored in the factory calibration data set of the sensor. If you use this attenuator, its effect on the measurement is compensated arithmetically.

Achieving Maximum Measurement Sensitivity

For maximum measurement sensitivity, you can choose from the following methods.

To operate the R&S NRP18S-xx without an attenuator

- ▶ Disable the S-parameter correction:
 - Each time after you have put the power sensor into operation.
`[SENSe<Sensor>:]CORRection:SPDevice:STATE.`
 - Permanently, using the S-Parameters tool.
See [Chapter 8.8.4.5, "Using the S-Parameters Tool"](#), on page 91.

To replace the delivered attenuator with any other 2-port device

1. Measure the S-parameters of the 2-port device.
2. Load the S-parameters into the power sensor.
In the user calibration data set of the power sensor, you can manage the S-parameters of several 2-port devices beside the S-parameters of the attenuator delivered with the power sensor. The sensor can apply the sets of S-parameters individually, depending on which S-parameter device you select as the active device.
3. Make sure that the S-parameter settings - selected S-parameter device, S-parameter correction state - always match the used hardware configuration.

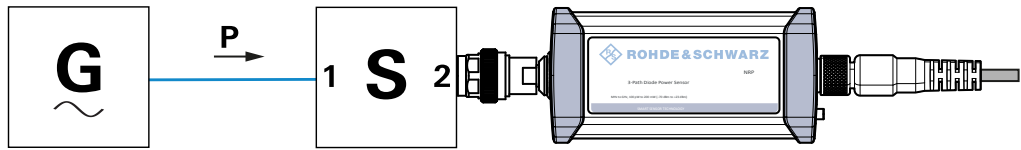


Figure 8-7: Operation with 2-port device between signal source and sensor input

Configuring the S-Parameter Correction

[SENSe<Sensor>:]CORRection:SPDevice:LIST?	89
[SENSe<Sensor>:]CORRection:SPDevice:SELection	89
[SENSe<Sensor>:]CORRection:SPDevice:STATe	89

[SENSe<Sensor>:]CORRection:SPDevice:LIST?

Queries the list of the S-parameter data sets that have been downloaded to the power sensor. The result of the query indicates the consecutive number and mnemonic of each data set.

Usage: Query only

[SENSe<Sensor>:]CORRection:SPDevice:SELection <num>

Selects a downloaded data set for S-parameter correction.

See also [Chapter 8.8.4.3, "S-Parameter Correction"](#), on page 87.

Parameters:

<num>	Range:	1 to 1999
	*RST:	1; can differ if a calibration set defines another value.

Manual operation: See ["S-Parameter Device Locked"](#) on page 94

[SENSe<Sensor>:]CORRection:SPDevice:STATe <state>

Activates or deactivates the S-parameter correction. If activated, uses the S-parameter data set selected by [SENSe<Sensor>:]CORRection:SPDevice:SELection.

See also [Chapter 8.8.4.3, "S-Parameter Correction"](#), on page 87.

Parameters:

<state>	*RST:	OFF; can differ if a calibration set defines another value.
---------	-------	---

Example:

```
CORR:SPD:SEL 1
Selects an S-parameter correction data set.
CORR:SPD:STAT ON
Activates the S-parameter correction.
```

Manual operation: See ["S-Parameter Correction State Locked"](#) on page 94

8.8.4.4 S-Gamma Corrections

Using the complex reflection coefficient, you can determine the power P delivered by the signal source with considerably greater accuracy.

The coefficient of the signal source Γ_{source} is defined by its magnitude and phase:

- `[SENSe<Sensor>:]SGAMma:MAGNitude`
- `[SENSe<Sensor>:]SGAMma:PHASe`

The complex reflection coefficient Γ_{sensor} of the sensor, which is also required for the correction, is prestored in the calibration data memory for many frequencies.

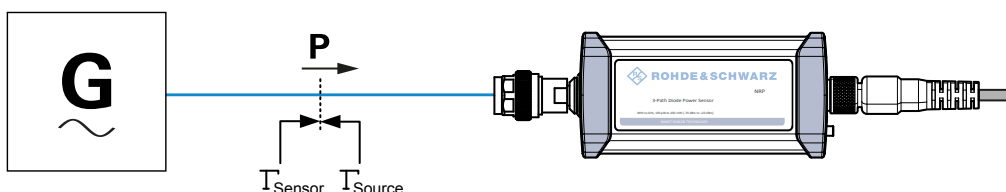


Figure 8-8: Correction of interactions between the power sensor and the signal source

If the gamma correction is performed in combination with an S-parameter correction (`[SENSe<Sensor>:]CORRection:SPDevice:STATe ON`), the following is taken into account:

- Interaction of the signal source with the S-parameter device
- Input of the power sensor, depending on the transmission, expressed by the term $S_{12}S_{21}$

If the S-parameter correction is enabled, the interaction between the complex reflection coefficient Γ_{sensor} of the power sensor and the reflection of port 2, expressed by s_{22} , is always taken into account, regardless whether gamma correction is performed or not.

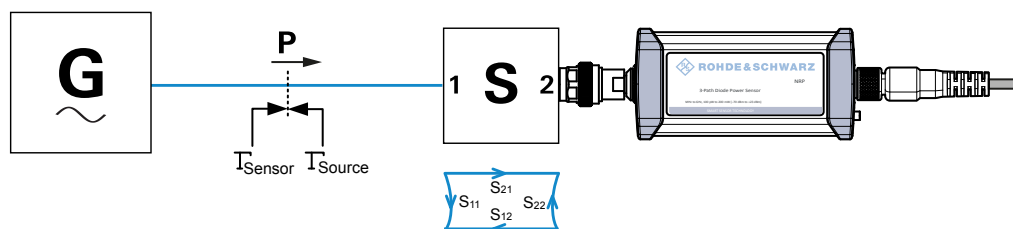


Figure 8-9: Correction of interactions between the power sensor, the signal source, and the S-parameter device

Remote commands:

<code>[SENSe<Sensor>:]SGAMma:CORRection:STATe</code>	91
<code>[SENSe<Sensor>:]SGAMma:MAGNitude</code>	91
<code>[SENSe<Sensor>:]SGAMma:PHASe</code>	91

[SENSe<Sensor>:]SGAMma:CORRection:STATe <state>

Enables or disables the use of the complex reflection coefficient to correct the interactions of power sensor and signal source.

Parameters:

<state> *RST: OFF

[SENSe<Sensor>:]SGAMma:MAGNitude <magnitude>

Sets the magnitude of the complex reflection coefficient of the source, Γ_{source} .

Parameters:

<magnitude> 0.0 \triangleq ideal matched source
 1.0 \triangleq total reflection
 Range: 0.0 to 1.0
 *RST: 0.0

[SENSe<Sensor>:]SGAMma:PHASe <phase>

Sets the phase angle of the complex reflection coefficient of the source, Γ_{source} .

Parameters:

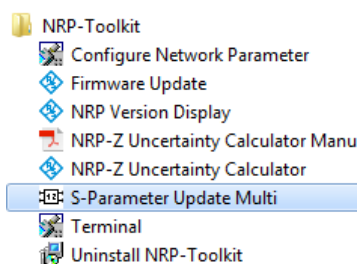
<phase> Range: -360.0 to 360.0
 *RST: 0.0

8.8.4.5 Using the S-Parameters Tool

The S-Parameters tool helps loading an S-parameter table into the power sensor. The S-Parameters tool is part of the R&S NRP Toolkit, see [Chapter 5.1, "R&S NRP Toolkit"](#), on page 18.

To start the S-Parameters tool

- In the Windows start menu, select "NRP Toolkit" > "S-Parameter Update Multi".



User Interface of the S-Parameters Tool

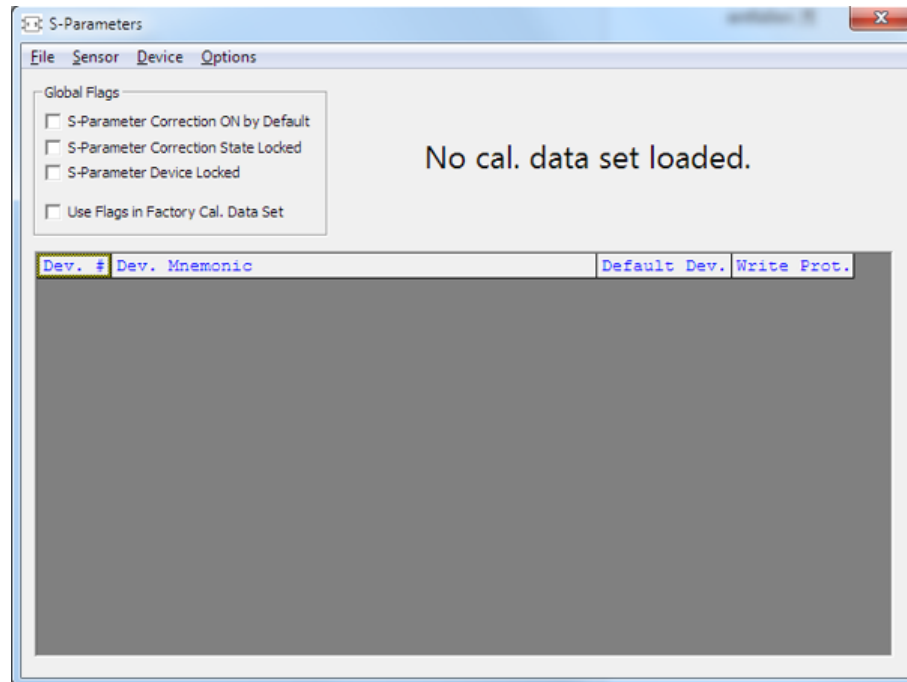


Figure 8-10: S-Parameters dialog

Menu bar.....	92
L File.....	92
L Sensor.....	93
L Device.....	93
L Options.....	93
L User Data.....	93
L Remote.....	93
L Show Cal. Data.....	93
Global Flags.....	94
L S-Parameter Correction ON by Default.....	94
L S-Parameter Correction State Locked.....	94
L S-Parameter Device Locked.....	94
L Use Flags in Factory Cal. Data Set.....	95
Device table.....	95

Menu bar

Contains the following submenus.

File ← Menu bar

Provides options for loading and saving calibration data files, see:

- ["To change the S-parameter data"](#) on page 97
- ["To load an uncertainty parameter file"](#) on page 98

Sensor ← Menu bar

Provides options for loading and saving calibration data directly from or to the sensor, see:

- ["To load a calibration data set from a power sensor"](#) on page 95
- ["To save the calibration data to the sensor"](#) on page 99

Device ← Menu bar

Provides functions for editing the table of S-parameter devices.

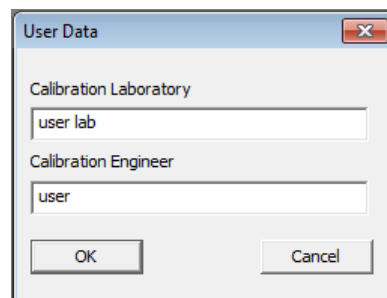
Options ← Menu bar

Provides functions for editing user data, changing remote control timeouts, and displaying calibration data as plain text.

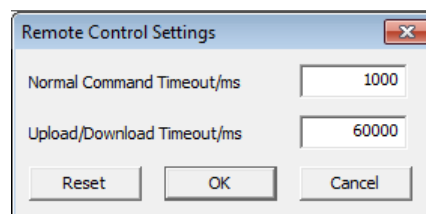
User Data ← Options ← Menu bar

Opens the "User Data" dialog.

Here, you can enter the name of the calibration laboratory and the calibration engineer that is stored in the calibration data set if changes are made.

**Remote ← Options ← Menu bar**

Opens the "Remote Control Settings" dialog. It is normally not necessary to change timeouts.

**Show Cal. Data ← Options ← Menu bar**

Displays the content of the calibration data set that has been loaded either from a file or directly from a sensor as a plain text.

You can copy the text output to the clipboard by clicking "Copy to Clipboard."

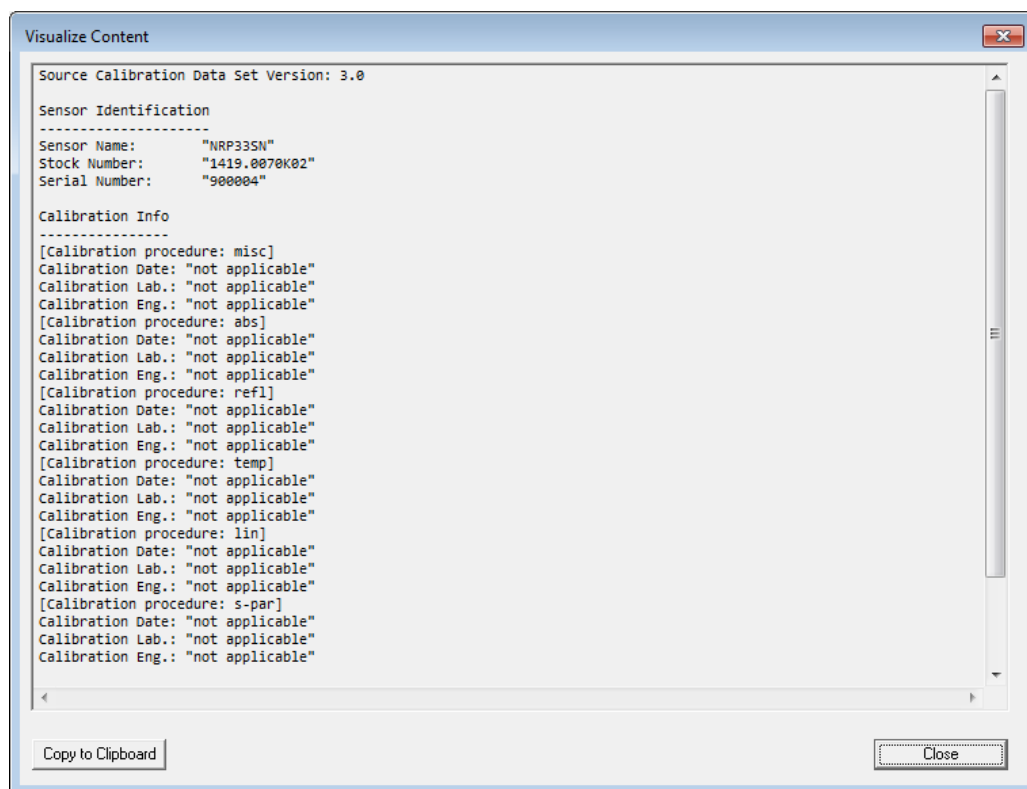


Figure 8-11: Example

Global Flags

Groups the settings for the power sensor behavior regarding S-parameter corrections.

S-Parameter Correction ON by Default ← Global Flags

If this option is enabled, the S-parameter correction is activated automatically when the sensor is started.

S-Parameter Correction State Locked ← Global Flags

If enabled, the state that is selected with "S-Parameter Correction ON by Default" is locked and cannot be changed using:

- `[SENSe<Sensor>:]CORRection:SPDevice:StAtE`
- R&S NRP2 base unit

S-Parameter Device Locked ← Global Flags

If enabled, the S-parameter device that is selected as the default device in the table of S-parameter devices is locked and cannot be changed using:

- `[SENSe<Sensor>:]CORRection:SPDevice:SElect`
- R&S NRP2 base unit

The default S-parameter device is the S-parameter device that you have selected when the power sensor is started.

