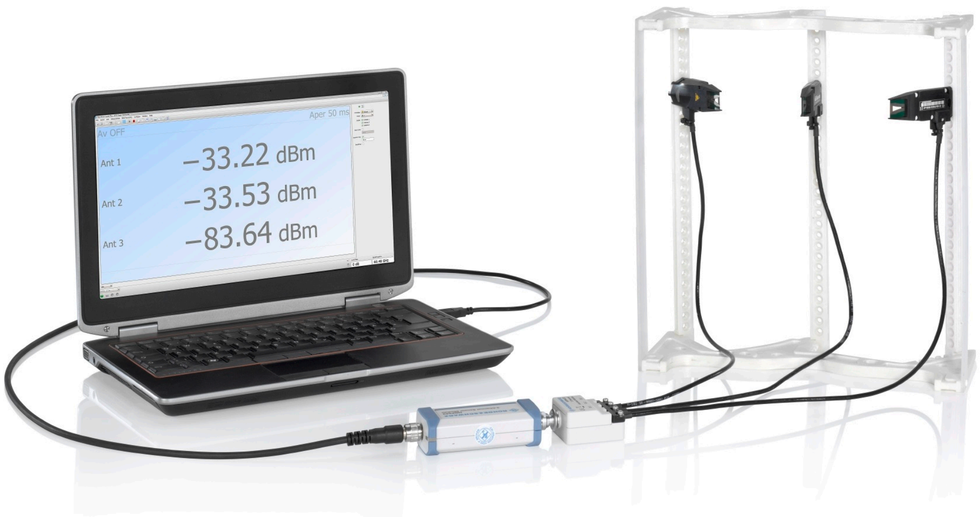
R&S®NRPM
OTA Power Measurement Solution
User Manual



1425866302

User Manual

Version 05

ROHDE&SCHWARZ

This document describes the R&S®NRPM OTA Power Measurement Solution, including the following components:

- R&S®NRPM3 Sensor Module, 1425.8563.02
- R&S®NRPM-A90 OTA Single Polarized Antenna Module, 1426.7760.02
- R&S®NRPM-A90D OTA Dual Polarized Antenna Module, 1426.7777.02
- R&S®NRPM-ZD3 Feedthrough Module, 1425.8786.02
- R&S®NRPM-Z3 Three Channel Interface Module, 1426.7602.02

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

1425.8663.02 | Version 05 | R&S®NRPM

The following abbreviations are used throughout this manual: R&S®NRPM is abbreviated as R&S NRPM, R&S®NRPM3 is abbreviated as R&S NRPM3, all R&S®NRPM-xxx are abbreviated as R&S NRPM-xxx, R&S®Power Viewer is abbreviated as R&S Power Viewer, R&S®Forum is abbreviated as R&S Forum, R&S®VISA is abbreviated as R&S VISA, R&S®TS7124 is abbreviated as R&S TS7124 and R&S®ATS1000 is abbreviated as R&S ATS1000. Microsoft®Windows®is abbreviated as MS Windows.

Safety Instructions

Instrucciones de seguridad

Sicherheitshinweise

Consignes de sécurité

⚠ WARNING

Risk of injury and instrument damage

The instrument must be used in an appropriate manner to prevent electric shock, fire, personal injury or instrument damage.

- Do not open the instrument casing.
- Read and observe the "Basic Safety Instructions" delivered as printed brochure with the instrument.
- Read and observe the safety instructions in the following sections. Note that the data sheet may specify additional operating conditions.
- Keep the "Basic Safety Instructions" and the product documentation in a safe place and pass them on to the subsequent users.

⚠ ADVERTENCIA

Riesgo de lesiones y daños en el instrumento

El instrumento se debe usar de manera adecuada para prevenir descargas eléctricas, incendios, lesiones o daños materiales.

- No abrir la carcasa del instrumento.
- Lea y cumpla las "Instrucciones de seguridad elementales" suministradas con el instrumento como folleto impreso.
- Lea y cumpla las instrucciones de seguridad incluidas en las siguientes secciones. Se debe tener en cuenta que las especificaciones técnicas pueden contener condiciones adicionales para su uso.
- Guarde bien las instrucciones de seguridad elementales, así como la documentación del producto, y entréguelas a usuarios posteriores.

WARNUNG

Gefahr von Verletzungen und Schäden am Gerät

Betreiben Sie das Gerät immer ordnungsgemäß, um elektrischen Schlag, Brand, Verletzungen von Personen oder Geräteschäden zu verhindern.

- Öffnen Sie das Gerätegehäuse nicht.
- Lesen und beachten Sie die "Grundlegenden Sicherheitshinweise", die als gedruckte Broschüre dem Gerät beiliegen.
- Lesen und beachten Sie die Sicherheitshinweise in den folgenden Abschnitten; möglicherweise enthält das Datenblatt weitere Hinweise zu speziellen Betriebsbedingungen.
- Bewahren Sie die "Grundlegenden Sicherheitshinweise" und die Produktdokumentation gut auf und geben Sie diese an weitere Benutzer des Produkts weiter.

AVERTISSEMENT

Risque de blessures et d'endommagement de l'appareil

L'appareil doit être utilisé conformément aux prescriptions afin d'éviter les électrocutions, incendies, dommages corporels et matériels.

- N'ouvrez pas le boîtier de l'appareil.
- Lisez et respectez les "consignes de sécurité fondamentales" fournies avec l'appareil sous forme de brochure imprimée.
- Lisez et respectez les instructions de sécurité dans les sections suivantes. Il ne faut pas oublier que la fiche technique peut indiquer des conditions d'exploitation supplémentaires.
- Gardez les consignes de sécurité fondamentales et la documentation produit dans un lieu sûr et transmettez ces documents aux autres utilisateurs.

Contents

1 Welcome to the R&S NRPM OTA Power Measurement Solution

The R&S NRPM OTA power measurement solution is designed to calibrate the transmit antenna output power and test the beamforming function over the air. Applications are in high frequency bands, used in modern high performance wireless system standards, e.g. 5G NR and IEEE802.11ay.

The R&S NRPM3 provides a USB high-speed host interface which constitutes both, the communication port and the power supply connection. This new sensor module and the specific antenna modules have been developed to provide measurements of radiated RF power.

Key features

The R&S NRPM OTA power measurement solution features:

- OTA power measurements with up to three antenna module readings per R&S NRPM sensor module
- Single and dual polarized antenna modules with integrated diode detector
- Very high sensitivity
- Fully calibrated system with specified system uncertainty
- Low-reflection antenna module
- Connection to a controller PC via USB
- Wide range of supported operating systems
- Easy operation with R&S NRP Toolkit and SCPI remote control.

About the R&S NRPM OTA power measurement solution

The power measurement solution consists of antenna modules, the three-channel sensor module, and interface modules for connecting the antenna modules to the sensor module. A PC measurement application controls the sensor module and the measurement.

The single or dual-polarized antenna modules (R&S NRPM-A90, R&S NRPM-A90D) are usually installed within an anechoic chamber, that can be:

- An RF test box, e.g. the R&S TS7124.
  The antenna modules are positioned inside the RF test box, the sensor module is outside. The sensor module is connected by the feedthrough module R&S NRPM-ZD3 that provides the interface for up to three antenna modules.
  Especially designed for use with the R&S TS7124, you can install up to six of the feedthrough modules. An antenna ring inside the R&S TS7124 allows you to flexibly position and align the antenna modules according to the application.
- An RF test chamber of larger dimensions, e.g. the R&S ATS1000 up to RF test rooms.
  The antenna modules and the sensor module are positioned inside the RF test chamber. The interface module R&S NRPM-Z3, directly connected to the sensor module provides the interface for up to three antenna modules.

With the single and dual polarized antenna modules (R&S NRPM-A90, R&S NRPM-A90D), you can calibrate the DUT (transmit antenna) output power and test the beamforming function.

The R&S NRPM measurement is controlled by an arbitrary user-definable measurement application. For a quick startup, Rohde & Schwarz provides the free interactive application R&S Power Viewer, which supports multiple sensor modules in the measurement modes "Continuous Average (reliable average power)" and "Trace (display of power versus time)".

The controller PC and the R&S NRPM3 communicate via a USB connection, using the standardized protocol USBTMC. For the controller PC, the drivers and APIs are provided for the major operating systems Linux, Mac OS X, MS Windows. It is required that you install a VISA driver on the host operating system, to use the I/O services provided by this standardized software interface library.

The sensor module also supports SCPI remote control, e.g. to automate a measurement with a scripting environment.

2 Preface

The R&S NRPM OTA power measurement solution is designated for the development, production and verification of electronic components and devices in industrial and laboratory environments. Always operate the system in an EM controlled environment, i.e. in an RF test box or RF test chamber. Use the OTA power measurement solution only for its designated purpose.

Observe the operating conditions and performance limits stated in the data sheet.

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

- In the "Basic Safety Instructions", safety issues are grouped according to subjects. For example, one subject is electrical safety. The "Basic Safety Instructions" are delivered with the R&S NRPM in different languages in print.
- Throughout the documentation, safety instructions are provided when you need to take care during setup or operation. Always read the safety instructions carefully. Make sure to comply fully with them. Do not take risks and do not underestimate the potential danger of small details such as a damaged power cable.

2.1 About this Manual

This user manual describes the R&S NRPM OTA power measurement solution.

The main focus in this manual is on the measurement setup, the results and the different solutions provided to configure and obtain them. The following topics are included:

- **Welcome to the R&S NRPM**
  Introduction to and getting familiar with the R&S NRPM and its specific application
- **Preparing for Use**
  Basic steps to be taken when setting up the R&S NRPM for the first time.
- **Measurement Setup**
  Test setup for the performance of power measurements with the R&S NRPM.
- **Performing Measurement Tasks**
  Programming examples for performing common measurement tasks.
- **Firmware Update**
  Description on updating the firmware.
- **Remote Control**
  Remote control interfaces and protocols, and all remote commands required to configure and perform measurements in a remote environment.
  Programming examples demonstrate the use of many commands and can usually be executed directly for test purposes.
- **Appendix**
  Extensive reference information on remote control.
- **List of Commands**
  Alphabetical list of all remote commands described in the manual.

- **Glossary**
  List of often used terms and abbreviations.
- **Index**

Details concerning the arrangement of the antenna modules in the test setup are not discussed in this manual.

2.2 Documentation Overview

This section provides an overview of the R&S NRPM user documentation. Unless specified otherwise, you find the documents on the product page at: www.rohde-schwarz.com/manual/nrpm.

2.2.1 User Manual

Introduces the R&S NRPM and describes how to set up and start working with the product. The manual includes general information, and the typical measurement application with programming examples. The sensor module specific functions, as well as an introduction to remote control and a complete description of the remote control commands are described.

A printed version is delivered with the R&S NRPM OTA Power Measurement Solution.

2.2.2 Basic Safety Instructions

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

2.2.3 Data Sheets and Brochures

The data sheet contains the technical specifications of the R&S NRPM OTA power measurement solution. It also lists the options and their order numbers as well as optional accessories.

The brochure provides an overview of the instrument and deals with the specific characteristics, see www.rohde-schwarz.com/brochure-datasheet/nrpm.

2.2.4 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 document provides verbatim license texts of the used open source software, see www.rohde-schwarz.com/firmware/nrpm

3 Preparing for Use

This section describes the basic steps to be taken when setting up the R&S NRPM OTA power measurement solution for the first time.

WARNING

Risk of injury due to disregarding safety information

Observe the information on appropriate operating conditions provided in the data sheet to prevent personal injury or damage to the instrument. Read and observe the basic safety instructions provided with the instrument, in addition to the safety instructions in the following sections.

In particular, do not open the instrument casing.

NOTICE

Risk of electrostatic discharge (ESD)

Electrostatic discharge can quickly and imperceptibly damage or destroy the electrostatic sensitive antenna modules.

To prevent ESD, protect the work area against electrostatic discharge. Use an anti-static wrist strap and cord and connect yourself to the ground, or use a conductive floor mat and heel strap combination.

For details, refer to the general safety instructions.

NOTICE

Risk of antenna module damage when touching the PCB of the antenna modules

Never touch the exposed top or bottom of the antenna module PCB (printed circuit board) during setting up. The PCB of the antenna module is sensitive. On contact, it can be bent and lead to measurement inaccuracies.

Always hold the antenna modules by the housing.

NOTICE

Risk of instrument damage during operation

An unsuitable operating site or test setup can damage the sensor module and connected devices. Make sure that the following operating conditions are met:

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

3.1 EMI Suppression

Electromagnetic interference (EMI) affects the measurement results.

To suppress generated electromagnetic interference:

- Use only the cables provided for the R&S NRPM OTA power measurement solution.
- Always terminate open cable ends.
- Note the EMC classification in the data sheet.

Since the antenna module receives signals in a wide band, also outside the specified frequency bands, perform the measurements always in a controlled EM environment. In particular, when working with the R&S NRPM-Z3 interface module.

3.2 Unpacking and Checking the Instrument

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

Packing material and transportation

Always make sure that sufficient mechanical and electrostatic protection is provided when transporting the R&S NRPM components.

3.3 R&S NRPM Tour

The following chapters introduce the main hardware components of the R&S NRPM OTA power measurement solution.

3.3.1 R&S NRPM3 Sensor Module

This chapter provides an overview of the available connectors and LED of the R&S NRPM3.

Figure 3-1: The R&S NRPM3 sensor module

1 = Antenna connector
2 = Host interface connector
3 = Trigger I/O connector
4 = Status LED

Antenna connector (1)

Multipole antenna connector for connecting the

- R&S NRPM3 to the interface module R&S NRPM-Z3.
- R&S NRPM3 to the filtered cable feedthrough module R&S NRPM-ZD3 with the interface cable R&S NRPM-ZKD3.