Use Flags in Factory Cal. Data Set ← Global Flags

Available if the power sensor supports two different calibration data sets:

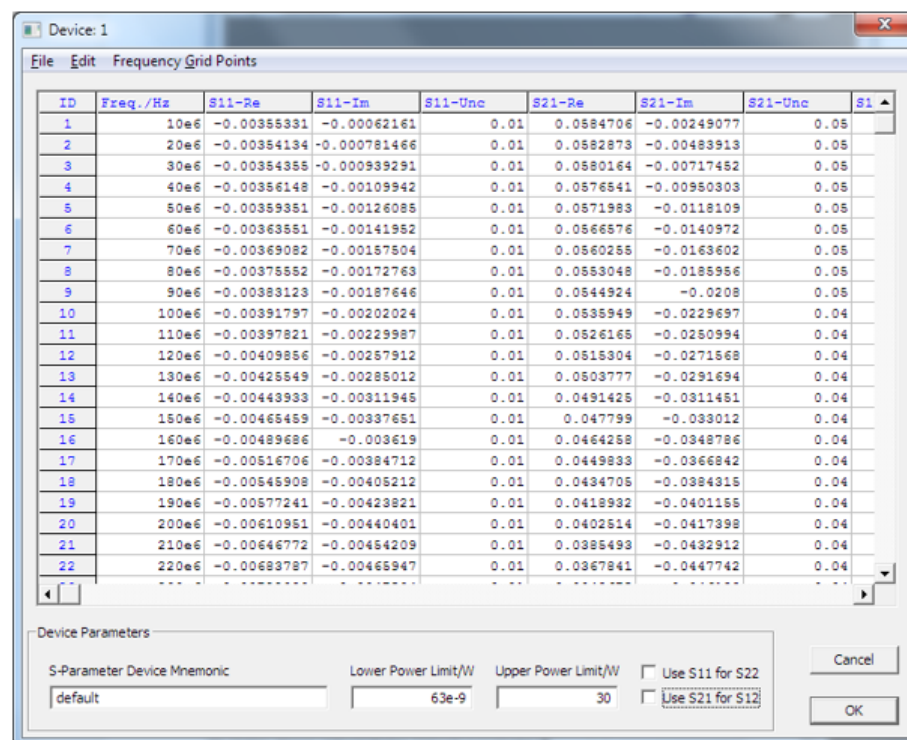
- Factory calibration data set containing all factory calibration data.
- User calibration data set that contains the S-parameter devices you have loaded.

Note: After you have added S-parameter devices and configured the global flags, disable this option. Otherwise, it is not possible to enable S-parameter correction because the flags in the factory calibration data set prevent it.

Device table

Shows a list of all S-parameter devices that are available in the calibration data set.

If you double-click an entry, a dialog for the device is opened that allows to import, export, and edit S-parameter data. See ["To change the S-parameter data"](#) on page 97.



Performing Configuration Tasks

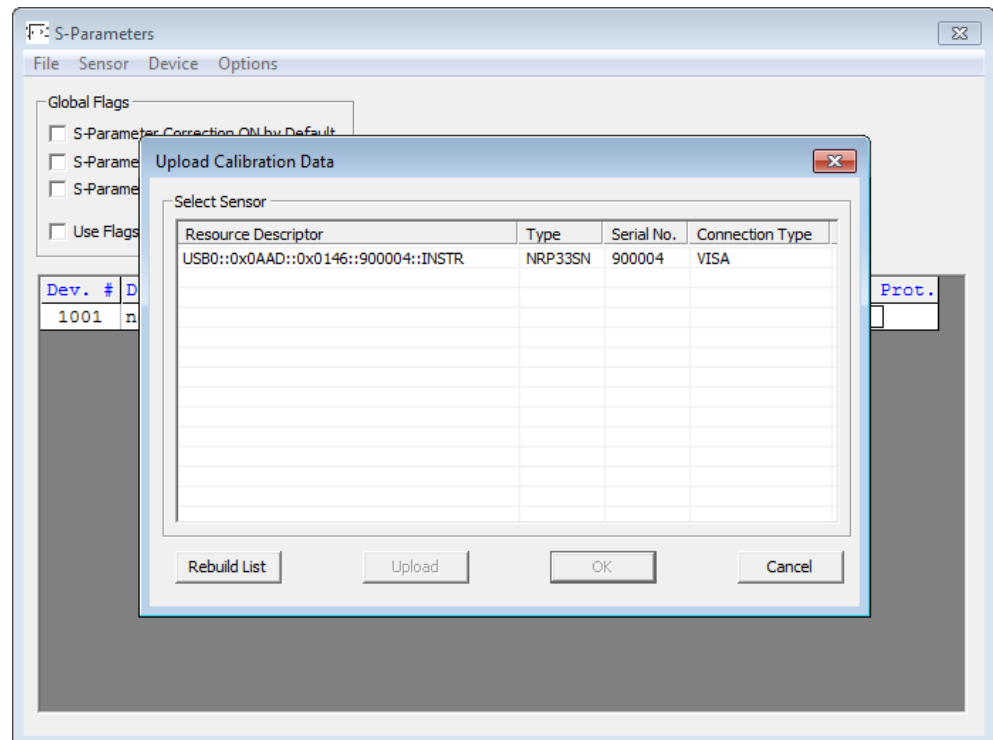
In this chapter, different configuration tasks performed with the power sensor and the "S-Parameter Update Multi" tool are described.

To load a calibration data set from a power sensor

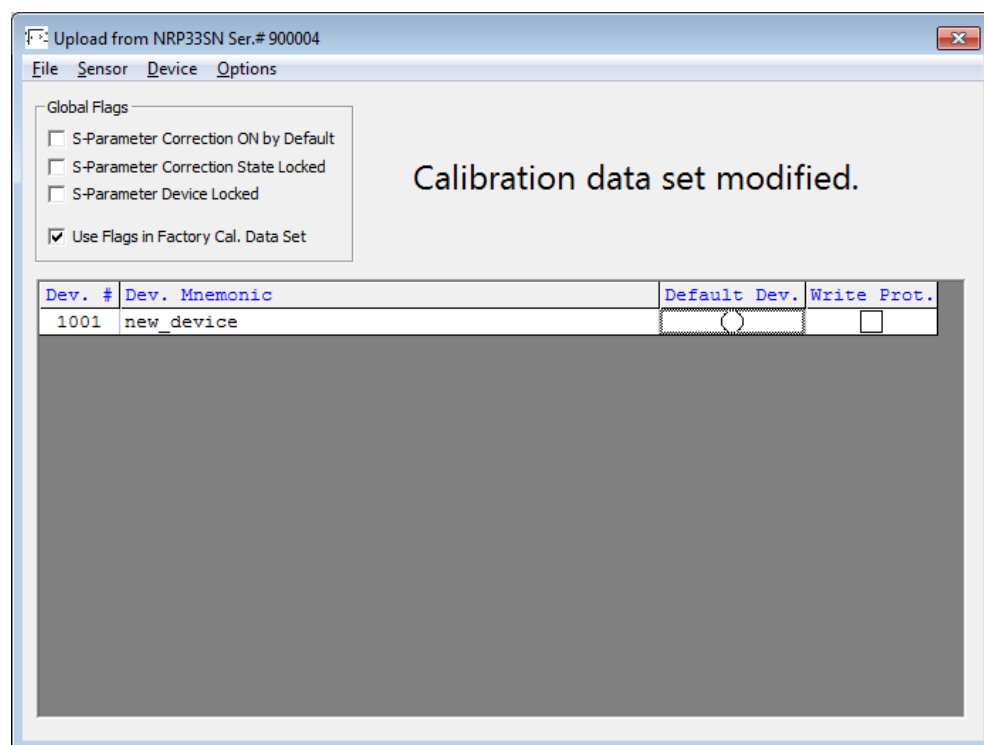
Prerequisites: The power sensor is connected to the computer and a connection is established.

1. Open the "S-Parameter Update Multi" program.
2. Select "Sensor" > "Load Calibration Data".

The "Upload Calibration Data" dialog opens. It shows a list of the available sensors.



3. If you cannot find your power sensor in the list, for example because of reconnecting the power sensor, you can reload the list by clicking "Rebuild List".
4. Click "Upload" to load calibration data from the power sensor.
After the upload is finished, the "OK" button is enabled.
5. Click "OK" to apply the changes.
If you want to discard the changes, exit the dialog by clicking "Cancel".
After a successful upload, the name and serial number is shown in the name of the main dialog.

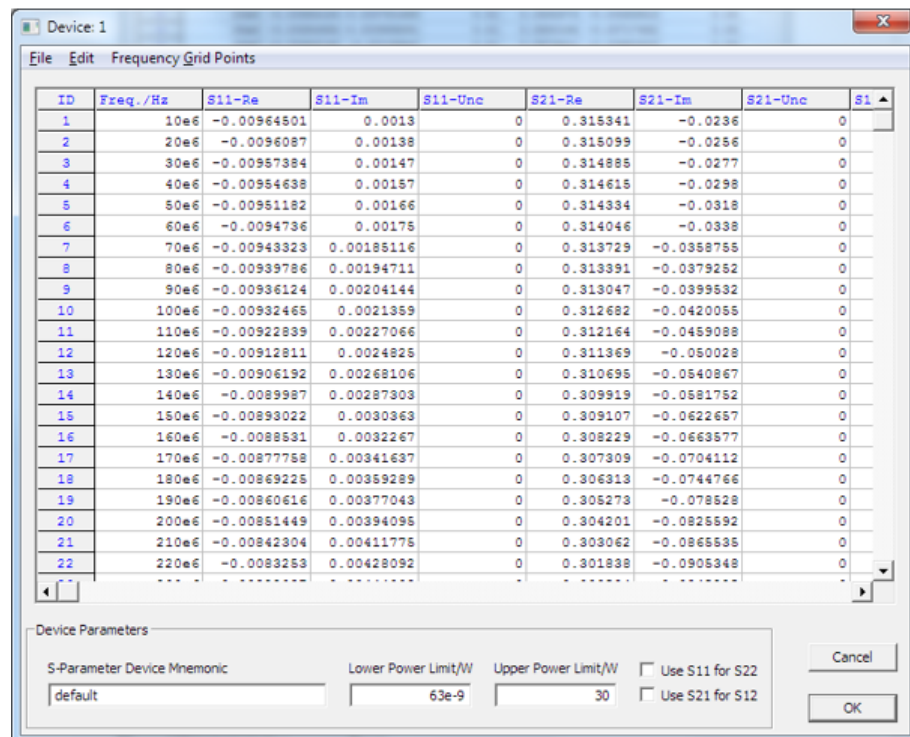


6. Create a backup of the calibration data set before making any changes.
Select "File" > "Save Calibration Data".
A dialog opens where you can select the location to save the calibration data.

To change the S-parameter data

1. In the device table, double-click an entry. See also "Device table" on page 95.
2. Select "File" > "Import S2P".
3. Select the *.S2P file you want to import, and confirm with "Open".

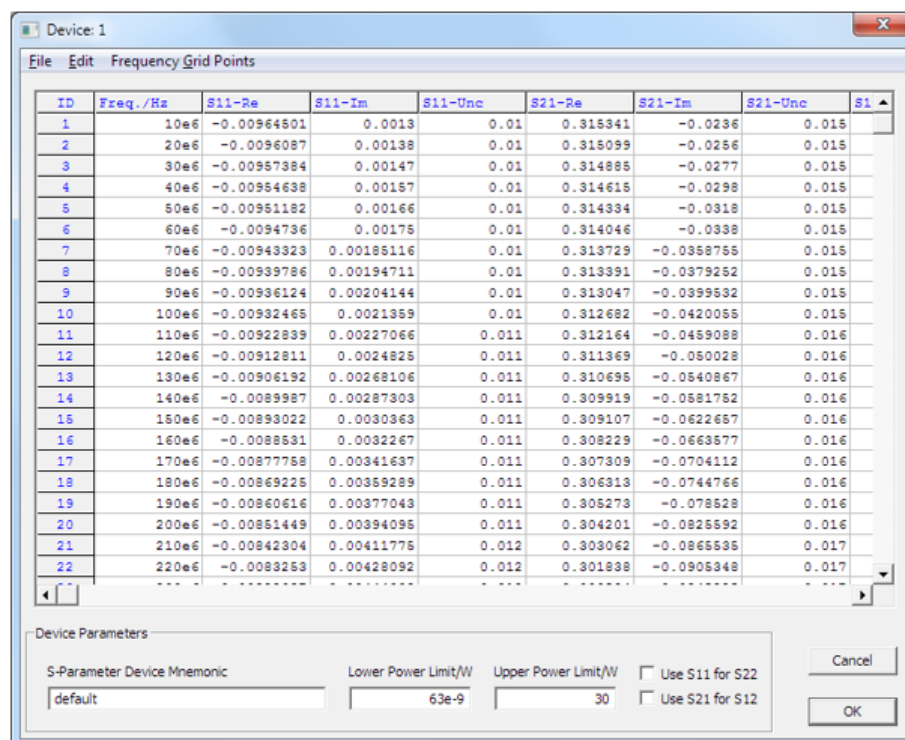
The data from the selected file is loaded in the device table.
All uncertainties are set to zero.



4. If needed, load uncertainty data. See ["To load an uncertainty parameter file"](#) on page 98.
5. Check the entries in the "S-Parameter Device Mnemonic", "Lower Power Limit/W" and "Upper Power Limit/W" fields and change them, if necessary.
For example, the lower and upper power limits are deduced from the power limits of the sensor itself and the minimum attenuation of the S-parameter device. If the "Upper Power Limit/W" entry is higher than the power dissipation rating of the attenuator, reduce it accordingly.
6. Click "OK" to apply the changes.
If you want to discard the change, click "Cancel".

To load an uncertainty parameter file

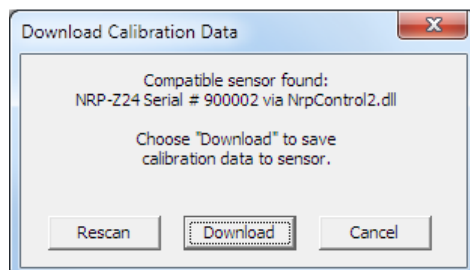
1. In the device table, double-click an entry. See also ["Device table"](#) on page 95.
2. Select "File" > "Import uncertainties".
3. Select the file you want to import, and confirm with "Open".
The data from the selected file is loaded in the device table.



To save the calibration data to the sensor

1. Select "Sensor" > "Save Calibration Data".

The "Download Calibration Data" dialog opens.



2. Confirm that the correct power sensor is selected by clicking "Download".

After a successful transfer of the data to the power sensor, a confirmation message is displayed.

The sensor can be used with the new S-parameter data.

S2P Measurement Data Files

S2P files are human-readable text files that contain header information and the complex S-parameters of the device under test in columns. This chapter briefly describes the format of the S2P file.

An S2P measurement data file has the following structure (square brackets indicate that the enclosed content is optional):

- **Option line**

The option line has the format `#[<frequency unit>][<parameter>][<format>][<R n>]`, where:

- `#`
Identifies the option line.
- `<frequency unit>`
Possible values are Hz, kHz, MHz or GHz. If a frequency unit is not specified, GHz is implicitly assumed.
- `<parameter>` For S-parameter files. If a parameter is not specified, S is implicitly assumed.
- `<format>`
Possible values are MA (linear magnitude and phase in degree), DB (magnitude in dB, phase in degree) or RI (real and imaginary part). If a format is not specified, MA is implicitly assumed.
- `<R n>`
R is optional and followed by the reference impedance in Ω . If no entry is made, R50 is implicitly assumed.

The option line therefore reads:

`# [HZ | KHZ | MHZ | GHZ] [S] [MA | DB | RI] [R 50]`.

- **Measurement frequencies**

The measurement frequencies are listed in ascending order and are specified as follows:

f_i $s_{11}(f_i)$ $s_{21}(f_i)$ $s_{12}(f_i)$ $s_{22}(f_i)$

where f_i is the i-th frequency and $s_{jk}(f_i)$ is the display format as specified in the option line:

- `| $s_{jk}(f_i)$ | arg $s_{jk}(f_i)$`
Display format for linear magnitude and phase in degree (MA)
- `20.lg| $s_{jk}(f_i)$ | arg $s_{jk}(f_i)$`
Display format for magnitude in dB and phase in degree (DB)
- `Re| $s_{jk}(f_i)$ | Im| $s_{jk}(f_i)$`
Display format for real and imaginary part (RI)

- **Comments**

Any line starting with an exclamation mark (!) is interpreted as a comment line.

Uncertainty Data Files

An uncertainty data file has the following structure (square brackets indicate that the enclosed content is optional):

- **Option line**

The option line has the format `#[<frequency unit>]<parameter>[<format>][<R n>]`, where:

- `#`
Identifies the option line.
- `<frequency unit>`
Possible values are Hz, kHz, MHz or GHz. If a frequency unit is not specified, GHz is implicitly assumed.
- `<parameter>`

U must be specified for uncertainty data files. If a parameter is not specified, S is implicitly assumed and as a result an error message is triggered.

- *<format>*
This value is ignored in uncertainty measurement files. The entry is therefore irrelevant.
- *<R n>*
 R is optional and followed by the reference impedance in Ω . If no entry is made, $R50$ is implicitly assumed.

The option line therefore reads:

[HZ | KHZ | MHZ | GHZ] U [MA | DB | RI] [R 50].

- **Measurement frequencies**

The measurement frequencies are listed in ascending order and are specified as follows:

$f_i \text{ } unc[s_{11}(f_i)] \text{ } unc[s_{21}(f_i)] \text{ } unc[s_{12}(f_i)] \text{ } unc[s_{22}(f_i)]$

where f_i is the i -th frequency and $unc[s_{jk}(f_i)]$ is the uncertainty of the S -parameters that is forwarded as follows:

- As extended absolute uncertainty ($k = 2$) for the magnitude of reflection parameters s_{11} and s_{22}
- As extended uncertainties ($k = 2$) in dB for the magnitude of transmission parameters s_{21} and s_{12}

- **Comments**

Any line starting with an exclamation mark (!) is interpreted as a comment line.

8.9 Calibrating/Zeroing the Power Sensor

Zeroing removes offset voltages from the analog circuitry of the sensors, so that there are only low powers displayed when there is no power applied. The zeroing process may take more than 8 seconds to complete.

Zeroing is recommended if:

- The temperature has varied by more than 5 K.
- The sensor has been replaced.
- No zeroing was performed in the last 24 hours.
- Signals of very low power are to be measured, for instance, if the expected measured value is less than 10 dB above the lower measurement range limit.



Turn off all test signals before zeroing. An active test signal during zeroing causes an error.

Remote commands:

CALibration:DATA.....	102
CALibration:DATA:LENGth?.....	102
CALibration:USER:DATA.....	102
CALibration:USER:DATA:LENGth?.....	102
CALibration<Channel>:ZERO:AUTO.....	103

CALibration:DATA <caldata>

Writes a binary calibration data set in the memory of the sensor.

Parameters:

<caldata> <block_data>

CALibration:DATA:LENGth?

Queries the length in bytes of the calibration data set currently stored in the flash memory. Programs that read out the calibration data set can use this information to determine the capacity of the buffer memory required.

Example: CAL:DATA:LENG?
 Query
 57392
 Response

Usage: Query only

CALibration:USER:DATA <caldata>

Transfers the user calibration data set, which mainly contains S-parameter sets for user-specific devices. The query returns the data as it was downloaded to the power sensor before.

After downloading of a new user calibration data set to the power sensor, the current S-parameter correction settings become invalid. Safe operation of the power sensor is only possible if the SElect and STATe commands are repeated after download. See also:

- [SENSe<Sensor>:]CORRection:SPDevice:STATe
- [SENSe<Sensor>:]CORRection:SPDevice:SElect

Parameters:

<caldata> <block_data>

CALibration:USER:DATA:LENGth?

Queries the length of the user calibration data block.

Usage: Query only

CALibration<Channel>:ZERO:AUTO <state>

Performs zero calibration.

Turn off all test signals before zeroing. An active test signal during zeroing causes an error.

While zero calibration is in progress, no queries or other setting commands are allowed, since the command is synchronous. Any communication attempt can run into a timeout.

After zero calibration, query the static error queue (**SYSTem:SERRor?**). The following responses are possible:

- 0
No error, the zero calibration was successful.
- -240
Warning, zero calibration failed. See also the example.

Suffix:

<Channel> 1...4
Measurement channel if more than one channel is available.

Parameters:

<state> **ONCE**
Only valid parameter for this command.
0
Return value if no calibration is in progress.
***RST:** OFF

Example:

*CLS
CAL1:ZERO:AUTO ONCE
Performs zeroing. Takes several seconds.
SYST:SERR?
Query
-240
Response: Warning; Zero Calibration failed; Results Degrading.

8.10 Testing the Power Sensor

The selftest allows a test of the internal circuitry of the sensor.



Do not apply a signal to the sensor while the selftest is running. If the selftest is carried out with a signal being present, error messages can erroneously be output for the test steps *Offset Voltages* and/or *Noise Voltages*.

TEST:SENSor? [<Item>]

Triggers a selftest of the sensor. In contrast to ***TST?**, this command returns detailed information that you can use for troubleshooting.

Query parameters:

<Item> String

Usage:

Query only

8.11 Configuring the System

The `SYSTem` subsystem contains a series of commands for general functions that do not directly affect the measurement.



The `SYSTem:COMMunicate:NETWork:...` commands are only available for the R&S NRP LAN power sensors.

Remote commands:

<code>SYSTem:COMMunicate:NETWork:REStart</code>	105
<code>SYSTem:COMMunicate:NETWork:RESet</code>	105
<code>SYSTem:COMMunicate:NETWork:STATus?</code>	105
<code>SYSTem:COMMunicate:NETWork[:COMMOn]:DOMain</code>	105
<code>SYSTem:COMMunicate:NETWork[:COMMOn]:HOSTname</code>	106
<code>SYSTem:COMMunicate:NETWork:IPADdress</code>	106
<code>SYSTem:COMMunicate:NETWork:IPADdress:GATeway</code>	106
<code>SYSTem:COMMunicate:NETWork:IPADdress:INFO?</code>	107
<code>SYSTem:COMMunicate:NETWork:IPADdress:MODE</code>	107
<code>SYSTem:COMMunicate:NETWork:IPADdress:SUBNet:MASK</code>	107
<code>SYSTem:DFPPrint<Channel>?</code>	107
<code>SYSTem:ERRor:ALL?</code>	108
<code>SYSTem:ERRor:CODE:ALL?</code>	108
<code>SYSTem:ERRor:CODE[:NEXT]?</code>	108
<code>SYSTem:ERRor:COUNT?</code>	108
<code>SYSTem:ERRor[:NEXT]?</code>	109
<code>SYSTem:FWUPdate</code>	109
<code>SYSTem:FWUPdate:STATus?</code>	109
<code>SYSTem:HELP:HEADers?</code>	110
<code>SYSTem:HELP:SYNTax?</code>	110
<code>SYSTem:HELP:SYNTax:ALL?</code>	110
<code>SYSTem:INFO?</code>	110
<code>SYSTem:INITialize</code>	111
<code>SYSTem:LANGuage</code>	111
<code>SYSTem:LED:COLor</code>	112
<code>SYSTem:LED:MODE</code>	112
<code>SYSTem:MINPower?</code>	112
<code>SYSTem:PARAmeters?</code>	113
<code>SYSTem:PARAmeters:DELTA?</code>	113
<code>SYSTem:PRESet</code>	113
<code>SYSTem:REBoot</code>	113
<code>SYSTem:REStart</code>	113
<code>SYSTem:RUTime</code>	113

SYSTem:SERRor?	114
SYSTem:SERRor:LIST:ALL?	114
SYSTem:SERRor:LIST[:NEXT]?	114
SYSTem:SUTime	115
SYSTem:TLEVels?	115
SYSTem:TRANsaction:BEgin	115
SYSTem:TRANsaction:END	115
SYSTem[:SENSor]:NAME	115
SYSTem:VERSion?	116

SYSTem:COMMunicate:NETWork:REStart

Effective only for the R&S NRP LAN power sensors.

Restarts the network connection to the DUT, that means terminates the connection and sets it up again.

Example: `SYST:COMM:NETW:REST`

Usage: Event

SYSTem:COMMunicate:NETWork:RESet

Effective only for the R&S NRP LAN power sensors.

Resets the LAN network settings to the default values.

Usage: Event

SYSTem:COMMunicate:NETWork:STATus?

Effective only for the R&S NRP LAN power sensors.

Queries the network configuration state.

Example: `SYST:COMM:NETW:STAT?`
 Query
 UP
 Response: The network is active.

Usage: Query only

SYSTem:COMMunicate:NETWork[:COMMON]:DOMain <domain>

Effective only for the R&S NRP LAN power sensors.

Sets the domain of the network.

Parameters:

<domain>

Example: `SYST:COMM:NETW:COMM:DOM 'ABC.DE'`
 Sets *ABC.DE* as domain of the network.

SYSTem:COMMunicate:NETWork[:COMMON]:HOSTname <hostname>

Effective only for the R&S NRP LAN power sensors.

Sets the individual hostname of the sensor.

In a LAN that uses a DNS server (domain name system server), you can access each connected instrument using a unique hostname instead of its IP address. The DNS server translates the hostname to the IP address. Using a hostname is especially useful if a DHCP server is used, as a new IP address can be assigned each time the instrument is restarted.

The sensor performs the change of the hostname immediately after the command is sent. For this purpose, the sensor restarts its connection to the network, which can take several seconds. During this time, you cannot address the sensor. After the restart, you can only address the sensor using the newly set hostname.

Note: It is recommended that you do not change the default hostname to avoid problems with the network connection. However, if you change the hostname, be sure to use a unique name.

Parameters:

<hostname>

Example:

```
SYST:COMM:NETW:COMM:HOST  
'powersensor-2nd-floor'
```

Sets *powersensor-2nd-floor* as new hostname.

SYSTem:COMMunicate:NETWork:IPAddress <ipaddress>

Effective only:

- For the R&S NRP LAN power sensors.
- If **SYSTem:COMMunicate:NETWork:IPAddress:MODE** is set to **STATIC**.

Sets the IP address of the sensor.

Parameters:

<ipaddress>

Example:

```
SYST:COMM:NETW:IPAD '192.168.10.29'
```

Sets *192.168.10.29* as IP address.

SYSTem:COMMunicate:NETWork:IPAddress:GATeway <gateway>

Effective only:

- For the R&S NRP LAN power sensors.
- If **SYSTem:COMMunicate:NETWork:IPAddress:MODE** is set to **STATIC**.

Sets the IP address of the default gateway.

Parameters:

<gateway>

Example: `SYST:COMM:NETW:IPAD:GAT '192.168.10.254'`
Sets 192.168.10.254 as IP address of the default gateway.

SYSTem:COMMunicate:NETWork:IPADdress:INFO?

Effective only for the R&S NRP LAN power sensors.

Queries the network status information.

Usage: Query only

SYSTem:COMMunicate:NETWork:IPADdress:MODE <mode>

Effective only for the R&S NRP LAN power sensors.

Sets whether the IP address is assigned automatically or manually.

Parameters:

<mode> AUTO | STATic

AUTO

Assigns the IP address automatically, provided the network supports DHCP.

STATic

Enables assigning the IP address manually.

*RST: AUTO

Example: `SYST:COMM:NETW:IPAD:MODE AUTO`
The IP address is assigned automatically.

SYSTem:COMMunicate:NETWork:IPADdress:SUBNet:MASK <netmask>

Effective only:

- For the R&S NRP LAN power sensors.
- If `SYSTem:COMMunicate:NETWork:IPADdress:MODE` is set to STATic.

Sets the subnet mask.

Parameters:

<netmask>

Example: `SYST:COMM:NETW:IPAD:SUBN:MASK '255.255.255.0'`
Sets 255.255.255.0 as subnet mask.

SYSTem:DFPRint<Channel>?

Reads the footprint file of the sensor.

Suffix:

<Channel> 1...4
Measurement channel if more than one channel is available.

Usage: Query only

SYSTem:ERRor:ALL?

Queries all unread entries in the error/event queue and removes them from the queue.

The response is a comma-separated list in first out order, each entry consisting of the error number and a short description of the error.

Positive error numbers are instrument-dependent. Negative error numbers are reserved by the SCPI standard.

Example: SYST:ERR:ALL?
Query
0, "No error"
Response

Usage: Query only

SYSTem:ERRor:CODE:ALL?

Queries all unread entries in the error/event queue and removes them from the queue. Only the error numbers are returned.

Example: SYST:ERR:CODE:ALL?
Query
0
Response: No errors have occurred since the error queue was last read out.

Usage: Query only

SYSTem:ERRor:CODE[:NEXT]?

Queries the oldest entry in the error queue and then deletes it. Only the error number is returned.

Example: SYST:ERR:CODE?
Query
0
Response: No errors have occurred since the error queue was last read out.

Usage: Query only

SYSTem:ERRor:COUNt?

Queries the number of entries in the error queue.

Example: SYST:ERR:COUN?
Query
1
Response: One error has occurred since the error queue was last read out.

Usage: Query only

SYSTem:ERRor[:NEXT]?

Queries the error/event queue for the oldest entry and removes it from the queue. The response consists of an error number and a short description of the error.

Positive error numbers are instrument-dependent. Negative error numbers are reserved by the SCPI standard.

Example: SYST:ERR?
 Query
 0, 'no error'
 Response: No errors have occurred since the error queue was last read out.

Usage: Query only

SYSTem:FWUPdate <fwudata>

Loads new operating firmware into the power sensor. Rohde & Schwarz provides the update file. For further details, see [Chapter 6, "Firmware Update"](#), on page 35.

If you want to integrate a firmware update function in an application, see the example given in [Chapter 6.2.2, "Using Remote Control"](#), on page 37.

Setting parameters:

<fwudata> <block_data>
 Definite length arbitrary block data containing the direct copy of the binary *.rsu file in the following format:
 #
 Single digit indicating how many digits follow to specify the size of the binary file.
 Number that specifies the size of the binary file.
 Binary data
 0x0a as appended delimiter for line feed

Usage: Setting only

SYSTem:FWUPdate:STATus?

Reads the result of the firmware update performed using [SYSTem:FWUPdate](#) on page 109.

While a firmware update is in progress, the LED of the sensor flashes in bright white color. When the firmware update is completed, you can read the result.

The result of the query is a readable string.

Example: SYST:FWUP:STAT?
 Query
 "Success"
 Response

Usage: Query only

SYSTem:HELP:HEADers? [<Item>]

Returns a list of all SCPI commands supported by the sensor.

Query parameters:

<Item> <block_data>

Usage: Query only

SYSTem:HELP:SYNTax? [<Item>]

Queries the relevant parameter information for the specified SCPI command.

Query parameters:

<Item>

Example: SYST:HELP:SYNT? 'sens:aver:coun'

Usage: Query only

SYSTem:HELP:SYNTax:ALL?

Queries the implemented SCPI commands and their parameters. Returns the result as a block data.

Usage: Query only

SYSTem:INFO? [<item>]

Queries information about the system.

If queried without parameters, the command returns all available information in the form of a list of strings separated by commas.

If you want to query specific information, add a query parameter.

Query parameters:

<item> "<string>" with the following values:

Manufacturer

Type

Stock Number

Serial

SW Build

MAC Address

Hostname

IP Address

Domain

Subnetmask

Gateway

Mode

Status
Sensor Name
Technology
Function
MinPower
MaxPower
MinFreq
MaxFreq
Resolution
Impedance
Coupling
Uptime
Cal. Misc.
Cal. Abs.
Cal. Refl.
Cal. Temp.
Cal. Lin.
Cal. S-Para.
Cal. S-Para. (User)
SPD Mnemonic
Cal. Due Date
Certificate No
Limit
TestLimit
TestLimit pd

Usage: Query only

SYSTem:INITialize

Sets the sensor to the standard state, i.e. the default settings for all test parameters are loaded in the same way as with [*RST](#). The sensor then outputs a complete list of all supported commands and parameters. With the command, the remote-control software can automatically adapt to the features of different types of sensors with different functionality.

Usage: Event

SYSTem:LANGuage <language>

Selects an emulation of a different command set.

Parameters:

<language> SCPI
 *RST: SCPI

SYSTem:LED:COLor <color>

Sets the color and the flash code of the system status LED, if the operating mode of the LED is set to **USER** ([SYSTem:LED:MODE](#)).

Parameters:

<color> Hexadecimal code described as
 0x0krrggbb
 with
 k = 0: steady on; k = 1: slowly flashing; k = 2: fast flashing
 rr = red
 gg = green
 bb = blue
 Range: 0x00 to 0x02FFFFFF
 *RST: 0x00A0A0A0

Example:

`SYST:LED:MODE USER`
 Sets the system status LED operating mode to user.

`SYST:LED:COL #H01a00000`
 The LED flashes slowly in red.

`SYSTem:LED:MODE SENSor`
 Sets the system status LED operating mode back to the sensor internal settings.

SYSTem:LED:MODE <mode>

Sets whether the color of the system status LED is controlled by the sensor firmware or by the user settings.

For more information, see [SYSTem:LED:COLor](#).

Parameters:

<mode> USER | SENSor
 *RST: SENSor

SYSTem:MINPower?

Queries the lower power measurement limit.

This value changes if [\[SENSe<Sensor>:\]CORRection:SPDevice:STATe](#) is set to **ON**. The lower measurement limit refers to the sensor or to the combination of a sensor and the components connected ahead of it.

Use this query to determine a useful resolution for the result display near the lower measurement limit.

Usage: Query only

SYSTem:PARAmeters?

Lists all commands with default values, limits and ranges.

Usage: Query only

SYSTem:PARAmeters:DELTA?

Lists all commands that differ from the defined default status set by *RST on page 45.

The commands are output with default values, limits and ranges.