Host interface connector (2)

The host interface connector is used for establishing a connection between the sensor module and the USB host. Use either the R&S NRP-ZKU or the NRP-ZK6 cable for the connection.

The connector also contains the trigger1 connection for input or output of a trigger signal, and for working in trigger master-slave mode (optional), e.g. to synchronize several sensor modules.

For information on how to assign the signals to the ports, see Chapter 7.8, "Configuring the Trigger", on page 90.

Trigger I/O connector (3)

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

You can use the connector for input or output of a trigger signal, and for working in trigger master-slave mode (optional), see also "Host interface connector (2)" on page 13.

For information on how to assign the signals to the ports, see Chapter 7.8, "Configuring the Trigger", on page 90.

Status LED (4)

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

Color	State
	Idle state. The sensor module performs no measurement and is ready for use.
blinking	Firmware update is in progress
	Wait for trigger state
	Measuring state
	Zeroing is in progress
slow blinking	A static error. You can query the type of the error with `SYSTem:SERRor?`
fast blinking	A critical static error. You can query the type of the error with `SYSTem:SERRor?` **Hint:** If a critical error occurs after a firmware update, the update was not successful. Perform the firmware update again.

3.3.2 R&S NRPM-A90 and R&S NRPM-A90D Antenna Modules

This section introduces the antenna modules of the R&S NRPM OTA power measurement solution.

Both, the R&S NRPM-A90 and the R&S NRPM-A90D antenna modules require an R&S NRPM3 sensor module.

R&S NRPM-A90

Single polarized antenna module with integrated diode detector

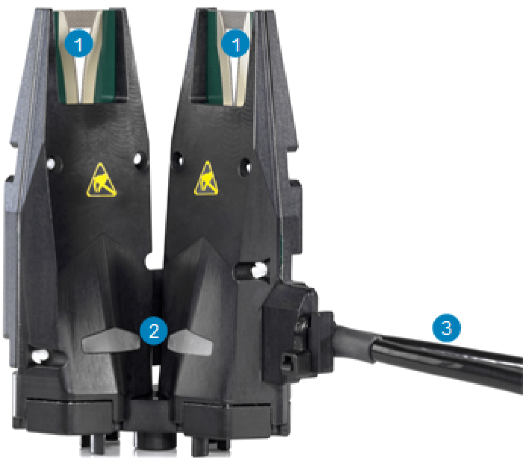
1 = PCB (printed circuit board)

2 = Signaling LED

3 = Interface cable

R&S NRPM-A90D

Dual polarized antenna module with integrated diode detectors



1 = PCBs (printed circuit boards)

2 = Signaling LEDs

3 = Interface cables

To distinguish the feeds, the carrier of the R&S NRPM-A90D antenna module is labeled on the back with the corresponding letters "1" and "2".

An R&S NRPM-A90D module occupies two channels on the R&S NRPM3.

R&S NRPM OTA power measurements with the antenna modules R&S NRPM-A90 and R&S NRPM-A90D require at least FW version `NRPM3N_18.05.08.03.rsu` on the sensor module. Earlier FW versions support only the predecessor antenna modules R&S NRPM-A66.

Printed circuit boards of the antenna modules (1)

NOTICE

Risk of antenna module damage when touching the PCB of the antenna modules (1)

Never touch the exposed top or bottom of the antenna module PCB (printed circuit board) during setting up. The PCB of the antenna module is sensitive. On contact, it can be bent and lead to measurement inaccuracies.

Always hold the antenna modules by the housing.

Signaling LED (2)

LED for signaling purposes.

You can use the LED for mapping between the antenna module location and the sensor module channel, or for own signaling purposes.

To turn on the LED, use the SCPI command `SYSTem:LED:CHANnel<Channel>:COLor` on page 58. The LED color is blue.

Antenna module cable (3)

Cable firmly connected to the antenna modules for connection to the R&S NRPM-ZD3 feedthrough or the R&S NRPM-Z3 interface modules.

The contact durability of these connectors is limited, therefore note the plug-in cycles specified in the data sheet.

3.3.3 R&S NRPM-ZD3 Feedthrough Module

This section introduces the filtered cable feedthrough module for installation, e.g. in the RF test box R&S TS7124.

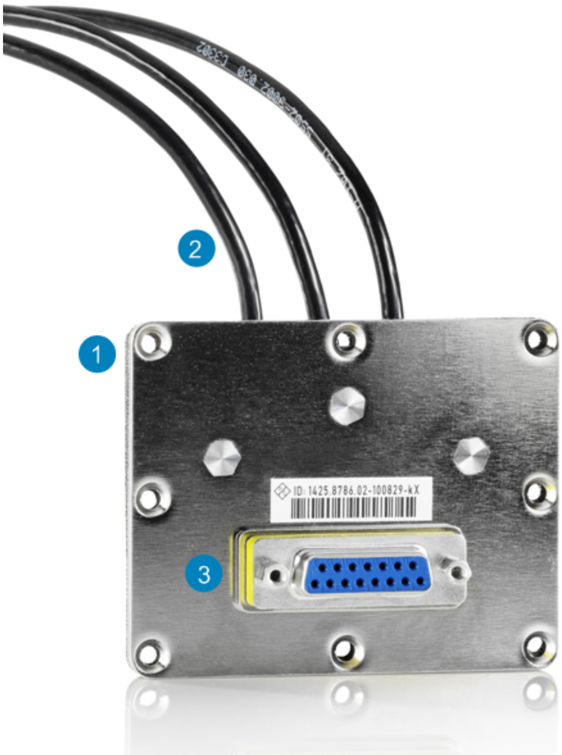


Figure 3-2: The R&S NRPM-ZD3 feedthrough module

1 = Cable feedthrough module
2 = Antenna module cable connectors
3 = Sensor module cable connector

Filtered cable feedthrough module (1)

R&S NRPM-ZD3 filtered cable feedthrough for combining three antenna module cables to one sensor module cable R&S NRPM-ZKD3.

Antenna module cable connectors (2)

Micro miniature connectors (10 pin) for connecting up to three antenna module cables.

Sensor module cable connector (3)

SUB-D connector (15 pin) for connecting to the sensor modules.

3.3.4 R&S NRPM-Z3 Interface Module

This section introduces the interface module for connecting, e.g. three antenna module cables for OTA power measurements in an EM-controlled environment, like the RF test chamber R&S ATS1000.

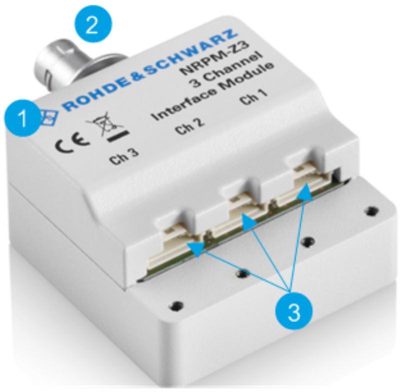


Figure 3-3: The R&S NRPM-Z3 interface module

1 = Interface module
2 = Connector to sensor module
3 = Antenna module cable connectors

Interface module (1)

R&S NRPM-Z3 three channel interface module to pass up to three antenna module cables to one R&S NRPM3 sensor module.

Connector to sensor module (2)

16-pin push-pull connector for connecting to the sensor module.

Antenna module cable connectors (3)

Micro miniature connectors (10 pin) for connecting the antenna module cables.

3.3.5 R&S NRPM-Z3 Connected to the R&S NRPM3

Figure 3-4: R&S NRPM-Z3 connected to R&S NRPM3

1 = R&S NRP-ZKU
2 = R&S NRPM3
3 = R&S NRPM-Z3
4 = Strain relieve
5 = Antenna module cables

Strain relieve (4)

Strain relieve for the antenna module cable connections.

Antenna module cables (5)

Cables firmly connected to the antenna modules for connection to the R&S NRPM-Z3 interface module or the R&S NRPM-ZD3 feedthrough interface modules.

The contact durability of these connectors is limited, therefore note the plug-in cycles specified in the data sheet.

3.4 Hardware and Software Requirements

3.4.1 Hardware

The basic measurement equipment for an OTA power measurement with the R&S NRPM consists of:

- Three channel sensor module R&S NRPM3.
- antenna module: one at a minimum, and three at a maximum per sensor module
- Antenna modules:
  – R&S NRPM-A90: one at a minimum, and three at a maximum per sensor module.
  – R&S NRPM-A90D: occupies two channels on the sensor module. You can split the antenna module cables of one R&S NRPM-A90D antenna module on two interface or feedthrough modules.
- One of the following:
  – Filtered cable feedthrough module R&S NRPM-ZD3 (RF test box) and the interface cable R&S NRPM-ZKD3.
  – Three channel interface module R&S NRPM-Z3 (RF test chamber).
  Tool for fixing the strain relieve: torx screwdriver TX8.
- Desktop PC or laptop, with:
  – Linux, Mac OS X, or MS Windows operating system.
  – The required software, see Mandatory Software.
  – USB interface.
- USB cable R&S NRP-ZKU, or NRP-ZK6.

See Chapter 4, "Setting Up a Measurement", on page 27 for examples on how to set up an R&S NRPM OTA power measurement.

3.4.2 Mandatory Software

The sensor modules are smart sensor modules that can be directly connected to a controlling PC.

To communicate with the sensor module, you can use the VISA I/O standard.

It is required to install:

- A driver software which supports I/O communication functions. It is required that you install a VISA software library, see Chapter 3.5.1, "VISA Driver Installation", on page 22. You can use any VISA library of your choice, however we recommend the use of the R&S VISA
  R&S VISA allows fast communication with the sensor module via USBTMC. It includes a trace tool for communication analysis, a testing tool for connection check, and a configuration tool for the definition of resources.
- R&S NRP Toolkit provides valuable utility programs, e.g. the firmware update program PureFW, see Chapter 3.5.2, "R&S NRP Toolkit Installation", on page 22.

The following table points to helpful links where you can find the corresponding drivers.

Table 3-1: Links to VISA driver software

OS	VISA driver
MS Windows	www.rohde-schwarz.com/rsvisa
Linux	www.rohde-schwarz.com/rsvisa Also ask our customer support if you do not find a suitable VISA library for your Linux version at customersupport@rohde-schwarz.com
Mac OS X	www.rohde-schwarz.com/rsvisa www.ni.com/visa

3.4.3 Optional Software

Optionally, you can use the following software applications:

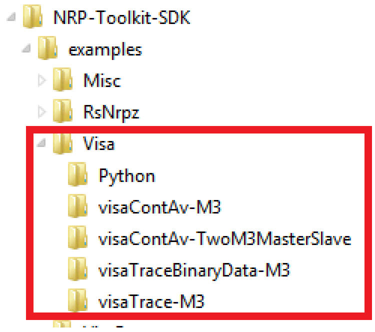
- R&S Power Viewer, the interactive power measurement software application from Rohde & Schwarz for measurements and evaluation of results with the sensor modules.
  As a prerequisite R&S Power Viewer requires a VISA driver, and the installation of the R&S NRP Toolkit, see Chapter 3.5.2, "R&S NRP Toolkit Installation", on page 22.
- A scripting tool for remote control of Rohde & Schwarz instruments. It is recommended that you use the R&S Forum, see Chapter 3.5.4, "R&S Forum Installation", on page 25.
  R&S Forum enables you to write, edit and run script sequences, and remote control during operation. Script files can range from simple command sequences (Win-

dows command syntax) to complex programs using the programming language Python.

- Programming examples for customer-specific applications.

  Rohde & Schwarz provides various programming examples containing:
  - The GUI application R&S Power Viewer.
  - Programming examples in C/C++ or Python source code, for VISA communication protocols.
  - Project and auxiliary files.

  The programming examples are included in the SDK (software development kit) of the R&S NRP Toolkit. You can select the SDK during installation, see "To install the R&S NRP Toolkit on MS Windows" on page 22. The folder labels are self-explanatory.



See Table 3-2, which shows where you can get the packages for the corresponding OS.

Table 3-2: Information to GUI application R&S Power Viewer and programming examples

OS	R&S Power Viewer	Programming examples
MS Windows	Available on the Internet, see "To install the R&S Power Viewer on an MS Windows:" on page 25.	User documentation CD-ROM (included in delivery). Available, part of the R&S NRP Toolkit, see Chapter 3.5.2, "R&S NRP Toolkit Installation", on page 22.
Linux	Available on request for various Linux distributions, e.g. CentOS, Debian, RedHat, SuSE, Ubuntu. See "To install the R&S Power Viewer on Linux" on page 25.	Packages available via the customer support: www.customersupport.rohde-schwarz.com
Mac OS X	Available, part of the R&S NRP Toolkit, Mac OS X package, see "To install the R&S NRP Toolkit on Mac OS X" on page 24.	

3.5 Installing the Software Application and Drivers

This section describes the steps to install the required software applications, tools and drivers. The procedures refer to the programs recommended by Rohde & Schwarz.

3.5.1 VISA Driver Installation

To install the R&S VISA standardized software library on MS Windows:

► Consult the Rohde & Schwarz http://www.rohde-schwarz.com/rsvisa web site, that provides the software and the necessary information.

If you are using VISA from another supplier, refer to the corresponding documentation on how to install the driver on your system. Usually, you start the received or downloaded installer `*.exe` or `*.msi` file.

To install a VISA driver on Linux or Mac OS X:

► Refer to the documentation of the VISA driver supplier for instructions on how to install the VISA driver.