Usage: Query only

SYSTem:PRESet

Triggers a sensor reset.

The command essentially corresponds to the *RST command, with the exception that the settings of the following commands are persistently held:

INITiate:CONTinuous

SENSe:AVERage:TCONtrol

SENSe:TRACe:AVERage:TCONtrol

Usage: Event

SYSTem:REBoot

Reboots the sensor.

Usage: Event

SYSTem:REStart

Restarts the firmware of the R&S NRP18S-xx.

Usage: Event

SYSTem:RUTime <update_time>

Effective only in the NRP legacy mode.

Sets the result update time. That is the maximum rate in which the power sensor can output measurement results. Relevant only in continuous measurement mode (INITiate:CONTinuous ON).

Parameters:

<update_time> Range: 0.0 to 10.0
 *RST: 0.1
 Default unit: Seconds

SYSTem:SERRor?

Queries the next static error, if available. Static errors are more severe than normal error conditions, which can be queried using `SYSTem:ERRor[:NEXT]?`.

Normal errors result from, for example, unknown commands or syntax errors and generally affect a single parameter or setting. Static errors, as a rule, prevent the execution of normal measurements.

Static errors occur when you select conflicting settings. This could, for example, occur in timeslot measurement mode with the following settings:

- Width of a timeslot: 100 μ s
- Exclude time at the start of the slot: 40 μ s
- Exclude time at the end of the slot: 60 μ s

Then there is "nothing left" to be measured, and a static error appears.

Usage: Query only

SYSTem:SERRor:LIST:ALL?

Returns a list of all static errors that have occurred but have already been resolved. For example, an overload of a short duration.

Example: `SYST:SERR:LIST:ALL?`
Response: 0,"reported at uptime:2942; notice;
 auto-averaging exceeded maximum time;
 Notification",0,"removed at uptime:2944;
 notice; auto-averaging exceeded maximum time;
 Notification".

Usage: Query only

SYSTem:SERRor:LIST[:NEXT]?

Queries the list of all static errors that have occurred but have already been resolved for the oldest entry and removes it from the queue. The response consists of an error number and a short description of the error.

Example: `SYST:SERR:LIST?`
Query
 0,"reported at uptime:2942; notice;
 auto-averaging exceeded maximum time;
 Notification"
Response

Usage: Query only

SYSTem:SUTime <update_time>

Effective only in the NRP legacy mode.

Sets the status update time. That is the maximum rate in which the power sensor can output measurement results. Relevant only in continuous measurement mode (**INITiate:CONTinuousON**).

Parameters:

<update_time> Range: 0.0 to 10.0
 *RST: 10e-3
 Default unit: Seconds

SYSTem:TLEvels?

Queries the possible power test levels of the sensor.

Usage: Query only

SYSTem:TRANsaction:BEIn

Starts a series of settings.

Usage: Event

SYSTem:TRANsaction:END

Ends a series of settings.

Usage: Event

SYSTem[:SENSor]:NAME <sensormname>

Sets the name of the sensor according to your requirements. The specified name is displayed in the web user interface of the network sensors.

The sensor name that you specify here is independent from the hostname of the sensor. However, if the sensor name is not specified, it defaults to the hostname.

Example:

SYST:NAME "InputModule-X5"

Results in a display as shown in [Figure 8-12](#).

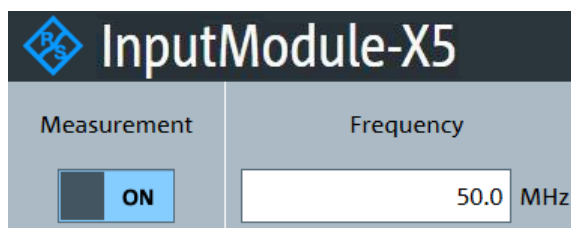


Figure 8-12: Sensor Name

Parameters:

<sensorname>

SYSTem:VERSIon?

Queries the SCPI version that the command set of the sensor complies with.

Example: SYST:VERS?
 Query
 1999.0
 Response: SCPI version from 1999.

Usage: Query only

8.12 Using the Status Register

Further information:

- [Chapter 10.3, "Status Reporting System"](#), on page 135

Contents:

- [General Status Register Commands](#)..... 116
- [Reading Out the CONDition Part](#)..... 117
- [Reading Out the EVENT Part](#)..... 117
- [Controlling the ENABLE Part](#)..... 118
- [Controlling the Negative Transition Part](#)..... 118
- [Controlling the Positive Transition Part](#)..... 118

8.12.1 General Status Register Commands

- [STATus:PRESet](#)..... 116
- [STATus:QUEue\[:NEXT\]?](#)..... 117

STATus:PRESet

Resets the edge detectors and **ENABLe** parts of all registers to a defined value.

Usage: Event

STATus:QUEue[:NEXT]?

Queries the most recent error queue entry and deletes it.

Positive error numbers indicate sensor specific errors. Negative error numbers are error messages defined by SCPI.

If the error queue is empty, the error number 0, "No error", is returned.

Usage: Query only

8.12.2 Reading Out the CONDition Part

Further information:

- [Chapter 10.3.2, "Structure of a SCPI Status Register"](#), on page 136

STATus:DEvIce:CONDition?**STATus:OPERation:CALibrating:CONDition?****STATus:OPERation:CONDition?****STATus:OPERation:LLFail:CONDition?****STATus:OPERation:MEASuring:CONDition?****STATus:OPERation:SENSe:CONDition?****STATus:OPERation:TRIGger:CONDition?****STATus:OPERation:ULFail:CONDition?****STATus:QUEStionable:CALibration:CONDition?****STATus:QUEStionable:CONDition?****STATus:QUEStionable:POWEr:CONDition?****STATus:QUEStionable:WINDow:CONDition?**

Usage: Query only

8.12.3 Reading Out the EVENT Part

Further information:

- [Chapter 10.3.2, "Structure of a SCPI Status Register"](#), on page 136

STATus:DEvIce[:EVENT]?**STATus:OPERation:CALibrating[:SUMMARY][:EVENT]?****STATus:OPERation[:EVENT]?****STATus:OPERation:LLFail[:SUMMARY][:EVENT]?****STATus:OPERation:MEASuring[:SUMMARY][:EVENT]?****STATus:OPERation:SENSe[:SUMMARY][:EVENT]?****STATus:OPERation:TRIGger[:SUMMARY][:EVENT]?****STATus:OPERation:ULFail[:SUMMARY][:EVENT]?****STATus:QUEStionable:CALibration[:SUMMARY][:EVENT]?****STATus:QUEStionable[:EVENT]?****STATus:QUEStionable:POWEr[:SUMMARY][:EVENT]?****STATus:QUEStionable:WINDow[:SUMMARY][:EVENT]?**

Usage: Query only

8.12.4 Controlling the ENABLE Part

Further information:

- [Chapter 10.3.2, "Structure of a SCPI Status Register"](#), on page 136

```

STATus:DEvIce:ENABle <value>
STATus:OPERation:CALibrating:ENABle <value>
STATus:OPERation:ENABle <value>
STATus:OPERation:LLFail:ENABle <value>
STATus:OPERation:MEASuring:ENABle <value>
STATus:OPERation:SENSe:ENABle <value>
STATus:OPERation:TRIGger:ENABle <value>
STATus:OPERation:ULFail:ENABle <value>
STATus:QUESTionable:CALibration:ENABle <value>
STATus:QUESTionable:ENABle <value>
STATus:QUESTionable:POWer:ENABle <value>
STATus:QUESTionable:WINDow:ENABle <value>
Parameters:
<value>          *RST:      0
  
```

8.12.5 Controlling the Negative Transition Part

Further information:

- [Chapter 10.3.2, "Structure of a SCPI Status Register"](#), on page 136

```

STATus:DEvIce:NTRansition <value>
STATus:OPERation:CALibrating:NTRansition <value>
STATus:OPERation:NTRansition <value>
STATus:OPERation:LLFail:NTRansition <value>
STATus:OPERation:MEASuring:NTRansition <value>
STATus:OPERation:SENSe:NTRansition <value>
STATus:OPERation:TRIGger:NTRansition <value>
STATus:OPERation:ULFail:NTRansition <value>
STATus:QUESTionable:CALibration:NTRansition <value>
STATus:QUESTionable:NTRansition <value>
STATus:QUESTionable:POWer:NTRansition <value>
STATus:QUESTionable:WINDow:NTRansition <value>
Parameters:
<value>          *RST:      0
  
```

8.12.6 Controlling the Positive Transition Part

Further information:

- [Chapter 10.3.2, "Structure of a SCPI Status Register"](#), on page 136

```

STATus:DEvIce:PTRansition <value>
STATus:OPERation:CALibrating:PTRansition <value>
  
```

STATus:OPERation:PTRansition <value>
STATus:OPERation:LLFail:PTRansition <value>
STATus:OPERation:MEASuring:PTRansition <value>
STATus:OPERation:SENSe:PTRansition <value>
STATus:OPERation:TRIGger:PTRansition <value>
STATus:OPERation:ULFail:PTRansition <value>
STATus:QUESTionable:CALibration:PTRansition <value>
STATus:QUESTionable:PTRansition <value>
STATus:QUESTionable:POWer:PTRansition <value>
STATus:QUESTionable:WINDow:PTRansition <value>
Parameters:
<value> *RST: 65535

9 Performing Measurement Tasks - Programming Examples

If you install the optional software development kit (SDK) of the R&S NRP Toolkit, programming examples are provided. See [Chapter 5.1, "R&S NRP Toolkit"](#), on page 18.

Under Windows, these examples are installed under:

```
C:\ProgramData\Rohde-Schwarz\NRP-Toolkit-SDK\examples
```

This chapter gives programming examples for measurement tasks performed with the series power sensors.

9.1 Performing the Simplest Measurement

The simplest way to obtain a result is to use the following sequence of commands:

```
*RST
INITiate
FETCh?
```

The `*RST` sets the continuous average mode.

`INITiate` initiates the measurement.

After `*RST`, the trigger system is set to `TRIGger:SOURce IMMEDIATE`. That means the power sensor starts measuring when the measurement is started without waiting for a trigger condition.

After the measurement has been completed, `FETCh<Sensor>[:SCALar] [:POWER] [:AVG] ?` delivers the result to the output queue from which it can be fetched.

9.2 Performing the Fastest Measurement in Continuous Average Mode

The fastest way to obtain results for different continuous measurements is described in this chapter.

9.2.1 Untriggered Fast Unchopped Continuous Average Measurement

This example, written in pseudo code, shows how to set up and execute an untriggered, fast unchopped continuous average measurement.

See also `[SENSe<Sensor>:] [POWER:] [AVG:] FAST` on page 72.

```
write( 'INIT:CONT OFF' )
write( 'ABORT' )
```


Performing the Fastest Measurement in Continuous Average Mode

```
write( '*RST' )

# Enable fast unchopped continuous average measurement
write( 'SENS:POW:AVG:FAST ON' )

# Define output format (float)
write( 'FORM:DATA REAL,32' )

# Select the trigger condition. Immediate means, that the sensor
# starts measuring when the measurement is started.
write( 'TRIG:SOUR IMM' )

# Select the maximum possible buffer size
BUFFER_SIZE_MAX = query( 'BUFF:SIZE? MAX' )
write( 'BUFF:SIZE ' + BUFFER_SIZE_MAX )
write( 'BUFF:STAT ON' )

# In this setting, trigger count needs to be the same as buffer size
write( 'TRIG:COUN ' + BUFFER_SIZE_MAX )

# Smallest aperture window is 10 us, resulting in 100000 meas/sec
write( 'SENS:POW:AVG:APER 10e-6' )

# Any errors occurred?
query( 'SYST:ERR:ALL?' )

# Start the configured (= untriggered) continuous measurement
write( 'INIT:CONT ON' )

# Let the sensor measure for 5 seconds
timeEnd = time.now() + 5.0
numData = 0

while (time.now() < timeEnd )
{
    # If there is any result in the buffer --> read it
    if ( query( 'BUFF:COUN?' ) > 0 )
    {
        result = queryBinary( 'BUFF:DATA?' )
        numData = numData + result.size
    }
}

# Stop the continuous measurement
utilDeviceIO.DeviceWrite( instrument, 'INIT:CONT OFF' )
```

9.2.2 Triggered Fast Unchopped Continuous Average Measurement

This example, written in pseudo code, shows how to set up and execute a fast unchopped continuous average measurement. The measurement is triggered on *each* pulse of a periodic input signal with 10 μ s period.

See also [SENSe<Sensor>:] [POWer:] [AVG:] FAST on page 72.

```

write( 'INIT:CONT OFF' )
write( 'ABORT' )
write( '*RST' )

# Enable fast unchopped continuous average measurement
write( 'SENS:POW:AVG:FAST ON' )

# Define output format (float)
write( 'FORM:DATA REAL,32' )

# Trigger on signal (here 0 dBm pulses with 100 kHz pulse freq.)
write( 'TRIG:SOUR INT' )
write( 'TRIG:LEV -15 DBM' )
write( 'TRIG:HYST 1' )

# Select the maximum possible buffer size
BUFFER_SIZE_MAX = query( 'BUFF:SIZE? MAX' )
write( 'BUFF:SIZE ' + BUFFER_SIZE_MAX )
write( 'BUFF:STAT ON' )

# In this setting, trigger count needs to be the same as buffer size
write( 'TRIG:COUN ' + BUFFER_SIZE_MAX )

# Smallest aperture window of the sensor is 10 us. However, for
# the fast measurement, you can set the aperture as low as 8 us in order
# to reliably detect each rising edge of a pulsed signal.
# In fact in triggered measurement, the aperture time should be 1 us less than the
# pulse period.
# With a 10 us periodic pulse input, this results in
# continuously acquiring 100000 meas/sec
write( 'SENS:POW:AVG:APER 8.5e-6' )

# Any errors occurred?
query( 'SYST:ERR:ALL?' )

# Start the configured (= triggered) continuous measurement
write( 'INIT:CONT ON' )

# Let the sensor measure for 10 seconds
timeEnd = time.now() + 10.0
numData = 0

while (time.now() < timeEnd )

```

```

{
    # If there is any result in the buffer --> read it
    if ( query( 'BUFF:COUN?' ) > 0 )
    {
        result = queryBinary( 'BUFF:DATA?' )
        numData = numData + result.size
    }
}

# Stop the continuous measurement
utilDeviceIO.DeviceWrite( instrument, 'INIT:CONT OFF' )

```

9.3 Performing a Buffered Continuous Average Measurement

This example, written in pseudo code, shows how to set up and execute a buffered continuous average measurement.

```

//Select whether using
//    'BUS Trigger'      --> true
// or 'EXT Trigger'     --> false
bool    bUseBUSTrigger = true;

// Use the first NRP series sensor which is found
if ( VI_SUCCESS == SENSOR.openFirstNrpSensor( "USB?::0X0AAD::?*::INSTR" ) )
{
    //Start with a clean state
    SENSOR.write( "*RST" );

    // Auto Averaging OFF and set Average Count = 4
    SENSOR.write( "SENS:AVER:COUN:AUTO OFF" );
    SENSOR.write( "SENS:AVER:COUN 4" );

    // Select the trigger source
    if ( bUseBUSTrigger )
    {
        // We want to use '*TRG' to trigger a single physical measurement
        SENSOR.write( "TRIG:SOUR BUS" );
    }
    else
    {
        // We get trigger pulses on the external input (SMB-type connector)
        SENSOR.write( "TRIG:SOUR EXT2" );
    }
    // Auto-Trigger OFF
    SENSOR.write( "TRIG:ATR:STAT OFF" );

    // Configure a buffered measurement

```

Performing a Buffered Continuous Average Measurement

```
// Buffer size is randomly selected to 17
SENSOR.write( "SENS:BUFF:SIZE 17" );
SENSOR.write( "SENS:BUFF:STAT ON" );
SENSOR.write( "TRIG:COUN 17" );

// Read out all errors / Clear error queue
SENSOR.query( "SYST:ERR:ALL?", szBuf, sizeof( szBuf ) );
printf( szBuf );

// Start a 'single' buffered measurement
// Since 17 trigger-counts have been configured,
// the 'single' buffered measurement, which becomes
// initiated by INIT:IMM, is not over until
// 17 physical measurements have been triggered
SENSOR.write( "INIT:IMM" );

// The end of a physical measurement can be recognized
// by a transistion to 'NOT MEASURING' which is a
// negative transistion on bit 1
SENSOR.write( "STAT:OPER:MEAS:NTR 2" );
SENSOR.write( "STAT:OPER:MEAS:PTR 0" );

// Collect 17 physical measurements
for ( int i = 0; i < 17; i++ )
{
    // As a pre-condition: clear the event register by reading it
    int iDummy;
    SENSOR.query( "STAT:OPER:MEAS:EVEN?", &iDummy );

    // Trigger a single physical measurement; either by '*TRG'
    // command or by an externally supplied pulse on the SMB-type connector
    if ( bUseBUSTrigger )
        SENSOR.write( "*TRG" );

    // Wait until the measurement is done
    int iMeasEvent = 0;
    while ( iMeasEvent != 2 )
    {
        SENSOR.query( "STAT:OPER:MEAS:EVEN?", &iMeasEvent );
        iMeasEvent &= 2;
    }
    printf( "Triggered!\n" );
}
// All 17 physical measurement have been executed.
// That means, buffer is full and can be read
SENSOR.query( "FETCH?", szBuf, sizeof( szBuf ) );
printf( szBuf );
```

```
}
```

9.4 Performing Trace Measurements

```
*RST

//Set the sensor's operation mode to trace
SENSe:FUNCTION "XTIME:POWeR"

//Set the carrier frequency
SENSe:FREQuency 1.8e9

//Set the number of points for the trace measurement
//Using 500 points usually represents a good compromise
//between USB transfer speed and resolution
SENSe:TRACe:POINTs 500

//Set the trace time. It influences the time length of a point since each point
//represents the time period resulting from the trace time divided by the
//number of points
SENSe:TRACe:TIME 20e-3

//Set the trace offset time to delay the start point
//of the trace measurement for the specified time
SENSe:TRACe:OFFSet:TIME 50e-6

//Configure the trigger
TRIGGer:SOURce INTernal
TRIGGer:SLOPe POSitive
TRIGGer:DTIME 0.001
TRIGGer:HYSTeresis 0.1
TRIGGer:LEVel 30e-6

//Enable and configure the averaging filter
SENSe:TRACe:AVERage:COUNT 8
SENSe:TRACe:AVERage:STATe ON

//Select the data output format
FORMat:DATA REAL

//Initiate the measurement
INITiate

//Query the measurement results
FETCh?
```

9.5 Trace Measurement with Synchronization to Measurement Complete

This example, written in pseudo code, shows how to set up and execute a trace measurement using a non-blocking technique.