3.5.2 R&S NRP Toolkit Installation

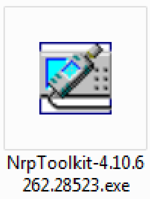
The R&S NRP Toolkit is a software-package that provides various utility programs for the whole Rohde & Schwarz NRP power sensor family, including the R&S NRPM3.

To install the R&S NRP Toolkit on MS Windows

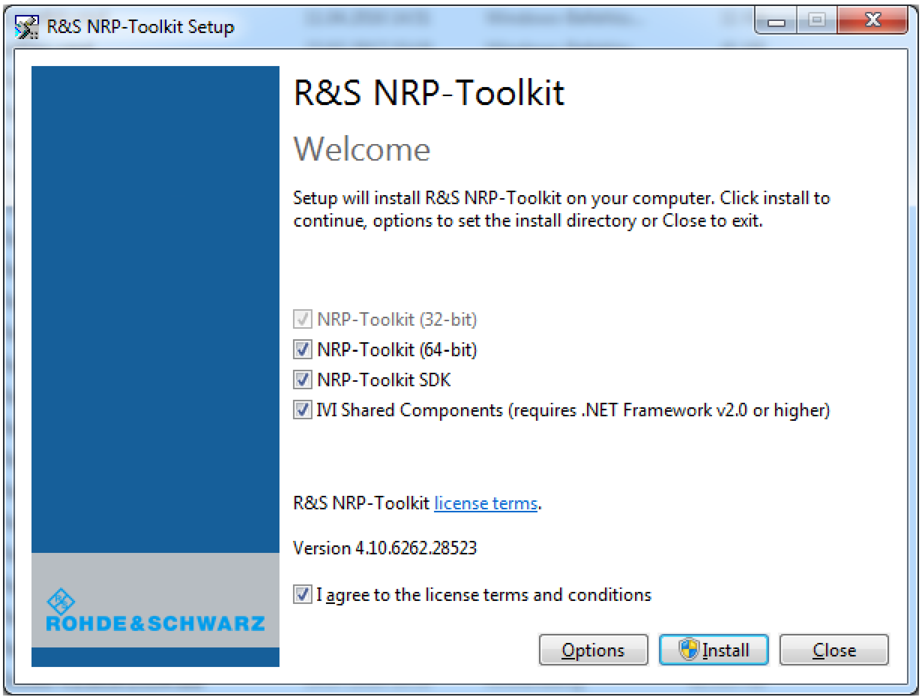
Rohde & Schwarz provides an R&S NRP Toolkit executable file, which installs the necessary low-level and high-level drivers and various tools on an MS Windows OS. Supported are MS Windows 7, MS Windows 8, MS Windows 8.1, and MS Windows 10 in both, the 32-bit and 64-bit architectures. For certain devices, drivers supporting MS Windows CE5 and MS Windows CE7 are available on request.

1. Download the latest installer file from the https://www.rohde-schwarz.com/software/nrp_s_sn website. The filename is similar to `NRP-Toolkit-A.B.C.D.exe`, where `A, B, C and D` are placeholders for the version number, e.g. `NRP-Toolkit-4.10.6262.28523.exe`.

2. Double-click the icon to start the installer.



A dialog similar to the following appears.



3. In the main dialog:
   a) Select the R&S NRP Toolkit 64-bit option if your MS Windows PC has 64-bit architecture. The 32-bit version is always installed.
   b) To include the programming examples, select "NRP-Toolkit SDK". To develop own application programs with the R&S NRPM3, we recommend that you activate this option to get various programming samples installed onto your PC.
   c) Read the legal terms under "license terms"
   d) Confirm that you agree to the license terms and conditions.
4. Continue with "Install" to complete the installation process.

   When completed, you can find the included programming examples under `C:/ProgramData/Rohde-Schwarz/NRP-Toolkit-SDK/examples` if the "NRP-Toolkit SDK" option was activated before.

To install the R&S NRP Toolkit on Linux

R&S NRP Toolkit versions for Linux distributions are available on request.
Rohde & Schwarz provides installation packages for many popular distributions like Debian, CentOS, Ubuntu, OpenSuse, Raspbian, and others, in various versions. The packages constitute installable `*.deb` or `*.rpm` files which can be processed by the standard package tools (like `dpkg` or `rpm`) of the corresponding Linux distribution.

1. To obtain an R&S NRP Toolkit for your distribution, contact the Rohde & Schwarz customer support: www.customersupport.rohde-schwarz.com.

2. The corresponding packages come with a `README.txt` file which describes in detail the steps to install the Linux R&S NRP Toolkit.

   An R&S NRP Toolkit for Linux normally consists of several files (`*.deb` or `*.rpm` files). The associated files for a certain distribution all together make up the individual R&S NRP-Toolkit for Linux. The following examples show two random representatives. The filenames vary depending on the distributions and the versions.

Example: Files used on Debian based systems

```
sudo dpkg  -i  nrpzmodule-dkms_4.2.2_all.deb      # Kernel driver
sudo dpkg  -i  libnrp_4.11-1_amd64.deb            # Transport layer
sudo dpkg  -i  librsnrpz_3.0.0.0-1_amd64.deb      # API
sudo dpkg  -i  libnrp-dev_4.11-1_amd64.deb        # Dev. files
                                                    (Transport layer)
sudo dpkg  -i  librsnrpz-dev_3.0.0.0-1_amd64.deb  # Dev.files
                                                    (API headers etc.)
```

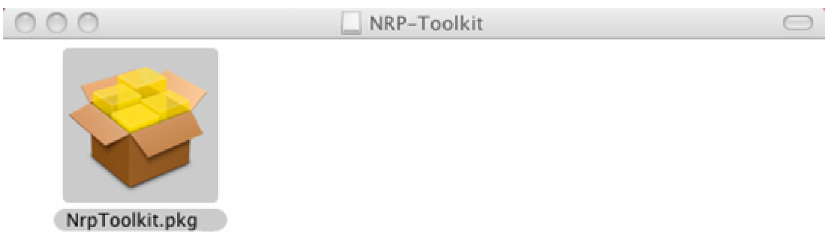
Example: Files used on RPM based systems

```
sudo rpm -i kmod-nrpzmodule-4.2.2-1.el7.centos.x86_64.rpm
                                                  # Kernel driver
sudo rpm -i libnrp-4.11-1.el7.centos.x86_64.rpm   # Transport layer
sudo rpm -i librsnrpz-3.0.0-1.el7.centos.x86_64.rpm
                                                  # API
sudo rpm -i libnrp-devel-4.11-1.el7.centos.x86_64.rpm
                                                  # Dev. files
                                                   (Transport layer)
sudo rpm -i librsnrpz-devel-3.0.0-1.el7.centos.x86_64.rpm
                                                  # Dev.files (API hdrs)
```

To install the R&S NRP Toolkit on Mac OS X

Rohde & Schwarz provides a Mac OS X NRP-Toolkit installer. The Mac OS X version differs from the Linux and Windows NRP-Toolkit in the way that it includes a version of the R&S Power Viewer application. On the other operating systems, the R&S Power Viewer is a separately installable package. This application provides OTA functions for easy measurements with the R&S NRPM3 sensor modules.

1. To obtain a Mac OS X NRP-Toolkit installer, contact the Rohde & Schwarz customer support: www.customersupport.rohde-schwarz.com.
2. Double-click the icon to start the installer.



3. Follow the instructions of the installer to complete the installation.

3.5.3 R&S Power Viewer Installation

The R&S Power Viewer is a software application that simplifies many power measurement tasks. It is part of the Mac OS X NRP-Toolkit installer, and is available as a *separately* installable package for Linux and MS Windows.

This program supports measuring the received signals of up to 12 antenna modules. It provides simultaneous OTA measurement of four sensor modules, each transmitting the readings of up to three antenna modules.

The following instructions describe how to install the R&S Power Viewer. For detailed information, see the user manual of the R&S Power Viewer, which is part of the installation.

To install the R&S Power Viewer on an MS Windows:

1. Download the software package, e.g. from https://www.rohde-schwarz.com/software/nrp_s_sn.
2. Start the installer program `PowerViewer_xxxxx_windows.msi` (xxxxx signifies a version number).
3. Follow the instructions of the installation wizard.

To install the R&S Power Viewer on Linux

The R&S Power Viewer for Linux comes as a self-extracting installation package (`*.run` file). Follow these steps to install the application in your Linux system:

1. Make sure that you have successfully installed a R&S NRP Toolkit for Linux suitable for your distribution.
2. Download the installer, e.g. from https://www.rohde-schwarz.com/software/nrp_s_sn, or contact the customer support to obtain the file.
3. Open a terminal window.
4. Enter `sudo ./powerviewer_9.0_linux_x86.run`
   Note: The actual number (here 9.0) can be different.

To install the R&S Power Viewer on Mac OS X

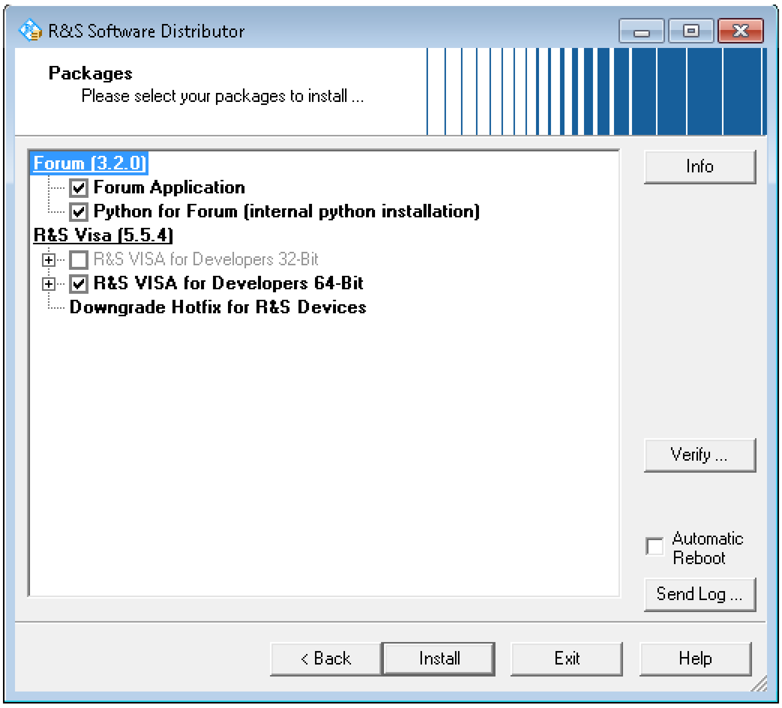
The R&S Power Viewer is part of the R&S NRP Toolkit.

► For installation, see "To install the R&S NRP Toolkit on Mac OS X" on page 24

3.5.4 R&S Forum Installation

R&S Forum is a scripting tool for remote control of Rohde & Schwarz instruments via the USB interface, using VISA. You can find more to this tool in the application note http://www.rohde-schwarz.com/appnote/1MA196.

R&S Forum must be installed on the PC, and requires a VISA library, e.g. the R&S VISA. If there is no VISA library available on the PC, you can include the R&S VISA during R&S Forum installation.



If you are using VISA from another supplier, refer to the corresponding documentation on how to install the driver on your system.

To install R&S Forum, perform the following steps:

1. For installation instructions, see release notes at the www.rohde-schwarz.com/forum website.
2. For download and installation, start the setup file (`*.exe`), also provided at this site.
3. Follow the instructions of the installer to complete the installation process.

4 Setting Up a Measurement

This section shows some test setup examples and brief instructions on how to connect the components of an OTA power measurement. You can also find references to the product page and the user manual of the RF test box R&S TS7124 or the RF test chamber R&S ATS1000.

The test setup examples show how to set up a measurement with an RF test box and within an RF test chamber. The main difference is that the sensor module is either inside, or outside the EM-controlled test environment:

- **RF test box**
  The sensor module is outside the RF test box, connected to the antenna modules via the feedthrough module (R&S NRPM-ZD3) and the interface cable (R&S NRPM-ZKD3).
- **RF test chamber**
  The sensor module is inside the RF test chamber, directly connected to the antenna modules via the interface module (R&S NRPM-Z3).

See also the application note 1GP118.

Always operate the R&S NRPM-A90 and R&S NRPM-A90D antenna modules in an environment that is protected against electromagnetic interference.

The described examples refer to measurements performed with R&S Power Viewer, which allows you to measure the readings of up to four R&S NRPM3 sensor modules (12 antenna module channels) at a time. This number corresponds to 12 single or 6 dual polarized antenna modules. Technically, more are possible, i.e. you can select any number of antenna modules for your measurement.

Single antenna module solution

The base configuration with one antenna module measures the power of the incident wave from the DUT to the antenna module, e.g. to calibrate the output power of your DUT.

Example:

Setup with one single polarized R&S NRPM-A90 antenna module installed in an RF test box (feedthrough module):

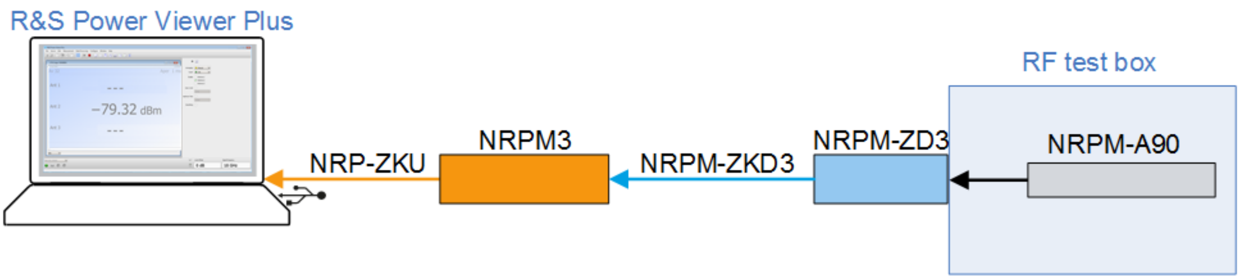


Figure 4-1: One R&S NRPM-A90 in an RF test box

Multiple spatially distributed antenna modules

With several antenna modules distributed in an area, you can test the beamforming function of a DUT. The more antenna modules are installed, the higher is the measurement accuracy during beamforming tests. In addition, you can derive matrices for swiveling the beam of the antenna module around two axes.