The advantage of using the `FETCH?` command (as shown in the previous example) is, that `FETCH?` waits (blocks) until a measurement result is available. However, this behavior can lead to situations where an application blocks for a longer time (until timeout). For example, if a trigger is missing and thus no results are ever becoming available.

For certain applications, especially interactive ones, it is not the desired behavior that you have to wait until a (probably long) timeout occurs. In these cases, start a measurement and then enter a loop to poll the sensor until the measurement is ready and the results can safely be retrieved. For such applications, it is recommended to use the status system of the sensor to find out whether the measurement is ready. The advantage of this approach is that the polling loop can be exited/canceled at any time and the application stays operable (i. e. does not block).

```
// basic setup, similar to the previous example
write( "*RST" );
write( "SENS:FUNC \"XTIM:POW\"" );
write( "SENS:FREQ 1.8e9" );
write( "SENS:TRAC:POIN 500" );
write( "SENS:TRAC:TIME 20e-3" );
write( "TRIG:SOUR INT" );
write( "TRIG:SLOP POS" );
write( "TRIG:DTIM 0.001" );
write( "TRIG:HYST 0.1" );
write( "TRIG:LEV 30e-6" );
write( "SENS:TRAC:AVER:COUN 8" );
write( "SENS:TRAC:AVER:STAT ON" );

// configuring the event system to recognize the
// end of measurement (i.e. a negative transition
// of bit 1 in the meas operation register)
write( "STAT:OPER:MEAS:NTR 2" );
write( "STAT:OPER:MEAS:PTR 0" );

// resetting the event information by an initial readout
int iEvent = 0;
query( "STAT:OPER:MEAS:EVENT?", &iEvent );

// Now starting the measurement
write( "INIT:IMM" );

bool bMeasReady = false;

// poll until measurement is ready...
```

```
// (this loop could also check for cancel-requests
// from the user or other break conditions)
while ( ! bMeasReady )
{
    query( "STAT:OPER:MEAS:EVEN?", &iEvent );
    bMeasReady = ((iEvent & 0x02) != 0);

    if ( ! bMeasReady )
        sleep( 1 );
}

if ( bMeasReady )
{
    query( "SENS:TRAC:DATA?", bufResult );

    // further process the result in 'bufResult'...
    // ::
    // ::
}
```

10 Remote Control Basics

• Remote Control Interfaces and Protocols.....	128
• SCPI Command Structure.....	129
• Status Reporting System.....	135

10.1 Remote Control Interfaces and Protocols

For remote control, communication between the R&S NRP18S-xx power sensors and the controlling host is established based on various interfaces and protocols.

Depending on the sensor type, the power sensors support different interfaces for remote control.

- R&S NRP18S-xx power sensors are always accessed using USB.

Table 10-1 describes the protocols that are supported for each interface.

Table 10-1: Remote control interfaces and protocols

Interface	Supported by	Protocols, VISA ^{*)} address string and Library
USB	All power sensors	<ul style="list-style-type: none"> • USBTMC USB::<vendor id="">::<product ID>::<serial number>[:INSTR] VISA</vendor> • NRP legacy protocol

(*) VISA is a standardized software interface library providing input and output functions to communicate with instruments. A VISA installation on the controller is a prerequisite for remote control over USBTMC and LAN (when using VXI-11 or HiSLIP protocol) interfaces.

10.1.1 USB Interface

For remote control using USB connection, the computer and the power sensors must be connected via the USB interface. A USB connection requires the VISA library to be installed. VISA detects and configures the R&S power sensors automatically when the USB connection is established. Apart from the USBTMC driver (which comes with the installation of the R&S NRP Toolkit), you do not have to install a separate driver.

USB Test & Measurement Class Specification (USBTMC) is a protocol that is built on top of USB for communication with USB devices. It defines class code information of the sensor, that identifies its functionality to load the respective device driver. Using VISA library, it supports service request, triggers and other operations that are commonly found in GPIB devices.

Besides USBTMC, the NRP legacy protocol is available to ensure the compatibility of the R&S NRP18S-xx power sensors with the R&S NRP-Z series of power sensors. The usage of this protocol is not recommended for new applications.

The resource string represents an addressing scheme that is used to establish a communication session with the sensor. It is based on the sensor address and some instrument- and vendor-specific information.

USB Resource String

The syntax of the used USB resource string is:

```
USB::::<product ID>::[::INSTR]
```

where:

- <vendor ID> is the vendor ID for Rohde & Schwarz (0x0AAD)
- <product ID> is the product ID for the Rohde & Schwarz sensor
- <serial number> is the individual serial number on the rear of the sensor

Table 10-2: R&S NRP18S-xx USB product IDs

R&S NRP18S-xx power sensor	USB product ID
R&S NRP18S-10	0x0148
R&S NRP18S-20	0x014A
R&S NRP18S-25	0x014C

Example:

```
USB::0x0AAD::0x0148::100001
```

0x0AAD is the vendor ID for Rohde & Schwarz.

0x0148 is the product ID for the R&S NRP18S-10 power sensor.

100001 is the serial number of the particular power sensor.

10.2 SCPI Command Structure

SCPI commands - messages - are used for remote control. Commands that are not taken from the SCPI standard follow the SCPI syntax rules. The power sensor supports the SCPI version 1999. The SCPI standard is based on standard IEEE 488.2 and aims at the standardization of device-specific commands, error handling and the status registers.

SCPI commands consist of a so-called header and, usually, one or more parameters. The header and the parameters are separated by a "white space" (ASCII code 0 to 9, 11 to 32 decimal, e.g. blank). The headers can consist of several mnemonics (keywords). Queries are formed by appending a question mark directly to the header.

The commands can be either device-specific or device-independent (common commands). Common and device-specific commands differ in their syntax.

10.2.1 Syntax for Common Commands

Common (=device-independent) commands consist of a header preceded by an asterisk (*) and possibly one or more parameters.

Examples:

*RST	RESET	Resets the instrument.
*ESE	EVENT STATUS ENABLE	Sets the bits of the event status enable registers.
*ESR?	EVENT STATUS QUERY	Queries the contents of the event status register.
*IDN?	IDENTIFICATION QUERY	Queries the instrument identification string.

10.2.2 Syntax for Device-Specific Commands

Long and short form

The mnemonics feature a long form and a short form. The short form is marked by upper case letters here, to distinguish it from the long form, which constitutes the complete word. Either the short form or the long form can be entered; other abbreviations are not permitted.

Example:

INITiate:CONTinuous is equivalent to INIT:CONT or init:cont.



Case-insensitivity

Upper case and lower case notation only serves to distinguish the two forms in the manual, the instrument itself is case-insensitive.

Numeric suffixes

If a command can be applied to multiple instances of an object, e.g. specific channels or sources, the required instances can be specified by a suffix added to the command. Numeric suffixes are indicated by angular brackets (<1...4>, <n>, <i>) and are replaced by a single value in the command. Entries without a suffix are interpreted as having the suffix 1.



Different numbering in remote control

For remote control, the suffix can differ from the number of the corresponding selection used in manual operation. SCPI prescribes that suffix counting starts with 1. Suffix 1 is the default state and used when no specific suffix is specified.

Some standards define a fixed numbering, starting with 0. If the numbering differs in manual operation and remote control, it is indicated for the corresponding command.

Optional mnemonics

Some command systems permit certain mnemonics to be inserted into the header or omitted. These mnemonics are marked by square brackets in the description. The instrument must recognize the long command to comply with the SCPI standard. Some commands are considerably shortened by these optional mnemonics.

Example:

Definition: `INITiate[:IMMEDIATE]`

Command: `INIT:IMM` is equivalent to `INIT`

Parameters

Parameters must be separated from the header by a "white space". If several parameters are specified in a command, they are separated by a comma.

For a description of the parameter types, refer to [Chapter 10.2.3, "SCPI Parameters"](#), on page 131.

Special characters

	Parameters A vertical stroke in parameter definitions indicates alternative possibilities in the sense of "or". The effect of the command differs, depending on which parameter is used.
[]	Mnemonics in square brackets are optional and can be inserted into the header or omitted. Example: <code>INITiate[:IMMEDIATE]</code> <code>INIT:IMM</code> is equivalent to <code>INIT</code>
{ }	Parameters in curly brackets are optional and can be inserted once or several times, or omitted.

10.2.3 SCPI Parameters

Many commands are supplemented by a parameter or a list of parameters. The parameters must be separated from the header by a "white space" (ASCII code 0 to 9, 11 to 32 decimal, e.g. blank). Allowed parameters are:

- Numeric values
- Special numeric values
- Boolean parameters
- Text
- Character strings
- Block data

The parameters required for each command and the allowed range of values are specified in the command description.

Numeric values

Numeric values can be entered in any form, i.e. with sign, decimal point and exponent. Values exceeding the resolution of the instrument are rounded up or down. The mantissa can comprise up to 255 characters, the exponent must lie inside the value range -32000 to 32000. The exponent is introduced by an "E" or "e". Entry of the exponent alone is not allowed.

Units

For physical quantities, you can enter the unit. Only basic units are allowed and recognized, see [Table 10-3](#). If you omit the unit, the basic unit is used.

Table 10-3: Units

Noted default unit	Corresponding basic unit
Frequency	Hz
Seconds	s
Watts	W
Angle	degrees
Percent	PCT
DB	dB
DBM	dBm
DBUV	dBuV

Special numeric values

The texts listed below are interpreted as special numeric values. For a query, the numeric value is provided.

- MIN/MAX**
 MINimum and MAXimum denote the minimum and maximum value.
- DEF**
 DEFault denotes a preset value which has been stored in the non-variable memory. This value conforms to the default setting, as it is called by the *RST command.
- UP/DOWN**
 UP, DOWN increases or reduces the numeric value by one step. The step width can be specified via an allocated step command for each parameter which can be set via UP, DOWN.
- INF/NINF**
 INFinity, Negative INFinity (NINF) represent the numeric values 9.9E37 or -9.9E37, respectively. INF and NINF are only sent as instrument responses.
- NAN**
 Not a number (NAN) represents the value 9.91E37. NAN is only sent as an instrument response. This value is not defined. Possible causes are the division by zero, the subtraction of infinite from infinite and the representation of missing values.

Boolean parameters

Boolean parameters represent two states. The "ON" state (logically true) is represented by "ON" or a numeric value 1. The "OFF" state (logically untrue) is represented by "OFF" or the numeric value 0. The numeric values are provided as the response for a query.

Example:

Setting command: `SENSe:AVERage:COUNT:AUTO ON`

Query: `SENSe:AVERage:COUNT:AUTO?`

Response: `1`

Text parameters

Text parameters observe the syntactic rules for mnemonics, i.e. they can be entered using a short or long form. Like any parameter, they have to be separated from the header by a white space. For a query, the short form of the text is provided.

Example:

Setting command: `TRIGger:SLOPe POSitive`

Query: `TRIG:SLOP?`

Response: `POS`

Character strings

Enter strings always in quotation marks (' or ").

Example:

Setting command: `SENSe:FUNCtion "POWer:AVG"`

Query: `SENS:FUNC?`

Response: `"POWer:AVG"`

Block data

Block data is a format which is suitable for the transmission of large amounts of data. A command using a block data parameter has the following structure:

Example:

`SYSTem:HELP:SYNTax:ALL?`

Response: `#45168xxxxxxxx`

The ASCII character # introduces the data block. The next number indicates how many of the following digits describe the length of the data block. In the example, the 4 following digits indicate the length to be 5168 bytes. The data bytes follow. During the transmission of these data bytes all end or other control signs are ignored until all bytes are transmitted.

#0 specifies a data block of indefinite length. The use of the indefinite format requires a `NL^END` message to terminate the data block. This format is useful when the length of the transmission is not known or if speed or other considerations prevent segmentation of the data into blocks of definite length.

10.2.4 Overview of Syntax Elements

The following table provides an overview of the syntax elements:

:	The colon separates the mnemonics of a command. In a command line, the separating semicolon marks the uppermost command level.
;	The semicolon separates two commands of a command line. It does not alter the path.
,	The comma separates several parameters of a command.
?	The question mark forms a query.
*	The asterisk marks a common command.
' "	Quotation marks introduce a string and terminate it (both single and double quotation marks are possible).
#	The hash symbol introduces binary, octal, hexadecimal and block data. <ul style="list-style-type: none"> • Binary: #B10110 • Octal: #O7612 • Hex: #HF3A7 • Block: #21312
	A "white space" (ASCII-Code 0 to 9, 11 to 32 decimal, e.g. blank) separates the header from the parameters.

10.2.5 Structure of a Command Line

A command line can consist of one or several commands. It is terminated by one of the following:

- a <New Line>
- a <New Line> with EOI
- an EOI together with the last data byte

Several commands in a command line must be separated by a semicolon ";". If the next command belongs to a different command system, the semicolon is followed by a colon.

If the successive commands belong to the same system, having one or several levels in common, the command line can be abbreviated. To this end, the second command after the semicolon starts with the level that lies below the common levels. The colon following the semicolon must be omitted in this case.

Example:

```
TRIG:LEV 0.1mW;TRIG:DEL 3E-3
```

This command line contains two commands. Both commands are part of the `TRIG` command system, i.e. they have one level in common.

When abbreviating the command line, the second command begins with the level below `TRIG`. The colon after the semicolon is omitted. The abbreviated form of the command line reads as follows:

```
TRIG:LEV 0.1E-3;DEL 3E-3
```

A new command line always begins with the complete path.

Example:

```
TRIG:LEV 0.1E-3
```

```
TRIG:DEL 3E-3
```

10.2.6 Responses to Queries

A query is defined for each setting command unless explicitly specified otherwise. It is formed by adding a question mark to the associated setting command. According to SCPI, the responses to queries are partly subject to stricter rules than in standard IEEE 488.2.

- The requested parameter is transmitted without a header.
Example: TRIG:SOUR?, response: INT
- Maximum values, minimum values and all other quantities that are requested via a special text parameter are returned as numeric values.
- Numeric values are output without a unit. Physical quantities are referred to the basic units or to the units set using the `Unit` command. The response `3.5E9` for example stands for 3.5 GHz.
- Truth values (Boolean values) are returned as 0 (for OFF) and 1 (for ON).

Example:

Setting command: SENS:AVER:COUN:AUTO ON

Query: SENS:AVER:COUN:AUTO?

Response: 1

- Text (character data) is returned in a short form.

Example:

Setting command: TRIGger:SOURce INTernal

Query: TRIG:SOUR?

Response: INT

10.3 Status Reporting System

The status reporting system stores all information on the current operating state of the sensor, and on errors which have occurred. This information is stored in the status registers and in the error queue. You can query both with the commands of the `STATus` subsystem.

10.3.1 Hierarchy of the Status Registers

[Fig.10-1](#) shows the hierarchical structure of information in the status registers.

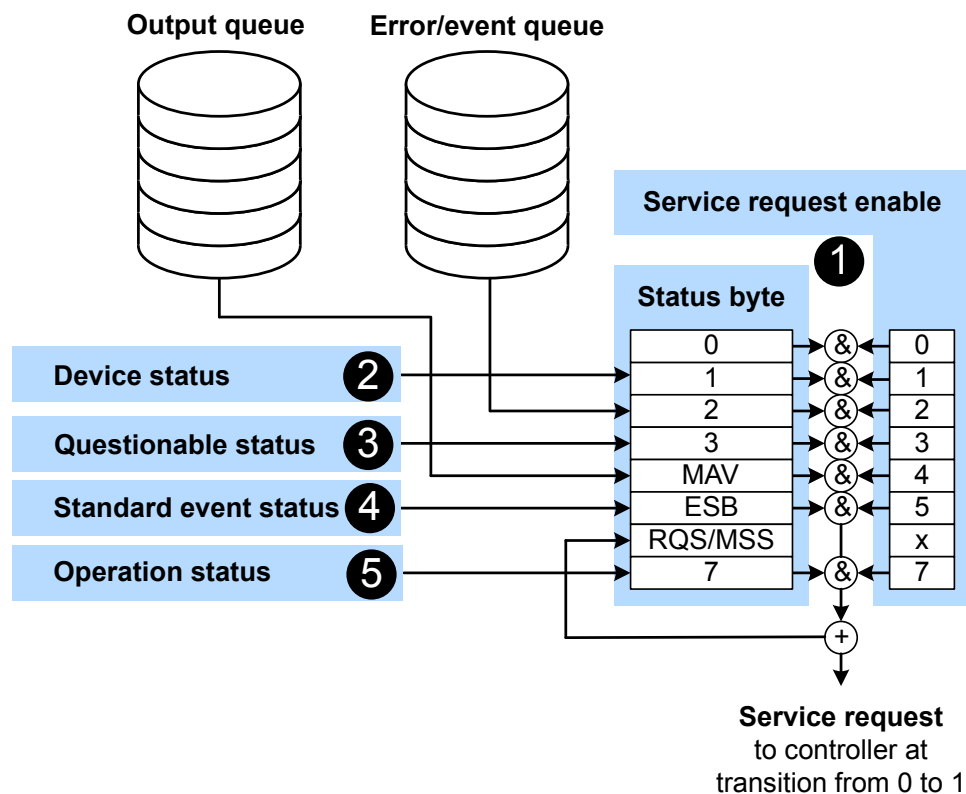


Figure 10-1: Status registers overview

- 1 = Chapter 10.3.3, "Status Byte (STB) and Service Request Enable Register (SRE)", on page 138
 2 = Chapter 10.3.5, "Device Status Register", on page 140
 3 = Chapter 10.3.6, "Questionable Status Register", on page 141
 4 = Chapter 10.3.7, "Standard Event Status and Enable Register (ESR, ESE)", on page 144
 5 = Chapter 10.3.8, "Operation Status Register", on page 145

The highest level is formed by the status byte register (STB) and the associated service request enable (SRE) register.

The status byte register (STB) receives its information from:

- Standard event status register (ESR)
- Associated standard event status enable register (ESE)
- SCPI-defined operation status register
- Questionable status register, which contains detailed information on the device.

10.3.2 Structure of a SCPI Status Register

Each SCPI register consists of five 16-bit registers that have different functions, see Figure 10-2. The individual bits are independent of each other, i.e. each hardware status is assigned a bit number which is the same for all five registers. Bit 15, the most-

significant bit, is set to 0 in all registers, thus preventing problems some controllers have with the processing of unsigned integers.

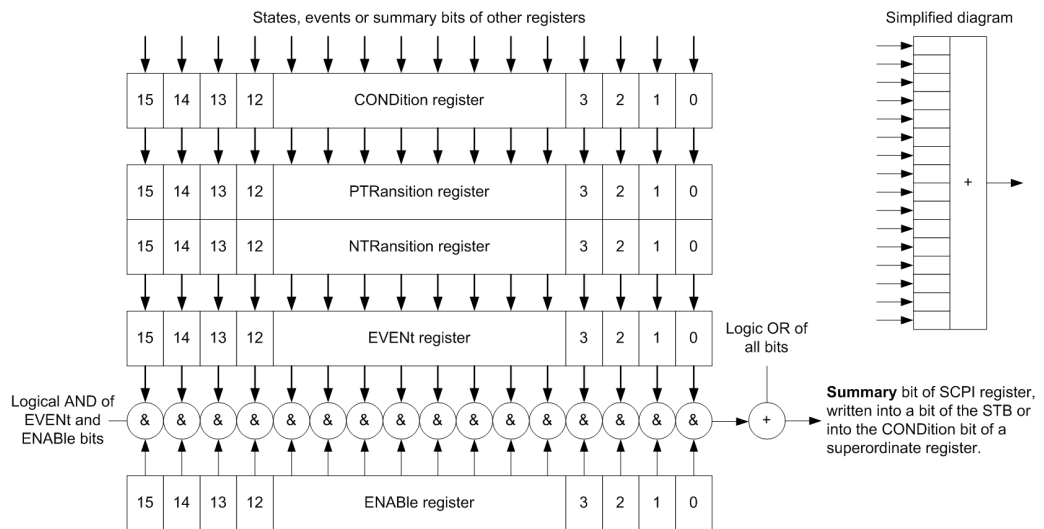


Figure 10-2: Standard SCPI status register

CONDition status register part

The five parts of a SCPI register have different properties and functions:

The **CONDition** part is written into directly by the hardware or the sum bit of the next lower register. Its contents reflect the current instrument status. This register part can only be read, but not written into or cleared. Its contents are not affected by reading.

PTRansition / NTRansition status register part

The two transition register parts define which state transition of the **CONDition** part (none, 0 to 1, 1 to 0 or both) is stored in the **EVENT** part.

The *Positive TRansition* part acts as a transition filter. When a bit of the **CONDition** part is changed from 0 to 1, the associated **PTR** bit decides whether the **EVENT** bit is set to 1.

- **PTR** bit = 1: The **EVENT** bit is set.
- **PTR** bit = 0: The **EVENT** bit is not set.

This part can be written into and read as required. Its contents are not affected by reading.

The *Negative TRansition* part also acts as a transition filter. When a bit of the **CONDition** part is changed from 1 to 0, the associated **NTR** bit decides whether the **EVENT** bit is set to 1.

- **NTR** bit = 1: The **EVENT** bit is set.
- **NTR** bit = 0: The **EVENT** bit is not set.

This part can be written into and read as required. Its contents are not affected by reading.

EVENT status register part

The `EVENT` part indicates whether an event has occurred since the last reading, it is the "memory" of the condition part. It only indicates events passed on by the transition filters. It is permanently updated by the instrument.

You can only read this part. Reading the register clears it. This part is often equated with the entire register.

ENABLE status register part

The `ENABLE` part determines whether the associated `EVENT` bit contributes to the sum bit (see below). Each bit of the `EVENT` part is "ANDed" with the associated `ENABLE` bit (symbol '&'). The results of all logical operations of this part are passed on to the sum bit via an "OR" function (symbol '+').

`ENABLE` bit = 0: The associated `EVENT` bit does not contribute to the sum bit.

`ENABLE` bit = 1: If the associated `EVENT` bit is 1, the sum bit is set to 1 as well.

You can read and write as required. Its contents are not affected by reading.

Sum bit

The sum bit is obtained from the `EVENT` and `ENABLE` part for each register. The result is then entered into a bit of the `CONDition` part of the higher-order register.

The instrument automatically generates the sum bit for each register. Thus an event can lead to a service request throughout all levels of the hierarchy.

10.3.3 Status Byte (STB) and Service Request Enable Register (SRE)

The status byte register is already defined in IEEE 488.2. It gives a rough overview of the sensor status, collecting information from the lower-level registers. It is comparable with the `CONDition` register of a SCPI defined register and is at the highest level of the SCPI hierarchy. Its special feature is that bit 6 acts as the summary bit of all other bits of the status byte register.

The status byte register is read by `*STB?` or a serial poll. The service request enable register is associated with the status byte register. The function of the service request enable register corresponds to that of the `ENABLE` register of the SCPI registers. Each bit of the status byte register is assigned a bit in the service request enable register. Bit 6 of the service request enable register is ignored. If a bit is set in the service request enable register and the associated bit in the status byte register changes from 0 to 1, a service request (SRQ) is generated on the IEC/IEEE bus. This service request triggers an interrupt in the controller configured for this purpose, and can be further processed by the controller.

Set and read the service request enable register using `*SRE`.

See [Figure 10-1](#).

Table 10-4: Used status byte bits and their meaning

Bit no.	Short description	Bit is set if
1	Device status register summary	A sensor is connected or disconnected or when an error has occurred in a sensor, depending on the configuration of the sensor status register. Chapter 10.3.5, "Device Status Register" , on page 140.
2	Error queue not empty	The error queue has an entry. If this bit is enabled by the service request enable register, each entry of the error queue generates a service request. An error can thus be recognized and specified in detail by querying the error queue. The query yields a conclusive error message. This procedure is recommended since it considerably reduces the problems of IEC/IEEE-bus control.
3	Questionable status register summary	An <code>EVENT</code> bit is set in the <code>QUESTionable</code> status register and the associated <code>ENABLE</code> bit is set to 1. A set bit denotes a questionable device status which can be specified in greater detail by querying the questionable status register. Chapter 10.3.6, "Questionable Status Register" , on page 141.
4	MAV Message available	A readable message is in the output queue. This bit can be used to automate reading of data from the sensor into the controller.
5	ESB Standard event status register summary	One of the bits in the standard event status register is set and enabled in the event status enable register. Setting this bit denotes a serious error which can be specified in greater detail by querying the standard event status register. Chapter 10.3.7, "Standard Event Status and Enable Register (ESR, ESE)" , on page 144.
6	MSS Master status summary	The sensor triggers a service request, which happens if one of the other bits of this register is set together with its enable bit in the service request enable register (SRE).
7	Operation status register summary	An <code>EVENT</code> bit is set in the operation status register and the associated <code>ENABLE</code> bit is set to 1. A set bit denotes that an action is being performed by the sensor. Information on the type of action can be obtained by querying the operation status register. Chapter 10.3.8, "Operation Status Register" , on page 145.

10.3.4 IST Flag and Parallel Poll Enable Register (PPE)

Similar to the service request (SRQ), the IST flag combines the complete status information in a single bit. It can be queried by a parallel poll or by `*IST?`.

The parallel poll enable register (PPE) determines which bits of the STB affect the IST flag. The bits of the STB are ANDed with the corresponding bits of the PPE; bit 6 is also used, in contrast to the service request enable register. The IST flag is obtained by ORing all results together.

Set and read the parallel poll enable register using `*PRE`.

10.3.5 Device Status Register

Sensor error summary	0	+
Sensor error	1	
Sensor error	2	
Sensor error	3	
Sensor error	4	
0	5	
0	6	
Legacy locked	7	
Reference PLL locked	8	
0	9	
0	10	
0	11	
0	12	
0	13	
0	14	
0	15	

Figure 10-3: Device status register

Querying the register:

- `STATus:DEvice:CONDition?`
- `STATus:DEvice[:EVENT]?`

Querying the static errors:

- `SYSTem:SERRor?`

Table 10-5: Used device status bits and their meaning

Bit no.	Short description	Bit is set if
0	Sum of SERR bits	Sum/combination of SERR bits 1 to 4.
1	SERR measurement not possible	
2	SERR erroneous results	Static error exists. The measurement result is possibly incorrect.
3	SERR warning	Static error exists. Status LED of the power sensor is slowly flashing red.
4	SERR critical	Critical static error exists. Status LED of the power sensor is fast flashing red.

Bit no.	Short description	Bit is set if
7	Legacy locked state	<p>The power sensor is locked in the NRP legacy mode. Via the SCPI channels (USBTMC or TCP/IP), only query commands can be sent, but no setting commands.</p> <p>When the first setting command is sent, the NRP legacy interface takes precedence over all other command channels. This bit is set to 1, and all other channels can only execute query commands. If a setting command is sent via a different channel, the power sensor indicates an error:</p> <p><i>-200, "Execution error; sensor in LEGACY mode"</i></p> <p>To leave this operating mode, close the NRP legacy channel. Either close the application which opened the NRP legacy channel or close at least the connection to the power sensor.</p>
8	Reference PLL locked state	<p>PLL for the clock reference is synchronized. The bit is useful when selecting an external clock source.</p> <p>The following states are possible:</p> <ul style="list-style-type: none"> • Internal clock ([SENSe<Sensor>:]ROSCillator:SOURce INT): <ul style="list-style-type: none"> – 1 (always) • External clock ([SENSe<Sensor>:]ROSCillator:SOURce EXT): <ul style="list-style-type: none"> – 1 if the sensor was able to synchronize with external clock – 0 if the sensor could not synchronize with external clock

10.3.6 Questionable Status Register

Contains information on questionable sensor states that occur if the power sensor is not operated in compliance with its specifications.

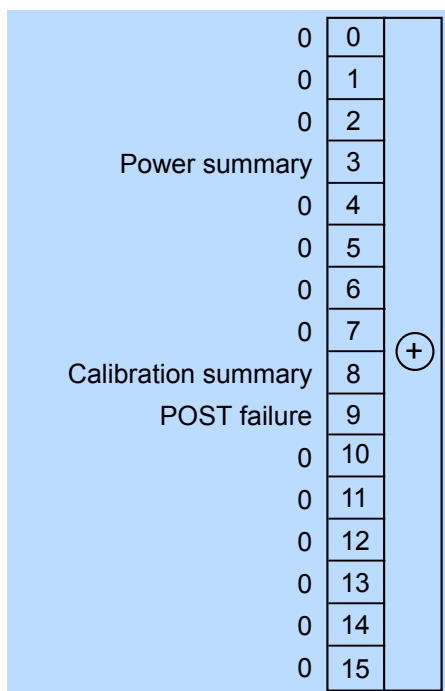


Figure 10-4: Questionable status register

Querying the register:

- `STATus:QUESTionable:CONDition?`
- `STATus:QUESTionable[:EVENT]?`

Table 10-6: Used questionable status bits and their meaning

Bit no.	Short description	Bit is set if
3	Power summary	Summary of Questionable Power Status Register exists.
8	Calibration summary	Summary of Questionable Calibration Status Register exists.
9	POST failure	Built-in test of the R&S NRP18S-xx that is carried out automatically upon power-up has generated an error.

10.3.6.1 Questionable Power Status Register

Contains information whether the measured power values are questionable.

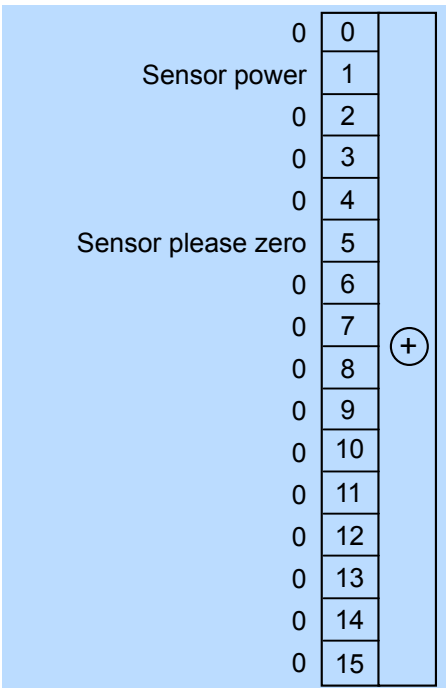


Figure 10-5: Questionable power status register

Querying the register:

- `STATUS:QUESTIONable:POWER:CONDition?`
- `STATUS:QUESTIONable:POWER[:SUMMARY][:EVENT]?`

Table 10-7: Used questionable power status bits and their meaning

Bit no.	Short description	Bit is set if
1	Sensor power	Measurement data of the power sensor is corrupt.
5	Sensor please zero	Zero correction for the power sensor is no longer correct. Perform a zero correction.