Example:

Setup with multiple single polarized R&S NRPM-A90 antenna modules in an RF test box (feedthrough modules):

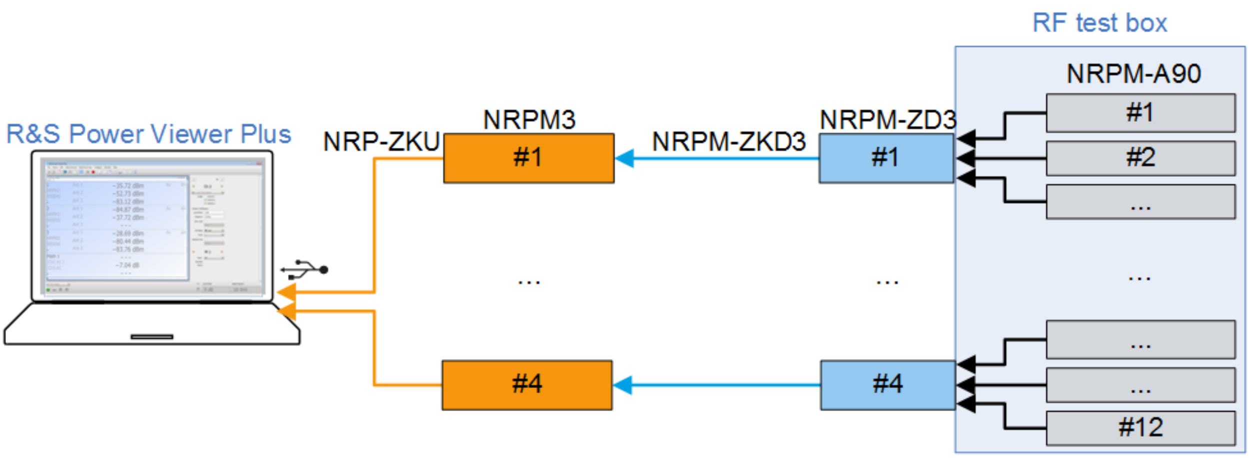


Figure 4-2: Multiple R&S NRPM-A90 in an RF test box

Example:

Setup with multiple single polarized R&S NRPM-A90 antenna modules in an RF test chamber (interface modules):

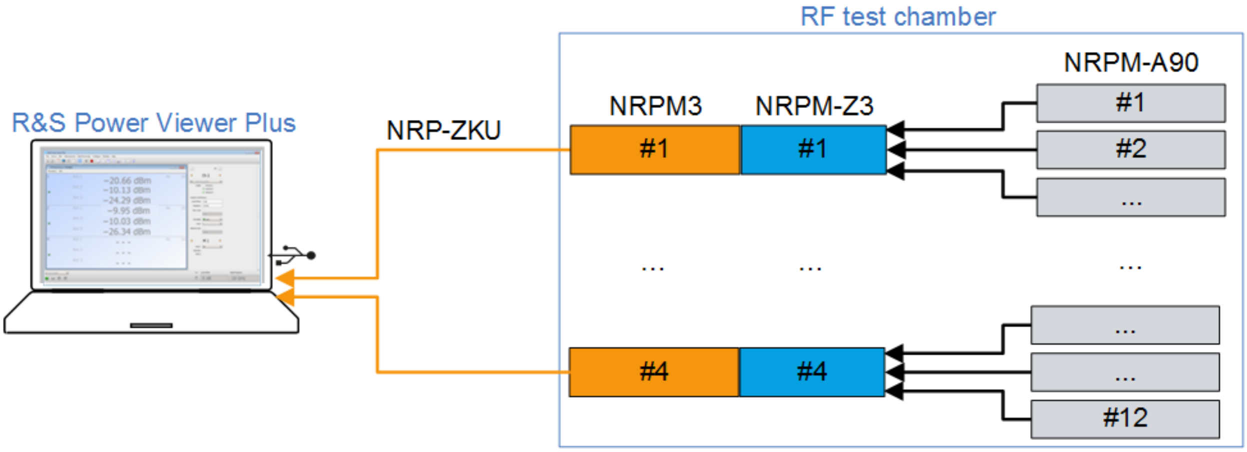


Figure 4-3: Multiple R&S NRPM-A90 in an RF test chamber

Example:

Setup with dual polarized R&S NRPM-A90D antenna modules in an RF test box with feedthrough modules:

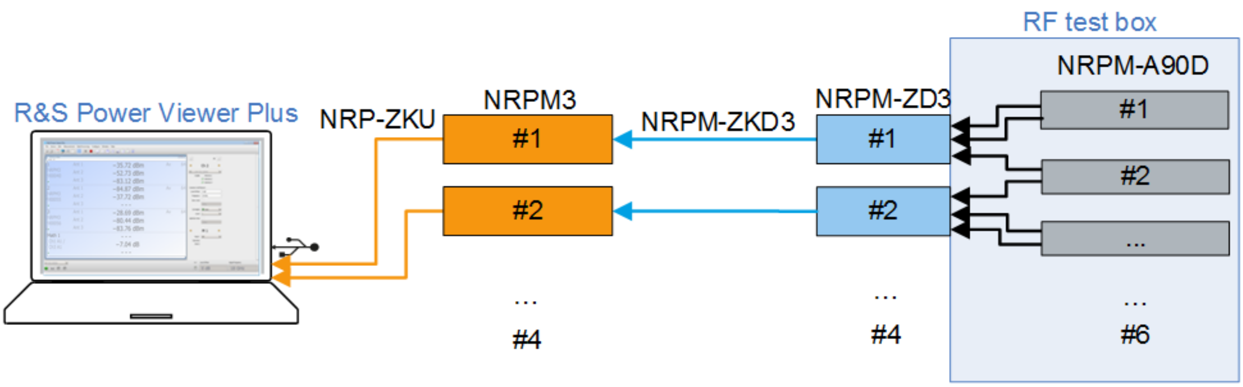


Figure 4-4: Setup with multiple R&S NRPM-A90D in an anechoic chamber

Note: Since one dual polarized antenna module allocates two channels of the sensor module, you can use a maximum of three antenna modules with two sensor modules. If your test setup requires more antenna modules, use additional sensor modules. In this test setup, you can measure the readings of up to six T-antenna modules with four sensor modules.

To set up a measurement with R&S NRPM-A90D dual polarized antenna modules in an RF test chamber, use the corresponding modules, as shown above.

4.1 R&S TS7124 RF Test Box

The R&S TS7124 RF test box enables reliable and reproducible measurements when a controlled EM test environment is needed.

As the R&S TS7124 is a stand-alone product of Rohde & Schwarz, this section does not describe the box and measurement setups in detail. If you perform an OTA measurement with the box, you find the necessary information at:

- www.rohde-schwarz.com/product/ts7124, providing an overview of the R&S TS7124 RF test box with the available models and the variety of antenna holders and feedthrough options.
- www.rohde-schwarz.com/manual/ts7124, providing the user manual, which describes the hardware, the options and the accessories, and how to install and configure the antenna modules.
- www.rohde-schwarz.com/brochure-datasheet/ts7124, providing the product brochure, which deals with specific characteristics, and contains the technical specifications and ordering information.

4.2 R&S ATS1000 RF Antenna Test System

The R&S ATS1000 RF test chamber enables reliable and reproducible measurements when a controlled EM test environment is needed.

As the R&S ATS1000 is a stand-alone product of Rohde & Schwarz, this section does not describe the chamber and measurement setups in detail. If you perform an OTA measurement within the chamber, you find the necessary information at:

- www.rohde-schwarz.com/product/ats1000, providing an overview of the R&S ATS1000 RF test chamber.
- www.rohde-schwarz.com/manual/ats1000, providing the user manual, which describes hardware and the standard equipment, the optional components and the accessories, and how to install and configure the antenna modules.
- www.rohde-schwarz.com/brochure-datasheet/ats1000, providing the product brochure, which deals with specific characteristics, and contains the technical specifications and ordering information.

4.3 Connecting the R&S NRPM3

To start up an R&S NRPM OTA power measurement, it is assumed that the following conditions are met:

- All required hardware is available and ready for use, see Chapter 3.4, "Hardware and Software Requirements", on page 19.
- The firmware version of the R&S NRPM3 is FW 18.05.08.03 or later, required for R&S NRPM OTA power measurements with the antenna modules R&S NRPM-A90 or R&S NRPM-A90D.

- The software and drivers are installed on the controller PC, see Chapter 3.5, "Installing the Software Application and Drivers", on page 21.
- The transmitting device (DUT) is installed and ready for operation.

The following steps describe the connections necessary for measurement setups in both, an RF test box or an RF test chamber.

The measurement is started with the R&S Power Viewer, exemplarily for a software application.

To set up the RF frontend in an RF test box

For connecting the RF frontend components, perform the following steps:

1. In the RF test box:
   a) Install a suitable antenna ring.
   b) Mount the antenna modules.
   Note:
   Never touch the exposed top or bottom of the antenna module PCB (printed circuit board) during setting up. The PCB of the antenna module is sensitive. On contact, it can be bent and lead to measurement inaccuracies.
   Always hold the antenna modules by the housing.
2. Connect each antenna module to the antenna module cable connectors of the feedthrough module R&S NRPM-ZD3 (2).
   Note:
   When you are working with the dual-polarized antenna modules R&S NRPM-A90D, notice the feeds of the vertical and horizontal antenna modules. You can find out the alignment with the identification numbers "1" (vertical) and "2" (horizontal) on the carrier of the antenna modules.
3. Outside the RF test box, connect the antenna connector of the R&S NRPM3 (1) to the feedthrough module R&S NRPM-ZD3 (3) with the interface cable (R&S NRPM-ZKD3).

To set up the RF frontend in an RF test chamber

For connecting the RF frontend components, perform the following steps:

1. In the RF test chamber, mount the antenna modules.
   Note:
   Never touch the exposed top or bottom of the antenna module PCB (printed circuit board) during setting up. The PCB of the antenna module is sensitive. On contact, it can be bent and lead to measurement inaccuracies.
   Always hold the antenna modules by the housing.
2. On the interface module, dismount the strain relieve NRPM-Z3 (4) using a torx screwdriver TX8.
3. Connect each antenna module cable to the antenna module cable connectors of the interface module NRPM-Z3 (3).
   Note:

When you are working with the dual-polarized antenna modules R&S NRPM-A90D, notice the feeds of the vertical and horizontal antenna modules. You can find out the alignment with the identification numbers "1" (vertical) and "2" (horizontal) on the carrier of the antenna modules.

4. Mount the strain relieve of the interface module.
5. Connect the interface module with the sensor module (R&S NRPM-Z3 connected to R&S NRPM3).
6. Connect the AC power.
   Establish the required control and measurement connections as described in the user documentation of the corresponding RF test chamber.

To set up the USB connection and start the measurement

For executing measurements with the R&S NRPM3, the module must be connected to a USB port of the host PC. An easy way for configuring OTA measurements and displaying results is given by the R&S Power Viewer software:

1. Connect the R&S NRPM3 host interface R&S NRPM3 (3) to the PC with a R&S NRP-ZKU USB cable.
2. Start the R&S Power Viewer application.

   The R&S Power Viewer identifies the R&S NRPM3 as a USBTMC device at start-up automatically. You can start with the measurement.

5 Performing Measurements

5.1 Sensor Module Readings

The antenna modules R&S NRPM-A90 and R&S NRPM-A90D have integrated diode detectors that convert the RF signal and transmit it directly to the sensor module.

You can measure the power of the incident electromagnetic wave towards the antenna module in various quantities:

- Equivalent isotropically received power $P_{ISO} = P_i$ in W or dBm (default):
  Equivalent detected power of an isotropic antenna with an ideal power detector at the phase center location of the antenna module assuming radiation only from boresight direction.
- Power at the internal RF detector in W or dBm: The measurement result without including antenna gain and frequency dependent calibration factors.
  $P_{Det}=(G_{RF,i}/k_{att,A,i}(f))\cdot P_{ISO}$
- Power density S in W/m^2, calculated as:
  $S=(4\pi/\lambda^2)\cdot P_{ISO}$
- Electric field strength E_{eff} in V/m, calculated as:
  $E_{eff}=\sqrt{(S\cdot Z_0)}$ with: Z_0=376.73 Ω.
- Magnetic field strength H_{eff} in A/m, calculated as:
  $H_{eff}=\sqrt{(S/Z_0)}$ with: Z_0=376.73 Ω.

► To convert the data, use the remote command:
`CALCulate:MATH[:EXPRession]` on page 73

5.2 Measurement Applications

For power measurement with the R&S NRPM3, you can either use the PC application R&S Power Viewer or an application that supports direct remote control of the sensor module.

The following sections introduce the power measurement with the R&S Power Viewer and provide a brief startup with R&S Forum. Section Chapter 8, "Programming Examples", on page 103 provides the corresponding programming examples for working in remote control mode. See the R&S Power Viewer manual for more information on how to use the various functions of the application.

The descriptions refer to the applications running on an MS Windows system. See Chapter 5.2.3, "References", on page 41 for short instructions on starting the R&S Power Viewer on Linux or MacOS X.

It is assumed, that the measurement is set up, and the required software and drivers are installed on the PC.

5.2.1 Using R&S Power Viewer

This section shows how to start the application and access the settings relevant for OTA measurements. For handling and using the tool in detail, see the "R&S Power Viewer Software Manual" that comes with the installation, see Chapter 3.5.3, "R&S Power Viewer Installation", on page 25.

The R&S Power Viewer provides three measurement modes for OTA measurements:

- "OTA Single"
  Controls one R&S NRPM3 sensor module, i.e. you can measure the continuous average power of up to 3 antenna module signals.
- "OTA Multi Sensor"
  Supports the measurements of up to four sensor modules, i.e. you can measure the continuous average power of up to 12 antenna module signals.
- "OTA Trace Measurement"
  Controls one sensor module, i.e. you can measure the power of up to 3 antenna module signals in trace mode.

R&S Power Viewer supports measuring the received signals of up to three antenna modules per sensor module.

Outlined are:

- To start the application.
- To configure an OTA single sensor ContAV measurement, for using the R&S NRPM3 with up to three antenna modules.
- To configure an OTA multi sensor ContAV measurement, for using of up to four sensor modules with up to three antenna modules each.
- To configure an OTA single sensor trace power measurement, for using the R&S NRPM3 with up to three antenna modules.

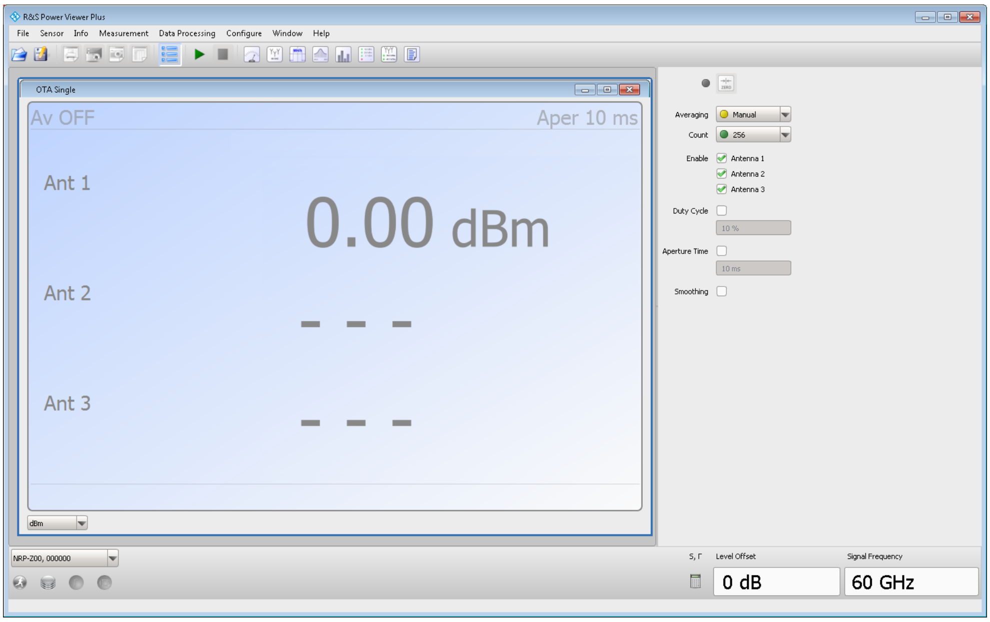
To start the application

► In the MS Windows start menu, select "Start" > "All Programs" > "R&S Power Viewer" > "Power Viewer".

The application starts. It provides buttons for OTA measurements in the toolbar.

To configure an OTA single sensor ContAV measurement

1. In the toolbar, select the "OTA Single" button to open the panel for the OTA measurements with one R&S NRPM3 sensor module.



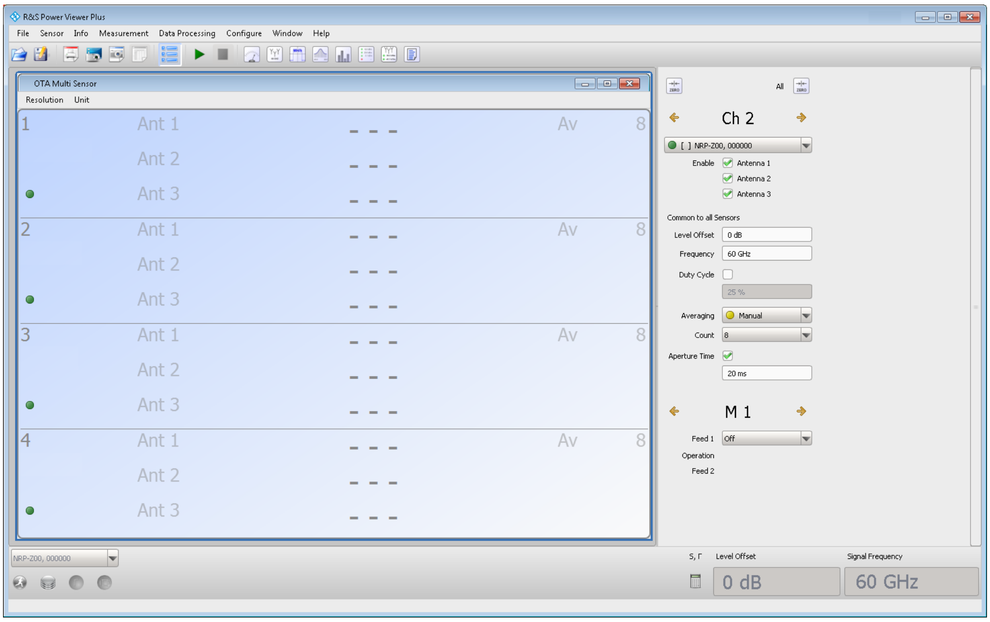
2. In the lower border toolbar, select the sensor module.
3. In the panel on the right, select the antenna modules for the measurement.
4. If necessary, set the parameters to configure the continuous average power measurement.
5. Set the frequency.
6. Start the measurement.

The measurement results window displays measured power in the sensor module channels.

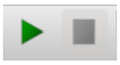
To configure an OTA multi sensor ContAV measurement

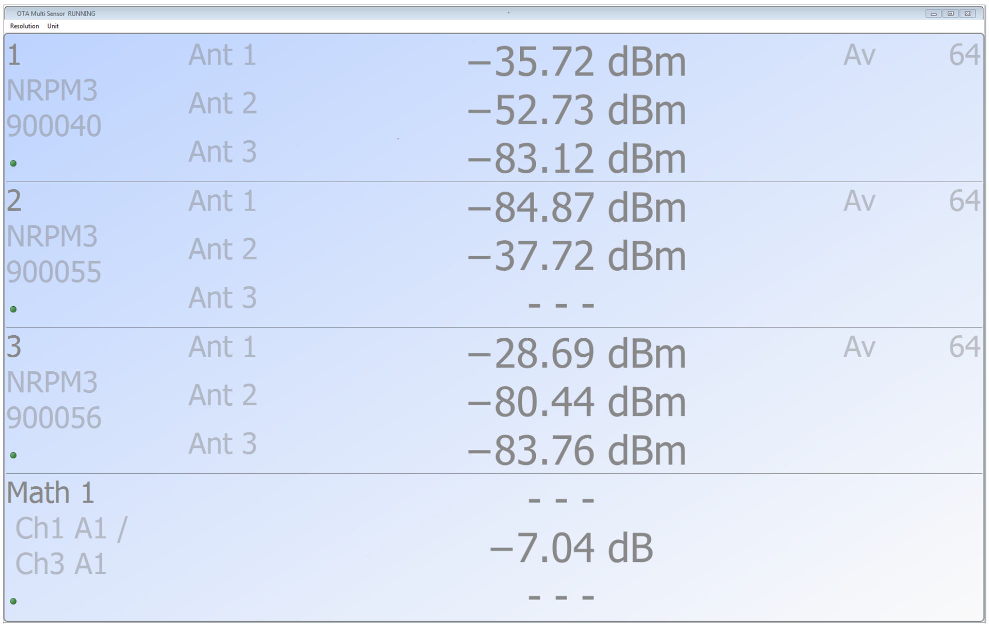
1. 

   In the toolbar, select the "OTA Multi Sensor" button to open the panel for the OTA measurements with several sensor modules.

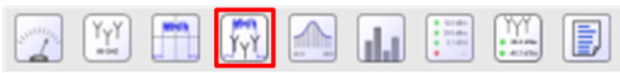
2. In the lower border toolbar, select the sensor module.
3. In the panel on the right, select the channel.
4. Enable the antenna modules of the selected channel.
5. If necessary, set the parameters provided for the measurement mode.
6. Repeat step 3 to step 5 for each channel.
7. Repeat step 2 to step 6 for each connected R&S NRPM3.
8. Set the frequency.
9. Start the measurement.



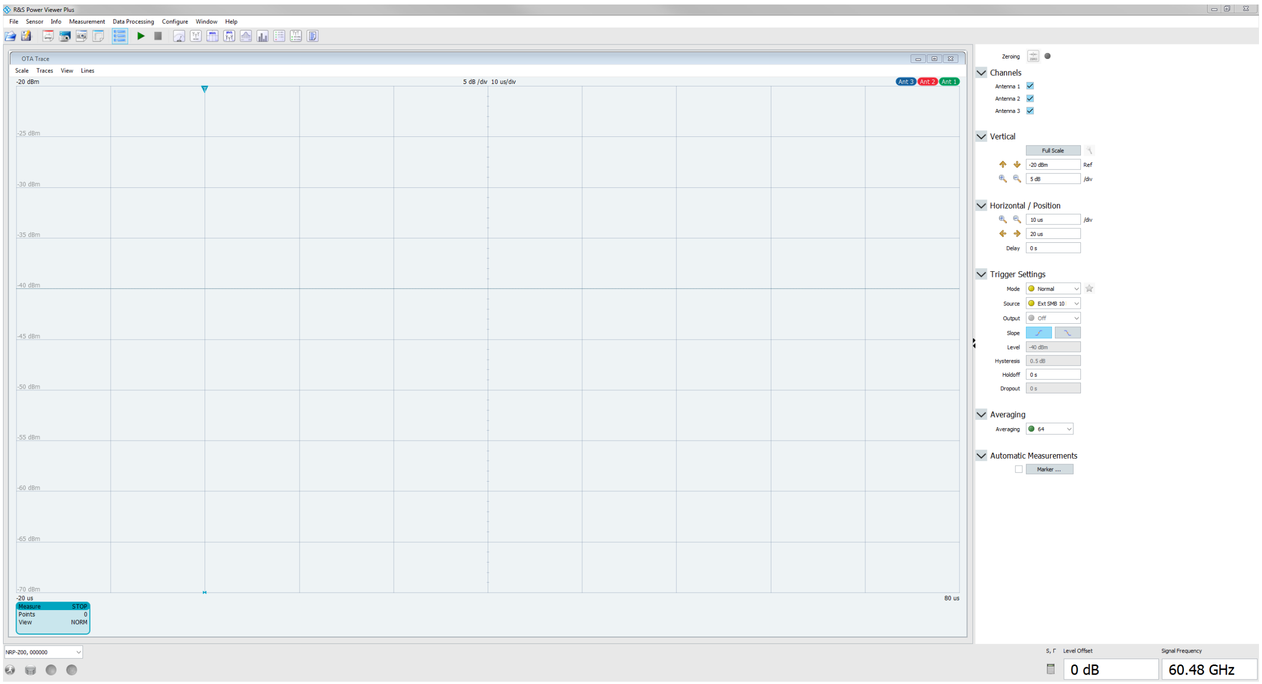


The measurement results window displays the results of the multi channel measurement.

To configure an OTA single sensor trace power measurement

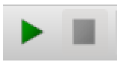
1. 

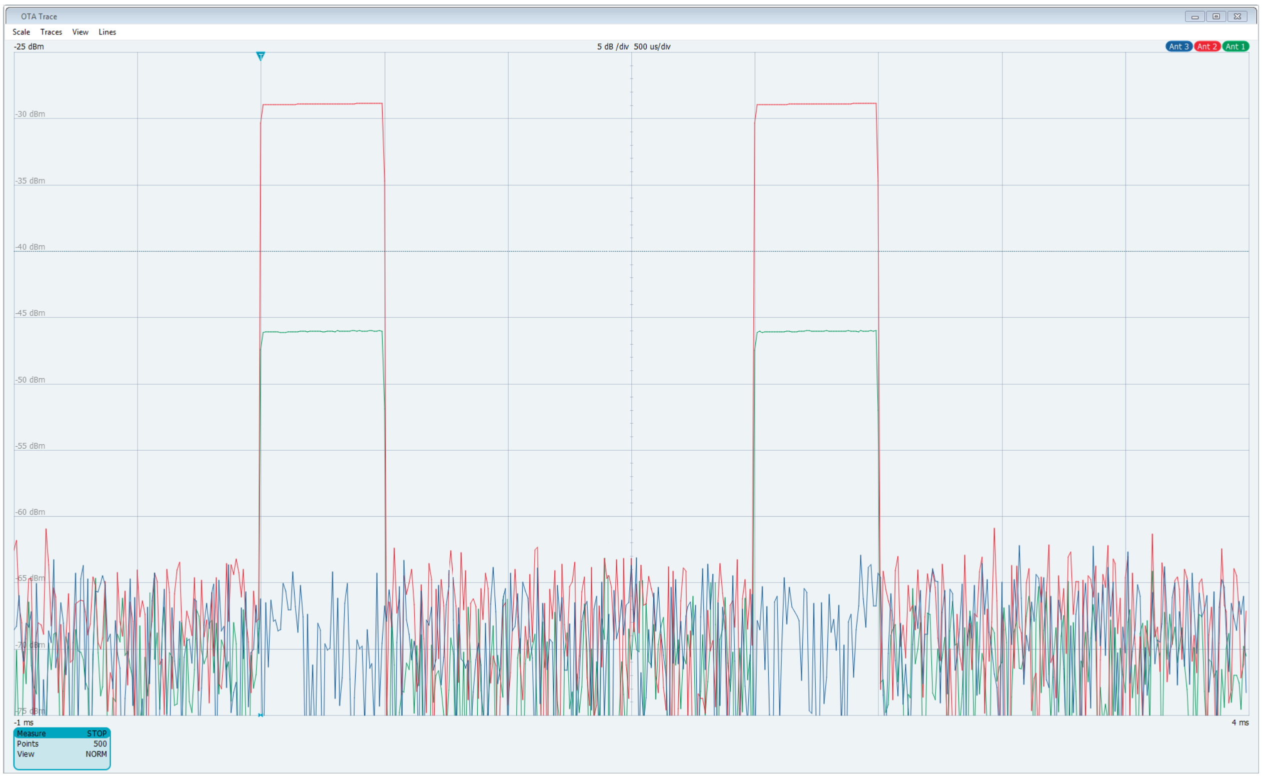
   In the toolbar, select the button to open the panel for trace measurements.