10.3.6.2 Questionable Calibration Status Register

Contains information whether the zeroing of the power sensor was successful.

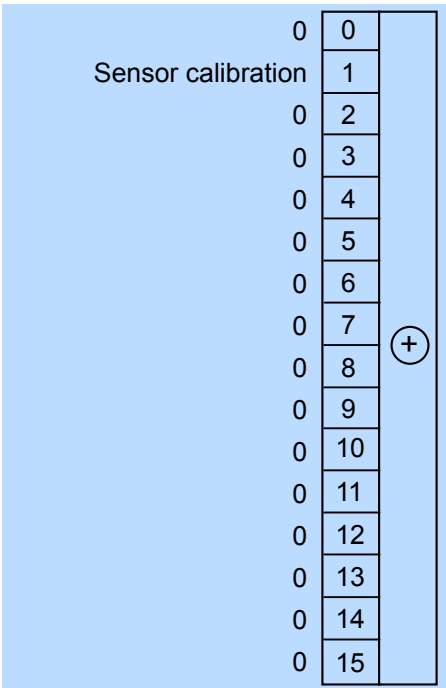


Figure 10-6: Questionable calibration status register

Querying the register:

- `STATUS:QUESTIONable:CALibration:CONDition?`
- `STATUS:QUESTIONable:CALibration[:SUMMARY][:EVENT]?`

Table 10-8: Used questionable calibration status bits and their meaning

Bit no.	Short description	Bit is set if
1	Sensor calibration	Zeroing of the power sensor was not successful.

10.3.7 Standard Event Status and Enable Register (ESR, ESE)

The `ESR` is already defined in the IEEE 488.2 standard. It is comparable to the `EVENT` register of a SCPI register. The standard event status register can be read out by `*ESR?`.

The `ESE` forms the associated `ENABLE` register. It can be set and read by `*ESE`.

Operation Complete	0	+
0	1	
Query Error	2	
Device-Dependent Error	3	
Execution Error	4	
Command Error	5	
User Request	6	
Power On	7	

Figure 10-7: Standard event status register (ESR)

Table 10-9: Used standard event status bits and their meaning

Bit no.	Short description	Bit is set if
0	Operation complete	All previous commands have been executed and *OPC is received.
2	Query error	The controller wants to read data from the sensor but has not sent a query, or it sends new commands to the sensor before it retrieves existing requested data. A frequent cause is a faulty query which cannot be executed.
3	Device-dependent error	A sensor-dependent error occurs. An error message with a number between -300 and -399 or a positive error number denoting the error in greater detail is entered in the error queue.
4	Execution error	The syntax of a received command is correct but the command cannot be executed due to various marginal conditions. An error message with a number between -200 and -300 denoting the error in greater detail is entered in the error queue.
5	Command error	An undefined command or a command with incorrect syntax is received. An error message with a number between -100 and -200 denoting the error in greater detail is entered in the error queue.
6	User request	The sensor is switched over to manual control.
7	Power on	The sensor is switched on.

10.3.8 Operation Status Register

Contains information on current operations, CONDition register, or operations performed since the last query, EVEnt register.

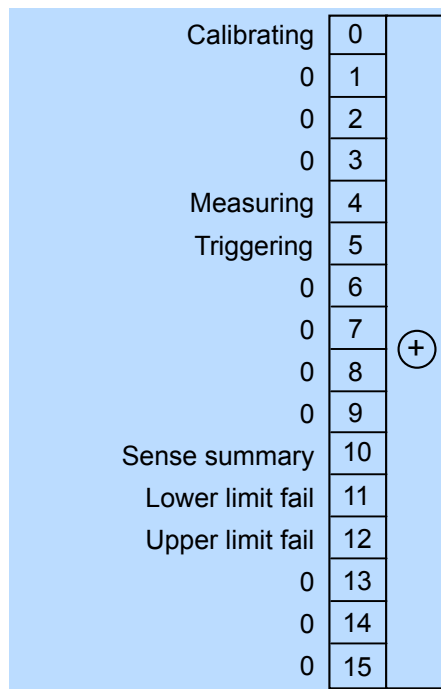


Figure 10-8: Operation status register

Querying the register:

- `STATus:OPERation:CONDition?`
- `STATus:OPERation[:EVENT]?`

Table 10-10: Used operation status bits and their meaning

Bit no.	Short description	Bit is set if
0	Calibrating	Summary of Operation Calibrating Status Register exists.
4	Measuring	Summary of Operation Measuring Status Register exists.
5	Triggering	Summary of Operation Trigger Status Register exists.
10	Sense summary	Summary of Operation Sense Status Register exists.
11	Lower limit fail	Summary of Operation Lower Limit Fail Status Register exists.
12	Upper limit fail	Summary of Operation Upper Limit Fail Status Register exists.

10.3.8.1 Operation Calibrating Status Register

The `CONDition` register contains information whether a power sensor is being calibrated. The `EVENT` register contains information whether a calibration was started or completed since the last query.

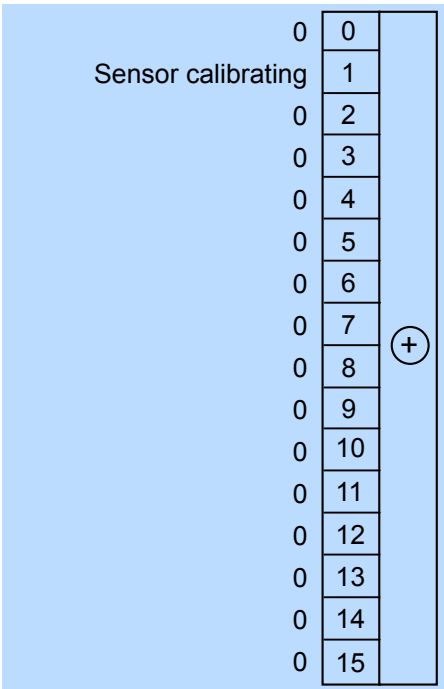


Figure 10-9: Operation calibrating status register

Querying the register:

- `STATUS:OPERation:CALibrating:CONDition?`
- `STATUS:OPERation:CALibrating[:SUMMARY][:EVENT]?`

Table 10-11: Used operation calibrating status bits and their meaning

Bit no.	Short description	Bit is set if
1	Sensor calibrating	Sensor is being calibrated.

10.3.8.2 Operation Measuring Status Register

The `CONDition` register contains information whether a power sensor is measuring. The `EVENT` register contains information whether a measurement was started or completed since the last query.

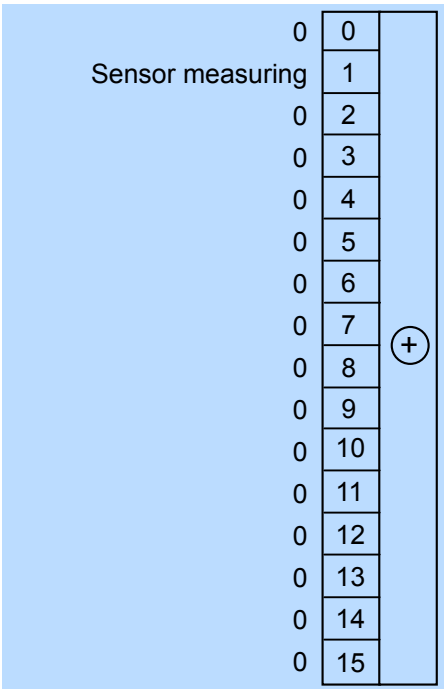


Figure 10-10: Operation measuring status register

Querying the register:

- `STATUS:OPERation:MEASuring:CONDition?`
- `STATUS:OPERation:MEASuring[:SUMMARY][:EVENT]?`

Table 10-12: Used operation measuring status bits and their meaning

Bit no.	Short description	Bit is set if
1	Sensor measuring	Sensor is measuring.

10.3.8.3 Operation Trigger Status Register

The `CONDition` register contains information whether a power sensor is waiting for a trigger event. The `EVENTt` register contains information whether the power sensor has been waiting for a trigger event since the last query.

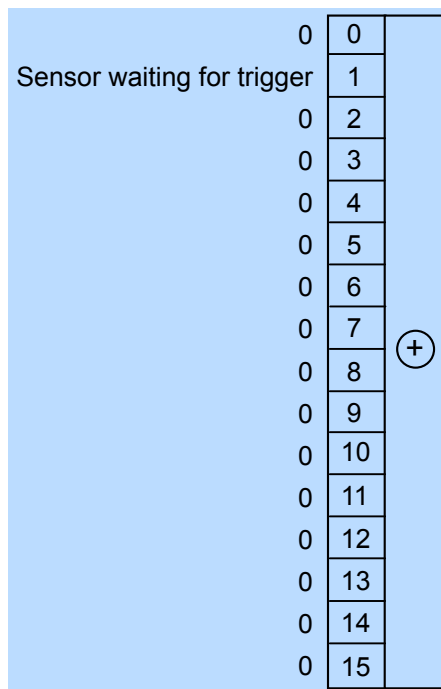


Figure 10-11: Operation trigger status register

Querying the register:

- `STATUS:OPERation:TRIGger:CONDition?`
- `STATUS:OPERation:TRIGger[:SUMMary][:EVENT]?`

Table 10-13: Used operation trigger status bits and their meaning

Bit no.	Short description	Bit is set if
1	Sensor waiting for trigger	Sensor is waiting for a trigger event. When the trigger event occurs, the sensor changes into the measuring state.

10.3.8.4 Operation Sense Status Register

The `CONDition` register contains information whether a power sensor is being initialized. The `EVENT` register contains information whether an initialization was started or completed since the last query.

A power sensor is initialized if:

- Supply voltage is switched on (power-up).
- Sensor was connected.
- Reset was performed using:
 - `*RST`
 - `SYSTEM:PRESet`

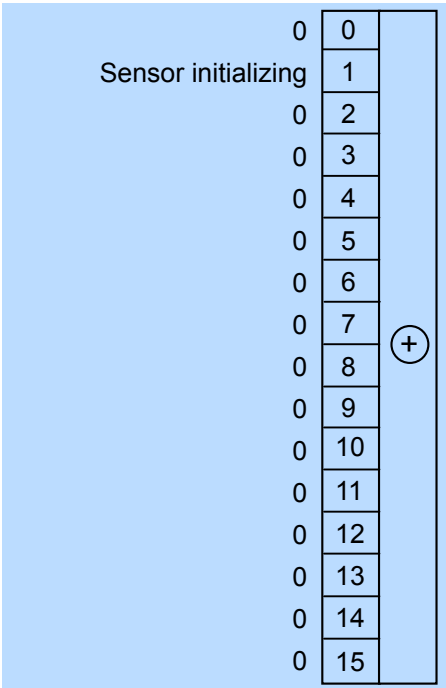


Figure 10-12: Operation sense status register

Querying the register:

- `STATUS:OPERation:SENSe:CONDition?`
- `STATUS:OPERation:SENSe[:SUMMARY][:EVENT]?`

Table 10-14: Used operation sense status bits and their meaning

Bit no.	Short description	Bit is set if
1	Sensor initializing	Sensor is being initialized.

10.3.8.5 Operation Lower Limit Fail Status Register

The CONDition registers contain information whether a measured value is below a configured lower limit. The EVENT registers contain information whether a measured value dropped below a limit value since the last query.

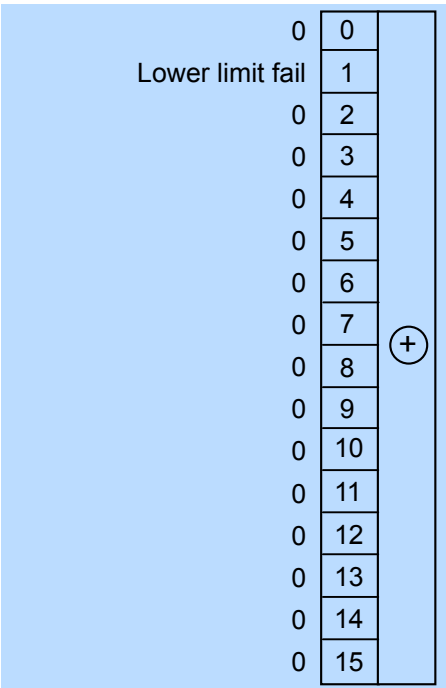


Figure 10-13: Operation lower limit fail status registers

Querying the register:

- `STATUS:OPERation:LLFail:CONDition?`
- `STATUS:OPERation:LLFail[:SUMMary][:EVENT]?`

Table 10-15: Used operation lower limit fail status bits and their meaning

Bit no.	Short description	Bit is set if
1	Lower limit fail	Measured value is below the lower limit value.

10.3.8.6 Operation Upper Limit Fail Status Register

The `CONDition` registers contain information whether a measured value currently exceeds a configured upper limit. The `EVENT` registers contain information whether a measured value exceeded an upper limit value since the last query.

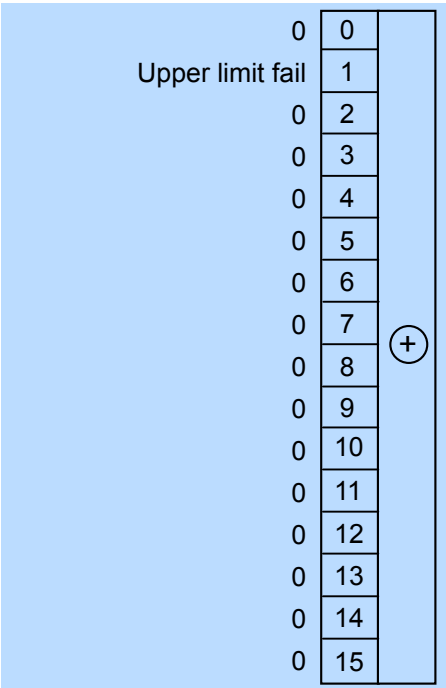


Figure 10-14: Operation upper limit fail status registers

Querying the register:

- `STATUS:OPERation:ULFail:CONDition?`
- `STATUS:OPERation:ULFail[:SUMMARY][:EVENT]?`

Table 10-16: Used operation upper limit fail status bits and their meaning

Bit no.	Short description	Bit is set if
1	Upper limit fail	Measured value exceeds the upper limit value.

11 Troubleshooting

- [Displaying Status Information](#)..... 153
- [Performing a Selftest](#)..... 153
- [Problems during a Firmware Update](#)..... 153
- [Contacting Customer Support](#)..... 154

11.1 Displaying Status Information

Status information is available in several ways.

Status LED of the R&S NRP18S-xx

The position of the status LED is indicated in [Chapter 4, "Power Sensor Tour"](#), on page 16.

The meaning of the different colors and blinking frequencies is explained in [Chapter 4.4, "Status LED"](#), on page 17.

11.2 Performing a Selftest

The selftest gives you detailed information that you can use for troubleshooting.



Do not apply a signal to the sensor while the selftest is running. If the selftest is carried out with a signal being present, error messages can erroneously be output for the test steps *Offset Voltages* and/or *Noise Voltages*.

Using remote control

- For a quick check, send `TEST:SENSor?`.

For each test step, PASS or FAIL is listed.

11.3 Problems during a Firmware Update

The firmware update is described in [Chapter 6, "Firmware Update"](#), on page 35.

Solutions for potential problems that can occur when using the Firmware Update for NRP Family, see ["Troubleshooting"](#) on page 37.

Firmware update was interrupted

If, for example, a power cut happened during the firmware update, problems can occur.

1. Perform the firmware update again. Sometimes, a further update fixes the problems.
2. If the power sensor is not accessible any more, contact the service.

Firmware update was aborted

If there is not enough free storage space, the firmware update is aborted. An error message is displayed, and the status LED of the power sensor starts flashing red.

1. Perform a sanitization procedure, as described in the instrument security procedures. This document is available on the product page, see [Chapter 1.1, "Documentation Overview"](#), on page 5, or the CD-ROM delivered with the power sensor.
2. Perform the firmware update again.