The R&S Power Viewer displays the trace measurement results windows, and the setting parameters in the panel on the right.

2. In the lower border toolbar, select the sensor module.
3. In the panel on the right, select the antenna modules for the measurement.
4. If necessary, adjust the scaling of the measurement results window.
5. If necessary, set the trigger and averaging parameters provided for "Trace" measurements.
6. Set the frequency.
7. Start the measurement.





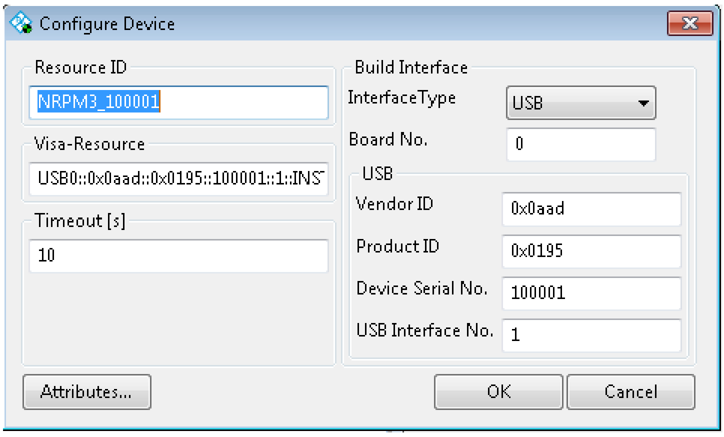
The results window displays the trace measurement results of the three antenna modules.

5.2.2 Using R&S Forum

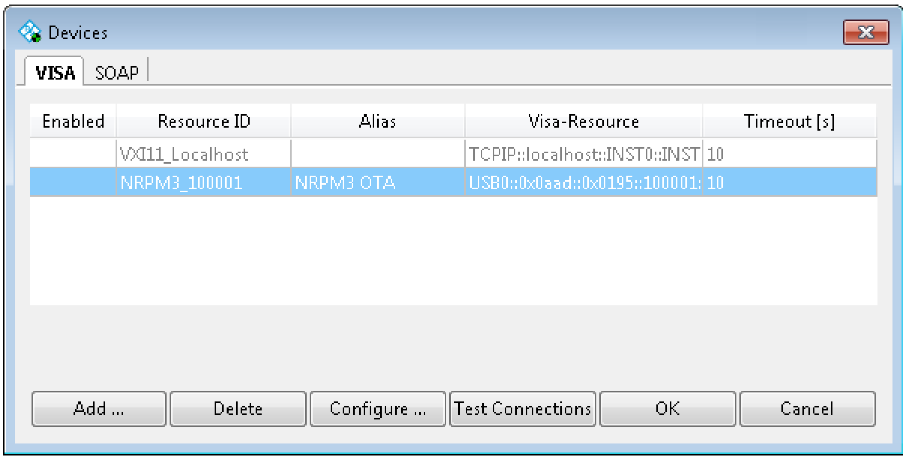
With the scripting tool R&S Forum, you can measure the power in remote control mode. This section briefly shows how to proceed by a programming example. For handling and using the tool in detail, see the documentation provided at the www.rohde-schwarz.com/forum website.

To start R&S Forum

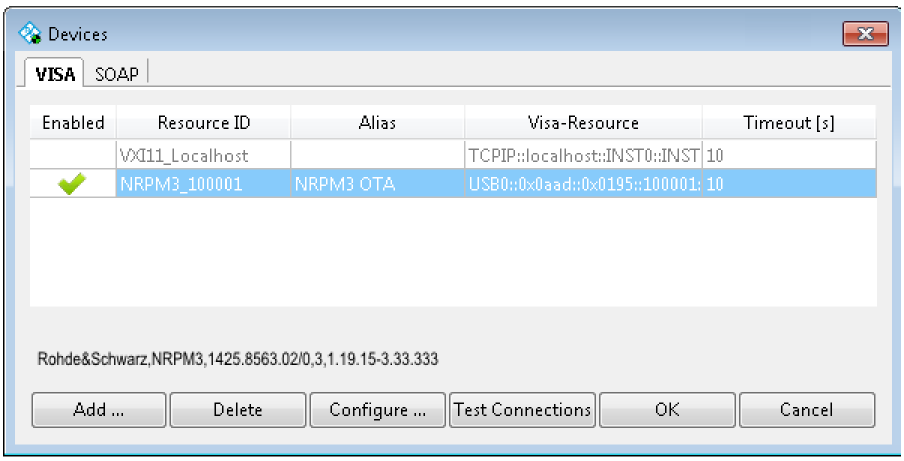
1. In the MS Windows "Start" menu, execute R&S Forum.
2. In the menu bar, select "Settings > Instruments" to establish the connection to the R&S NRPM3.
3. In the "Devices" dialog, select "Configure".



4. Enter the resource parameters.
5. Confirm with "Ok".



6. In the "Enabled" column, check the R&S NRPM3.
7. Select "Test Connection".



R&S Forum confirms the connection by displaying the response of the sensor module.

8. Close the dialog with "Ok".

   In the interface window, you can control the R&S NRPM3 using single commands, command sequences or executing loaded script files.

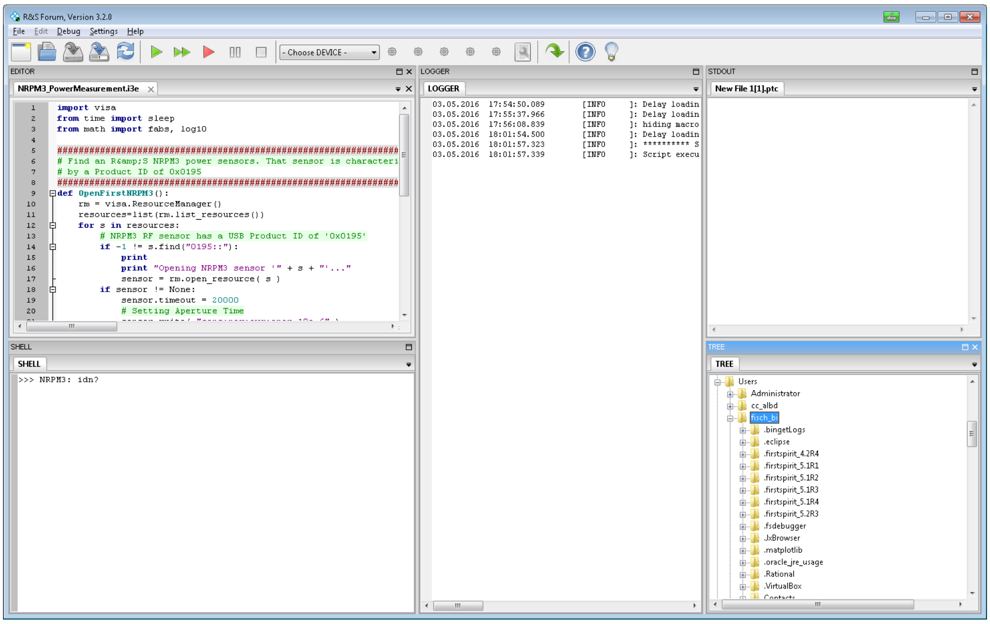


Figure 5-1: R&S Forum interface window

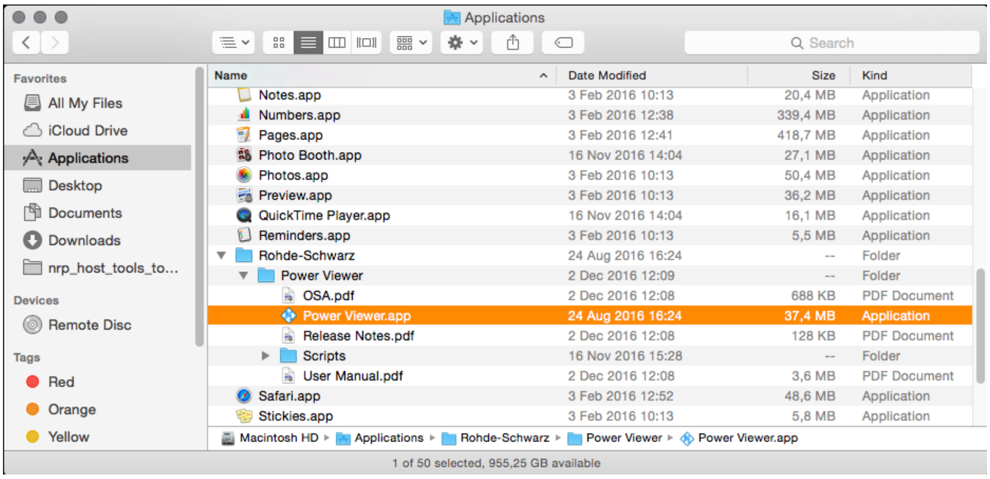
5.2.3 References

This section provides information on starting the R&S Power Viewer on operating systems other than MS Windows.

Using the R&S Power Viewer on Mac OS X

The R&S Power Viewer is part of the R&S NRP Toolkit for Mac OS X. For installation, see "To install the R&S NRP Toolkit on Mac OS X" on page 24.

1. To start the R&S Power Viewer application, select "Applications > Rohde-Schwarz > Power Viewer > Power Viewer App.".
2. Open a "Finder" window.
3. Navigate to the corresponding applications subfolder, as shown in the picture below:



4. Double-click "Power Viewer.app"
5. Connect the sensor modules to a USB port of your system.

   The measurement is set up and ready for operation.

Using R&S Power Viewer on Linux

The Linux version of R&S Power Viewer is available as an installable package for selected Linux distributions. See "To install the R&S Power Viewer on Linux" on page 25 for the installation procedure.

The R&S Power Viewer is launched by the start script `PowerViewerPlus`. The script sets the proper environment and then runs the R&S Power Viewer application.

To execute OTA measurements:

1. Start the R&S Power Viewer.
2. Connect the sensor modules to a free USB port of your system.

   The measurement is set up and ready for operation.

6 Firmware Update

6.1 Installation of New Firmware

Use the firmware update program (PureFW) to load new firmware for the sensor modules. It is part of the R&S NRP Toolkit that is provided on the Internet, e.g. on the www.rohde-schwarz.com/software/nrpm/ website.

6.2 Hardware and Software Requirements

To perform a firmware update, you need:

- PC with free USB port
- R&S NRP-ZKU interface cable
- Operating system Microsoft Windows 7, Microsoft Windows 8 or Microsoft Windows 10
- **VISA software must be installed on your PC.**
- The R&S NRP Toolkit software must be installed on your PC (includes firmware update program).
- A Rohde & Schwarz update file (*.rsu) for the sensor module must be available.

The latest firmware update files are available on the Rohde & Schwarz product website.

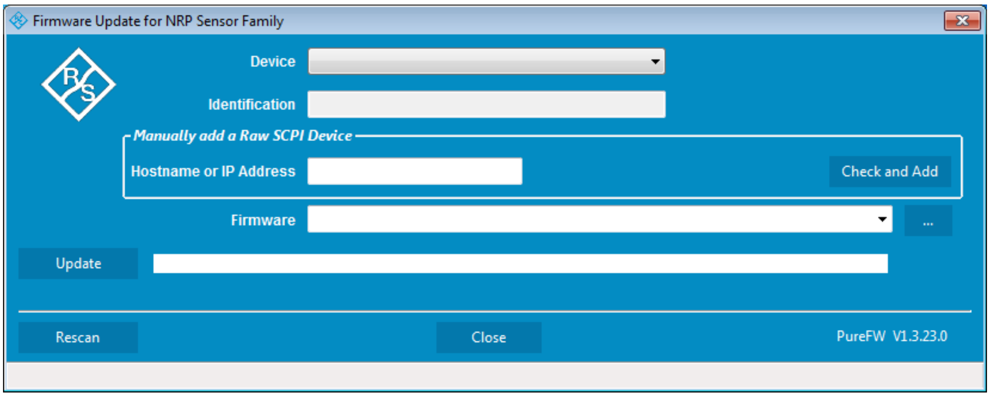
6.3 Preparation

1. Make sure that a recent VISA software is installed. You can only update the firmware with PureFW, when the device is recognized as a VISA device.
2. Connect the sensor module module to the PC using an R&S NRP-ZKU interface cable.
   Shortly afterwards, the PC identifies the new USB hardware.