11.4 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 11-1: QR code to the Rohde & Schwarz support page

List of Commands

[SENSe<Sensor>:][POWer:][AVG:]APERture.....	70
[SENSe<Sensor>:][POWer:][AVG:]BUFFer:CLEar.....	71
[SENSe<Sensor>:][POWer:][AVG:]BUFFer:COUNT?.....	71
[SENSe<Sensor>:][POWer:][AVG:]BUFFer:DATA?.....	71
[SENSe<Sensor>:][POWer:][AVG:]BUFFer:SIZE.....	71
[SENSe<Sensor>:][POWer:][AVG:]BUFFer:STATe.....	71
[SENSe<Sensor>:][POWer:][AVG:]FAST.....	72
[SENSe<Sensor>:][POWer:][AVG:]SMOothing:STATe.....	72
[SENSe<Sensor>:][POWer:]BURSt:DTOLerance.....	73
[SENSe<Sensor>:][POWer:]BURSt:LENGth?.....	74
[SENSe<Sensor>:][POWer:]TSLot[:AVG]:COUNT.....	75
[SENSe<Sensor>:][POWer:]TSLot[:AVG]:WIDTh.....	75
[SENSe<Sensor>:][POWer:]TSLot[:AVG][:EXCLude]:MID:OFFSet[:TIME].....	75
[SENSe<Sensor>:][POWer:]TSLot[:AVG][:EXCLude]:MID:TIME.....	75
[SENSe<Sensor>:][POWer:]TSLot[:AVG][:EXCLude]:MID[:STATe].....	76
[SENSe<Sensor>:]AUXiliary.....	68
[SENSe<Sensor>:]AVERAge:COUNT.....	81
[SENSe<Sensor>:]AVERAge:COUNT:AUTO.....	82
[SENSe<Sensor>:]AVERAge:COUNT:AUTO:MTIME.....	82
[SENSe<Sensor>:]AVERAge:COUNT:AUTO:NSRatio.....	82
[SENSe<Sensor>:]AVERAge:COUNT:AUTO:RESolution.....	83
[SENSe<Sensor>:]AVERAge:COUNT:AUTO:SLOT.....	83
[SENSe<Sensor>:]AVERAge:COUNT:AUTO:TYPE.....	84
[SENSe<Sensor>:]AVERAge:RESet.....	83
[SENSe<Sensor>:]AVERAge:TCONtrol.....	84
[SENSe<Sensor>:]AVERAge[:STATe].....	84
[SENSe<Sensor>:]CORRection:DCYCLe.....	86
[SENSe<Sensor>:]CORRection:DCYCLe:STATe.....	87
[SENSe<Sensor>:]CORRection:OFFSet.....	87
[SENSe<Sensor>:]CORRection:OFFSet:STATe.....	87
[SENSe<Sensor>:]CORRection:SPDevice:LIST?.....	89
[SENSe<Sensor>:]CORRection:SPDevice:SElect.....	89
[SENSe<Sensor>:]CORRection:SPDevice:STATe.....	89
[SENSe<Sensor>:]FREQuency.....	85
[SENSe<Sensor>:]FUNction.....	66
[SENSe<Sensor>:]RANGe.....	48
[SENSe<Sensor>:]RANGe:AUTO.....	48
[SENSe<Sensor>:]RANGe:CLEVel.....	48
[SENSe<Sensor>:]ROSCillator:SOURce.....	47
[SENSe<Sensor>:]SGAMma:CORRection:STATe.....	91
[SENSe<Sensor>:]SGAMma:MAGNitude.....	91
[SENSe<Sensor>:]SGAMma:PHASe.....	91
[SENSe<Sensor>:]TIMing:EXCLude:STARt.....	86
[SENSe<Sensor>:]TIMing:EXCLude:STOP.....	86
[SENSe<Sensor>:]TRACe:AVERAge:COUNT.....	76
[SENSe<Sensor>:]TRACe:AVERAge:TCONtrol.....	76
[SENSe<Sensor>:]TRACe:AVERAge[:STATe].....	77

[SENSe<Sensor>:]TRACe:DATA?	77
[SENSe<Sensor>:]TRACe:MPWidth?	80
[SENSe<Sensor>:]TRACe:OFFSet:TIME	80
[SENSe<Sensor>:]TRACe:POINts	80
[SENSe<Sensor>:]TRACe:REALtime	80
[SENSe<Sensor>:]TRACe:TIME	81
*CLS	43
*ESE	44
*ESR?	44
*IDN?	44
*IST?	44
*OPC	44
*OPT?	45
*PRE	45
*RCL	45
*RST	45
*SAV	45
*SRE	46
*STB?	46
*TRG	46
*TST?	46
*WAI	47
ABORt	49
CALCulate:FEED	67
CALibration:DATA	102
CALibration:DATA:LENGth?	102
CALibration:USER:DATA	102
CALibration:USER:DATA:LENGth?	102
CALibration<Channel>:ZERO:AUTO	103
FETCh<Sensor>:ARRay[:POWer][:AVG]?	67
FETCh<Sensor>[:SCALar][:POWer]:BURSt?	67
FETCh<Sensor>[:SCALar][:POWer]:TSLot?	67
FETCh<Sensor>[:SCALar][:POWer][:AVG]?	67
FORMat:BORDer	64
FORMat:SREGister	65
FORMat[:DATA]	65
INITiate:ALL	50
INITiate:CONTInuous	50
INITiate[:IMMediate]	50
STATus:DEVice:CONDition?	117
STATus:DEVice:ENABLe	118
STATus:DEVice:NTRansition	118
STATus:DEVice:PTRansition	118
STATus:DEVice[:EVENT]?	117
STATus:OPERation:CALibrating:CONDition?	117
STATus:OPERation:CALibrating:ENABLe	118
STATus:OPERation:CALibrating:NTRansition	118
STATus:OPERation:CALibrating:PTRansition	118
STATus:OPERation:CALibrating[:SUMMary][:EVENT]?	117
STATus:OPERation:CONDition?	117

STATus:OPERation:ENABLE.....	118
STATus:OPERation:LLFail:CONDition?.....	117
STATus:OPERation:LLFail:ENABLE.....	118
STATus:OPERation:LLFail:NTRansition.....	118
STATus:OPERation:LLFail:PTRansition.....	119
STATus:OPERation:LLFail[:SUMMARY][:EVENT]?.....	117
STATus:OPERation:MEASuring:CONDition?.....	117
STATus:OPERation:MEASuring:ENABLE.....	118
STATus:OPERation:MEASuring:NTRansition.....	118
STATus:OPERation:MEASuring:PTRansition.....	119
STATus:OPERation:MEASuring[:SUMMARY][:EVENT]?.....	117
STATus:OPERation:NTRansition.....	118
STATus:OPERation:PTRansition.....	119
STATus:OPERation:SENSe:CONDition?.....	117
STATus:OPERation:SENSe:ENABLE.....	118
STATus:OPERation:SENSe:NTRansition.....	118
STATus:OPERation:SENSe:PTRansition.....	119
STATus:OPERation:SENSe[:SUMMARY][:EVENT]?.....	117
STATus:OPERation:TRIGger:CONDition?.....	117
STATus:OPERation:TRIGger:ENABLE.....	118
STATus:OPERation:TRIGger:NTRansition.....	118
STATus:OPERation:TRIGger:PTRansition.....	119
STATus:OPERation:TRIGger[:SUMMARY][:EVENT]?.....	117
STATus:OPERation:ULFail:CONDition?.....	117
STATus:OPERation:ULFail:ENABLE.....	118
STATus:OPERation:ULFail:NTRansition.....	118
STATus:OPERation:ULFail:PTRansition.....	119
STATus:OPERation:ULFail[:SUMMARY][:EVENT]?.....	117
STATus:OPERation[:EVENT]?.....	117
STATus:PRESet.....	116
STATus:QUEStionable:CALibration:CONDition?.....	117
STATus:QUEStionable:CALibration:ENABLE.....	118
STATus:QUEStionable:CALibration:NTRansition.....	118
STATus:QUEStionable:CALibration:PTRansition.....	119
STATus:QUEStionable:CALibration[:SUMMARY][:EVENT]?.....	117
STATus:QUEStionable:CONDition?.....	117
STATus:QUEStionable:ENABLE.....	118
STATus:QUEStionable:NTRansition.....	118
STATus:QUEStionable:POWer:CONDition?.....	117
STATus:QUEStionable:POWer:ENABLE.....	118
STATus:QUEStionable:POWer:NTRansition.....	118
STATus:QUEStionable:POWer:PTRansition.....	119
STATus:QUEStionable:POWer[:SUMMARY][:EVENT]?.....	117
STATus:QUEStionable:PTRansition.....	119
STATus:QUEStionable:WINDow:CONDition?.....	117
STATus:QUEStionable:WINDow:ENABLE.....	118
STATus:QUEStionable:WINDow:NTRansition.....	118
STATus:QUEStionable:WINDow:PTRansition.....	119
STATus:QUEStionable:WINDow[:SUMMARY][:EVENT]?.....	117
STATus:QUEStionable[:EVENT]?.....	117

STaTus:QUEue[:NEXT]?	117
SYSTem:COMMunicate:NETWork:IPAdDress	106
SYSTem:COMMunicate:NETWork:IPAdDress:GATeway	106
SYSTem:COMMunicate:NETWork:IPAdDress:INFO?	107
SYSTem:COMMunicate:NETWork:IPAdDress:MODE	107
SYSTem:COMMunicate:NETWork:IPAdDress:SUBNet:MASK	107
SYSTem:COMMunicate:NETWork:RESet	105
SYSTem:COMMunicate:NETWork:REStart	105
SYSTem:COMMunicate:NETWork:STaTus?	105
SYSTem:COMMunicate:NETWork[:COMMon]:DOMain	105
SYSTem:COMMunicate:NETWork[:COMMon]:HOSTname	106
SYSTem:DFPRint<Channel>?	107
SYSTem:ERRor:ALL?	108
SYSTem:ERRor:CODE:ALL?	108
SYSTem:ERRor:CODE[:NEXT]?	108
SYSTem:ERRor:COUNT?	108
SYSTem:ERRor[:NEXT]?	109
SYSTem:FWUPdate	109
SYSTem:FWUPdate:STaTus?	109
SYSTem:HELP:HEADers?	110
SYSTem:HELP:SYNTax:ALL?	110
SYSTem:HELP:SYNTax?	110
SYSTem:INFO?	110
SYSTem:INITialize	111
SYSTem:LANGuage	111
SYSTem:LED:COLor	112
SYSTem:LED:MODE	112
SYSTem:MINPower?	112
SYSTem:PARameters:DELta?	113
SYSTem:PARameters?	113
SYSTem:PRESet	113
SYSTem:REBoot	113
SYSTem:REStart	113
SYSTem:RUTime	113
SYSTem:SERRor:LIST:ALL?	114
SYSTem:SERRor:LIST[:NEXT]?	114
SYSTem:SERRor?	114
SYSTem:SUTime	115
SYSTem:TLEvels?	115
SYSTem:TRANsaction:BEgin	115
SYSTem:TRANsaction:END	115
SYSTem:VERsion?	116
SYSTem[:SENSor]:NAME	115
TEST:SENSor?	103
TRIGger:ATRigger:DELay	59
TRIGger:ATRigger:EXECuted?	59
TRIGger:ATRigger[:STATe]	59
TRIGger:COUNT	59
TRIGger:DELay	60
TRIGger:DELay:AUTO	60

TRIGger:DTIME.....	60
TRIGger:EXTerナル<2...2>:IMPedance.....	60
TRIGger:HOLDoff.....	61
TRIGger:HYSTeresis.....	61
TRIGger:IMMediate.....	61
TRIGger:LEVel.....	61
TRIGger:LEVel:UNIT.....	62
TRIGger:MASTer:PORT.....	62
TRIGger:MASTer:STATe.....	62
TRIGger:SLOPe.....	63
TRIGger:SOURce.....	63
TRIGger:SYNC:PORT.....	63
TRIGger:SYNC:STATe.....	63
UNIT:POWer.....	64

Index

A

Android device	
R&S Power Viewer Mobile	29
Application cards	6
Application notes	6

B

Basic measurement parameters	
Remote control commands	81
Boolean parameters	133
Brochures	6

C

Case-sensitivity	
SCPI	130
Checking	9
Commands	
Command-line structure	134
Common commands	43
Syntax	130
Computer	
Connection types	12
Simple USB connection	12
USB sensor hub	13
CONDition	137
Connecting	
Base unit	15
RF	10
Supported Rohde & Schwarz instrument	15
To a computer	12
USB host	11
Connector	
Host interface	17
Overview	16
RF	16
Trigger I/O	17

D

Data sheets	6
Device status register	140
Diagnostics	
Selftest	153
Disconnecting	
Host interface	12
RF	11
Documentation overview	5
Download	
R&S NRP Toolkit	18
Dropout time	
Trigger	52
DUT	
Connecting	10
Disconnecting	10

E

Electromagnetic interference (EMI)	10
Electrostatic discharge (ESD)	10
ENABLE	138
Enable register	144

ESE	144
ESR	144
EVENT	138

F

Firmware update	20, 35
Potential risks	35
Problems	153
Remote control	37
Requirements	35
Firmware Update for NRP Family	35

G

Getting started	5
-----------------------	---

H

Hold-off time	
Trigger	52

I

Instrument security procedures	5
Interface	
Overview	16
Introduction	6

K

Keywords	
optional	42

L

LAN connection	
Configuration	20
LED	
Status	17
Low-level communication program	20

M

Measurement	
Control	49
Mnemonics	129
Mobile measurements	29

N

NTRansition	137
-------------------	-----

O

Open source acknowledgment (OSA)	6
Operating concepts	18
App for Android devices	29
R&S NRP Toolkit	18
R&S NRP2	31
R&S NRPV	21
R&S NRX	29
R&S Power Viewer	25
Operating conditions	9
Operation calibrating status register	146

Operation lower limit fail status register	150
Operation measuring status register	147
Operation sense status register	149
Operation status register	145
Operation trigger status register	148
Operation upper limit fail status register	151

P

Parallel poll enable register	139
Parameters	
Block data	133
Special numeric values	132
String	133
Text	133
PPE	139
Preparing for use	9
PTRansition	137

Q

Queries	135
Questionable calibration status register	143
Questionable power status register	142
Questionable status register	141

R

R&S NRP Toolkit	18
Communication program	20
Components for Windows-based systems	19
Firmware update	20
Installation under Windows	19
Network Configuration	20
S-parameter update	20
System requirements	18
Uncertainty calculator	20
Versions	18
Versions of R&S NRP software	20
R&S NRP-Z5	13
R&S NRP2	31
R&S NRPV	21
R&S NRX	29
R&S Power Viewer	25
R&S Power Viewer Mobile	29
Release notes	6
Remote control	
Common commands	43
Interfaces	128
Protocols	128
USB interface	128
Remote control commands	
Basic measurement parameters	81
RF connector	10, 16

S

S-parameter update	20
Safety information	9
Safety instructions	5
SCPI	
Parameters	131
Syntax	130
SCPI status register	
Structure	136
Security procedures	5
Selftest	153

Service request enable register	138
Sources	
Trigger	51
Special characters	
SCPI	43
SRE	138
Standard event status register	144
Status	
Information	153
Status byte register	138
Status LED	17
Status register	136
Device status register	140
Enable register	144
Hierarchy	135
IST flag	139
Operation calibrating status register	146
Operation lower limit fail status register	150
Operation measuring status register	147
Operation sense status register	149
Operation status register	145
Operation trigger status register	148
Operation upper limit fail status register	151
Parallel poll enable register	139
Questionable calibration status register	143
Questionable power status register	142
Questionable status register	141
Remote control	116
Service request enable register	138
Standard event status register	144
Status byte register	138
Status registers	
CONDition	137
ENABle	138
EVENT	138
NTRansition	137
PTRansition	137
Sum bit	138
STB	138
Suffixes	42, 130
Sum bit	138

T

Trigger	50
Dropout time	52
Event	51
Hold-off time	52
Sources	51
States	51
Trigger I/O connector	17
Troubleshooting	153
Firmware update	153
Selftest	153
Status information	153
Tutorials	5

U

Uncertainty calculator	20
Units	132
Unpacking	9
USB connection	12
USB interface	
Remote control	128
USB product ID	129
USB resource string	129

USB sensor hub	13
User manual	5

V

Versions of R&S NRP software	20
------------------------------------	----

W

White papers	6
--------------------	---



北京海洋兴业科技股份有限公司 (证券代码: 839145)

北京市西三旗东黄平路19号龙旗广场4号楼(E座)906室

电话: 010-62176775 62178811 62176785

企业QQ: 800057747 维修QQ: 508005118

企业官网: www.hyxyyq.com

邮编: 100096

传真: 010-62176619

邮箱: market@oitek.com.cn

购线网: www.gooxian.com



扫描二维码关注我们
查找微信公众号: 海洋仪器