6.4 Updating the Application Firmware

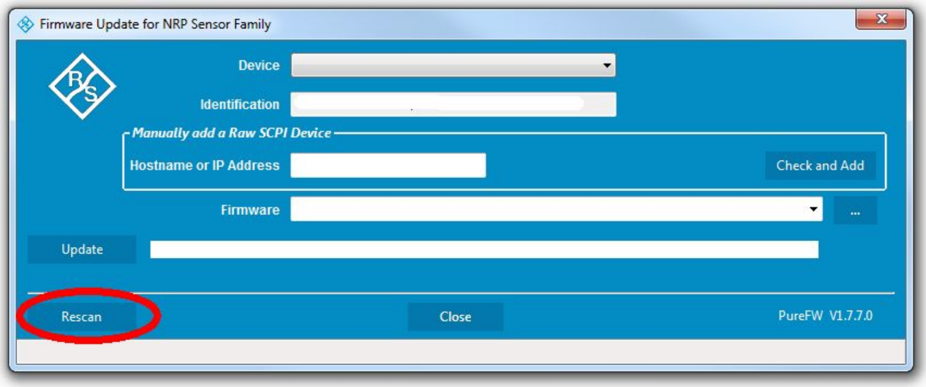
To perform a firmware update:

1. Start the firmware update program (PureFW) via "Start menu > NRP-Toolkit > Firmware Update". The following window appears:

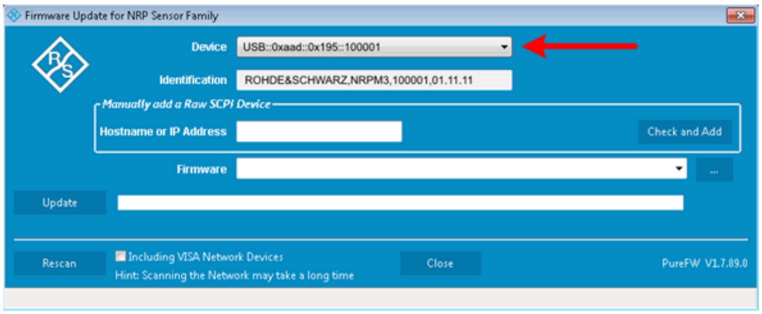


The program automatically starts scanning for R&S NRPM3 sensor modules which are connected via USB. When the scan is completed, all recognized sensor modules are listed in the "Device" dropdown control.

2. If the sensor module you want to update is not listed in the "Device" dropdown control, perform one of the following:
   a) Make sure the sensor module is physically connected.
   b) Check whether all necessary drivers are installed on the computer.
   For example, if no VISA library is installed on the computer, no VISA sensor module is accessible.
   c) Select "Rescan". This function starts a new search for USB sensor modules.

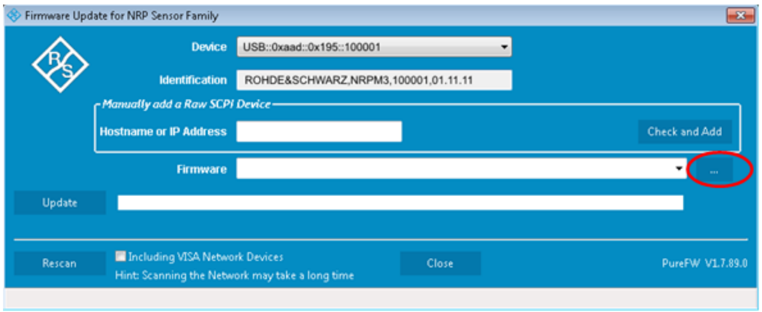


3. In the "Device" line, select the sensor module you want to update.

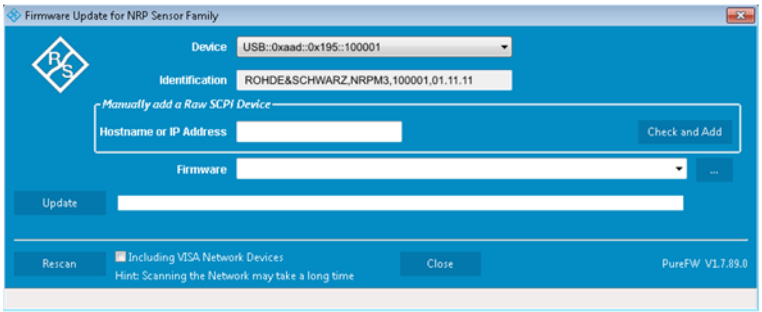


The "Hostname or IP Address" field is for adding sensor modules which are connected to the LAN. Since there is no LAN version of the sensor module, this field is not used.

4. In the "Firmware" field, enter the full path and file name of the update file or press the ellipsis button to browse the file system for it. New firmware for the R&S NRPM3 sensor modules generally has an `*.rsu` (Rohde & Schwarz Update) extension.

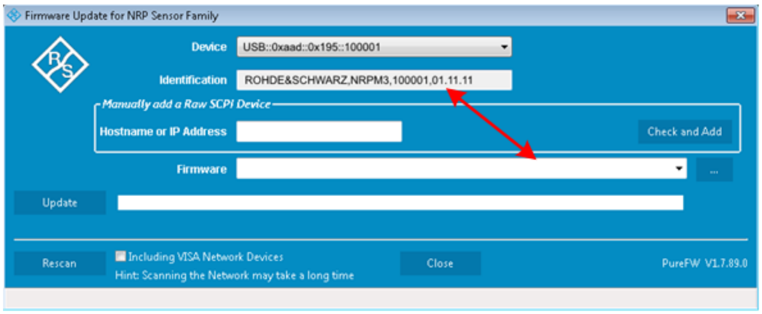


5. Select "Update" to download the new firmware and program it into the flash memory of the sensor module.



During the update process, a bar displays the state of progress. The update sequence takes a couple of minutes, depending on the sensor module model and the size of the selected file. You can also watch the status line at the bottom of the dialog, which informs about the currently executed steps.

6. Check if the update was successful. It is the case if the firmware version in the "Identification" field is the same as the one you loaded in the "Firmware" field.



NOTICE

Potential damage to the firmware of the device

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

Special care must be taken on not disconnecting the power supply while the update is in progress. Interrupting the power supply during the firmware update will most likely lead to an unusable device which needs to be sent in for maintenance.

7 Remote Control

7.1 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. Mostly, the notation used complies with SCPI specifications.

7.1.1 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 NRPM 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.

7.1.2 Notations

For a detailed description of SCPI notations, see Chapter A, "Remote Control Basics", on page 110.

Numeric suffixes <n>

If a command can be applied to multiple instances of an object, e.g. specific sensor modules, 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.

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 A.1.3, "SCPI Parameters", on page 112.

Example:

Definition: `[SENSe<Sensor>:]AVERage:COUNt:AUTO:NSRatio <nsr>`

Command: `AVER:COUN:AUTO:NSR 0.01`

Special characters | and { }

|	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. **Example**: Definition: `INITiate:CONTinuous ON	OFF` Command `INITiate:CONTinuous ON` starts the measurements Command `INITiate:CONTinuous OFF` stops the measurements
{ }	Parameters in braces may be included in the command once, several times or not at all.

7.1.3 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.

Remote Commands:

***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 STatus 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>

ReCaLl

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 theSYSTem: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` `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
`*TST?`
Query
`1`
Response: Failed

Usage: Query only

***WAI**

WAIt to continue

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

Usage: Event

7.2 Configuring the General Functions

7.2.1 Configuring the System

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

Remote commands:

SYSTem:DFPRint<Channel>?

Reads the footprint file of the sensor module.

Suffix:

<Channel> 1...4
Measurement channel if more than one channels are 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 of error numbers and a short description of the error in the first in first out order.

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

Example: `SYSTem:ERRor:ALL?`
Response: `0,"No Error"`

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 and not the entire error text.

Example: `SYSTem:ERRor:CODE:ALL?`
Response: `0`
No errors occurred since the last read out of the error queue.

Usage: Query only

SYSTem:ERRor:CODE[:NEXT]?

Queries the oldest entry in the error queue and then deletes it. The query returns only the error number, not the entire error text.

Example: `SYSTem:ERRor:CODE?`
Response: `0`
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: `SYSTem:ERRor:COUNt?`
Response: `1`
One error occurred since the last read out of the error queue.

Usage: Query only

SYSTem:ERRor[:NEXT]?

Queries the error/event queue for the oldest item and removes it from the queue. The query returns 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: `SYSTem:ERRor?`
Queries the oldest entry in the error queue.
Response: `0,"no error"`
No errors occurred since the last read out of the error queue.

Usage: Query only

SYSTem:FWUPdate <fwudata>

Loads new operating firmware into the device. Rohde & Schwarz provides new firmware in files with the extension `*.rsu`.

Usually, you find the `*.rsu` file at the Rohde & Schwarz web site. Otherwise, consult the customer support or the product marketing.

In addition, Rohde & Schwarz provides dedicated programs for loading new firmware into a R&S NRPM sensor module (e.g. PureFW).

If you want to integrate a firmware update function in your own application, use the `SYSTem:FWUPdate` command. The parameter of this command is a "Definite Length Arbitrary Block Data", containing the direct copy of the binary `*.rsu` file.

A "Definite Length Datablock" has a pre-defined format.It consist of:

- A '#' sign.
- A single digit indicating the length of the number which represents the size of the binary file
- The binary data.
- An appended delimiter (LF, 0x0a).

Example:

If you want to update the firmware of the R&S NRPM, you first need an update file, e.g. `nrpm3_FW_15.02.12.01.rsu`.

Lets assume that this file has a size of 10242884 bytes. To send the file to the sensor module for updating the firmware to the new one, your application must assemble a memory block. The memory block consists of the command, the"Definite Length Block" header, the contents of the `*.rsu` file and a trailing delimiter (0x0a = linefeed).

First, have a look at the size of the binary data; it is 10242884 in this case. This number has eight digits. Now you have all the information to assemble everything:

- The `SYST:FWUP` command
- A blank as a separator.
- The '#' sign.
- The '8' for the length of the file size.
- The '10242884' specifying the size of the file.
- .... (the contents of the `*.rsu` file)..... .
- 0x0a as a delimiter.

In this example, you would write exactly 10242905 bytes to the sensor module (for example via a 'viWrite()' function).

The result sums up from the values of the above list to:

9 + 1 + 1 + 1 + 8 + 10242884 + 1= 10242905

In a (pseudo) string notation, it is:

```
SYST:FWUP #810242884.........(file content)........ <LF> ,
```

Where `<LF>` is a single 0x0a character and `........(file content)........` is the direct byte-by-byte contents of the `*.rsu` file.

Setting parameters:
<fwudata> <block_data>

Usage: Setting only

SYSTem:FWUPdate:STATus?

While a firmware update is in progress, the LED of the sensors flashes in bright white color. When the firmware update is completed, you can read the result of the update with the `SYST:FWUP:STAT?` command.

The result of the query is a readable string.

Example: `SYSTem:FWUPdate:STATus?`
Response: `"Success"`

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:ALL?

Returns a block data with all SCPIs and the relevant parameter infos for each SCPI.

Usage: Query only

SYSTem:HELP:SYNTax? [<Item>]

Returns the relevant parameter information for the specified SCPI.

Query parameters:
<Item>

Example: `SYST:HELP:SYNT? 'sens:aver:coun'`

Usage: Query only

SYSTem:INFO? [<item>]

Returns information about the system.

`SYSTem:INFO?<string_value>` is used to query a specific information item. Without `<string_value>`, the command returns all available information in a list string, separated by commas.

`<string_value>` can query the information to the following system parameters:

- Manufacturer
- Type
- Stock Number
- Serial
- SW Build
- Sensor Name
- Technology
- Function
- MinPower
- MaxPower
- MinFreq
- MaxFreq
- Resolution
- Cal. Misc.
- Cal. Abs.
- Cal. Temp.
- Cal. Lin.
- Cal. Due Date
- Antenna 1 Type
- Antenna 1 Stock number
- Antenna 1 Serial
- Antenna 1 Property
- Antenna 2 Type
- Antenna 2 Stock number
- Antenna 2 Serial
- Antenna 2 Property
- Antenna 3 Type
- Antenna 3 Stock number
- Antenna 3 Serial
- Antenna 3 Property
- Uptime

If no antenna module is connected, the command returns "not present" for all parameters of the corresponding antenna input.

Query parameters:
<item>

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 returns 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:CHANnel<Channel>:COLor <color>

Sets the color of the antenna module LED.

The suffix <channel> selects the corresponding antenna module, and the suffix <color> selects LED on/off.

Suffix:

<Channel> 1...3

Parameters:

<color> **0**

The LED is off.

> 0

The LED shines blue.

Range: 0 to 255

*RST: 0

SYSTem:LED:COLor <color>

Sets the color of the sensor module status LED, if the LED operating mode SYSTem:LED:MODE USER) is selected.

Parameters:

<color> Range: 0 to 0x0FFFFFF

*RST: 0xA0A0A0

Example: `SYSTem:LED:MODE USER`
Selects "User" mode for the system status LED.

`SYSTem:LED:COLor #HA000A0`
Sets the LED color to magenta.

`SYSTem:LED:COLor #H00C000`
Sets the LED color to green.

`SYSTem:LED:MODE SENSor`
Sets the system status LED operating mode back to the sensor internal settings.

SYSTem:LED:MODE <mode>

Selects whether the color of the system status LED is controlled by the firmware of the sensor internally or by the user settings.

For more information, see `SYSTem:LED:COLor`.

Parameters:
<mode> USER | SENSor
*RST: SENSor

SYSTem:MINPower?

Yields the lower power measurement limit.

The lower measurement limit refers to the sensor or to the combination of a sensor and the components connected ahead of it. This query can be used 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`.

The commands are output with default values, limits and ranges.

Usage: Query only

SYSTem:PRESet

Triggers a sensor reset.

The command corresponds to the *RST command.

Usage: Event

SYSTem:REBoot

Reboots the power sensor.

Usage: Event

SYSTem:RESTart

Restarts the sensor module.

Usage: Event

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: SYSTem:SERRor: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 eldest 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
Response: 0, "reported at uptime:2942; notice; auto-averaging exceeded maximum time; Notification"

Usage: Query only

SYSTem:SERRor?

Returns the next static error (if any). Static errors are more severe than normal error conditions, which can be queried with SYSTem:ERRor[:NEXT]?. While normal errors result from, e.g. unknown commands or syntax errors and generally affect a single parameter or setting, the static errors, usually prevent the execution of normal measurements.

Usage: Query only

SYSTem:TLEVels?

Queries the possible power test levels of the sensor.

Usage: Query only

SYSTem:TRANsaction:BEGin

Starts a series of settings.

Usage: Event

SYSTem:TRANsaction:END

Ends a series of settings.

Usage: Event

SYSTem:VERSion?

Queries the SCPI version the sensor's command set complies with.

Example: `SYSTem:VERSion?`
Queries the SCPI version.
Response: `1999.0`
The sensor complies with the SCPI version from 1999.

Usage: Query only

SYSTem[:SENSor]:NAME <sensorname>

Queries the sensor name, or assigns an arbitrary alias name to the sensor.

If not specified, it defaults to the hostname of the sensor module.

Parameters:
<sensorname>

Example: `SYSTem[:SENSor]:NAME "SensorModuleInput_A90D"`
`SYSTem[:SENSor]:NAME?`
Queries the sensor name.
Response: `"SensorModuleInput_A90D"`

7.2.2 Selecting a Measurement Channel

The `SENSe:CHANnel` subsystem contains commands for selecting a channel for connected antenna modules.

Remote commands:

[SENSe<Sensor>:]CHANNel<Channel>:PRESence?

Queries for channels that are connected to an antenna module.

Suffix:

<Channel>	1...3

Usage: Query only

[SENSe<Sensor>:]CHANNel<Channel>[:ENABle] <state>

Activates channels that are connected to an antenna module.

Suffix:

<Channel>	1...3

Parameters:

<state>	*RST: ON

7.2.3 Selecting the Reference Source

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

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

Selects the source of the reference frequency.

Parameters:

<source> INTernal | EXTernal | HOST

INTernal
Internal precision oscillator.

EXTernal
External signal supplied at the host interface connector.

*RST: INTernal

Example: `ROSCillator:SOURce EXTernal`

7.2.4 Setting the Power Unit

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

UNIT:POWer <unit>

Sets the output unit for the measured power values.

Parameters:

<unit>	DBM | W *RST: W

7.2.5 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 sensor modules if high-level measurement commands are used.

Remote commands:

FORMat:BORDer <border>

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

Parameters:

<border>	NORMal | SWAPped **NORMal** The 1st byte is the LSB, the 8th byte the MSB. This format fulfills the "Little Endian" convention. **SWAPped** The 1st byte is the most significant byte (MSB), the 8th byte the least significant byte (LSB). This format fulfills the "Big Endian" (big end comes first) convention. *RST: NORMal

---

FORMat:SREGister <sregister>

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

Parameters:

<sregister>	ASCii | HEXadecimal | OCTal | BINary *RST: ASCii

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

Specifies whether block data is transferred in plain text or binary format.

Parameters:

<data,length> ASCii | REAL

ASCii
Transmits data as character strings in plain text.

REAL
Transmits data in binary blocks with 32 bit or 64 bit length.

*RST: ASCii

<length> Range: 32, 64
*RST: 32

Example:
```
FORMat:DATA REAL,64
```
Binary DOUBLE

```
FORMat:DATA ASCii,3
```
3.124

7.3 Controlling the Measurement

The sensor module 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 retrieve each new average value as a measurement result or do you want to bundle more measured values into one result?

The following chapter introduces in general the principle of triggering and the controlling mechanisms for the output of the measurement results.

7.3.1 Triggering

In a basic continuous measurement, the measurement is started immediately after the initiate command, see also "Waiting for a trigger event" on page 65. 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.

Trigger states

The sensor module has trigger states to define the exact start and stop times of a measurement and the sequence of a measurement cycle.

The following states are defined:

- Idle
  The sensor module does not perform a measurement. After switching on, it is initially in the idle state.

Before a trigger can be executed, set the sensor module to the waiting for trigger state, using one of the commands described under "Waiting for a trigger event" on page 65.

- Waiting for trigger
  The sensor module waits for a trigger event.
  The occurrence of the trigger depends on the selected trigger source, see "Trigger sources" on page 65.
  When the trigger event occurs, the sensor module starts measuring.
- Measuring
  The sensor module was triggered and is measuring. After completing the measurement, the sensor module exits the measurement state and continues depending on the set initiate commands, see Waiting for a trigger event.

Waiting for a trigger event

Depending on how many measurement cycles you want to execute, select the corresponding command to set the sensor module to the waiting for trigger state.

- `INITiate:CONTinuous` on page 89
  Starts a new measurement cycle automatically after the previous one has been completed.
- `INITiate[:IMMediate]` on page 89
  Starts as many measuring cycles in succession as set with the command `TRIGger:COUNt` on page 91.
  With `TRIGger:COUNt 1` for example, the command starts a single measurement cycle that renders one result. Every time you send this command, a new measurement cycle starts.

Trigger sources

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

If the signal power exceeds or falls below a reference level set by the trigger level, the measurement starts 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.	`TRIGger:IMMediate`
Immediate	Measures immediately, does not wait for trigger condition.	-
Internal	Uses the input signal as trigger signal.	`TRIGger:IMMediate`
External 1	Uses the digital input signal supplied using a differential pair in the 8-pin sensor module cable.	`TRIGger:IMMediate`

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

7.3.2 Controlling the Measurement Results

The R&S NRPM3 sensor module 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 to retrieve the measurement results.

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

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

Repeating termination control

Returns a measurement result when the measurement has been completed. The number of measurement cycle repetitions is equal to the set average count. If the average count is large, the measurement time can take long time.

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

Moving termination control

Returns intermediate values to facilitate early detection of changes in the measured quantity. For each partial measurement, the sensor module returns a new average value as a measurement result. Thus, the measurement result is a moving average of the last partial measurements. The parameter average count defines how many partial measurements are included.

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

7.3.3 Interplay of the Controlling Mechanisms

The following examples use continuous measurement scenarios. But these scenarios apply also to single measurements. The only difference is that a single measurement is not repeated.

7.3.3.1 Continuous Average Mode

General settings for these examples:

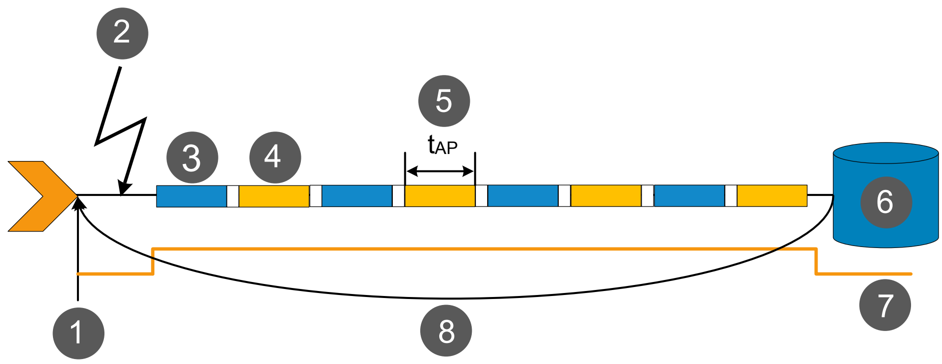
- INITiate:CONTinuous On
- [SENSe<Sensor>:]AVERage:COUNt 4
- [SENSe<Sensor>:]AVERage:COUNt:AUTO Off

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 x t_{AP}) ≙ length of one chopper phase
6 = Measurement result
7 = Trigger synchronization
8 = Return to the start of the measurement cycle

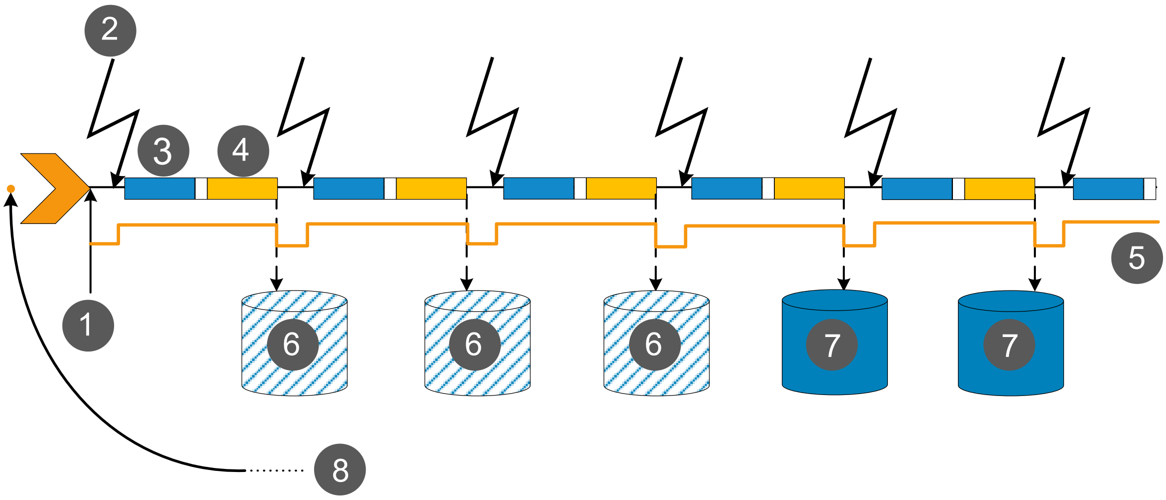
Example: Moving termination control

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), the sensor module 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

7.3.3.2 Trace Mode

General settings for the first two examples:

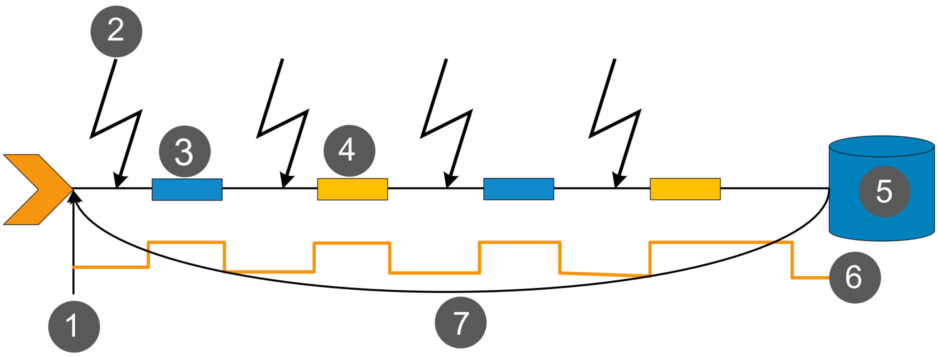
- INITiate:CONTinuous On
- [SENSe<Sensor>:]AVERage:COUNt 2

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.



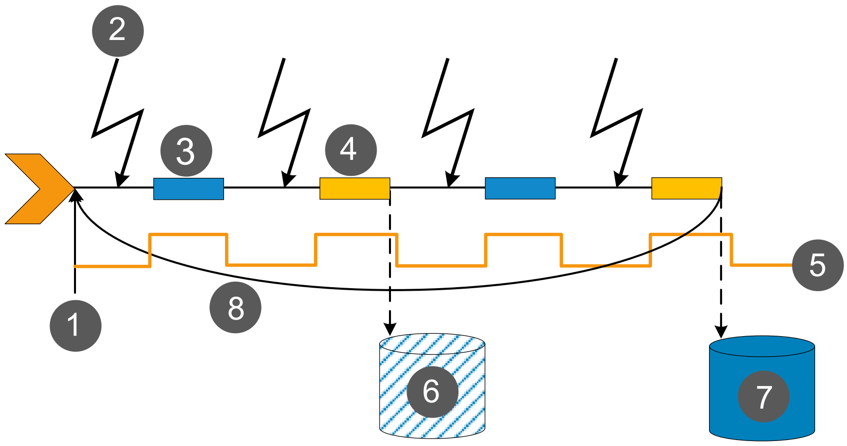
1 = Start of the measurement cycle
2 = Trigger event
3 = Noninverted chopper phase
4 = Inverted chopper phase
5 = Measurement result
6 = Trigger synchronization
7 = Return to the start of the measurement cycle

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.

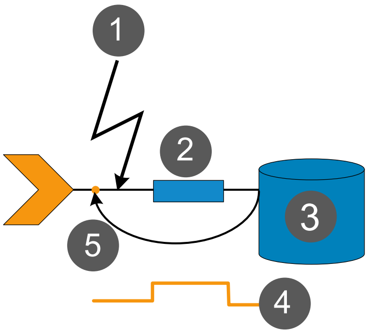


1 = Start of the measurement cycle
2 = Trigger event
3 = Noninverted chopper phase
4 = Inverted chopper phase
5 = Trigger synchronization
6 = Measurement result before average count is reached
7 = Averaged measurement result after average count is reached
8 = Return to the start of the measurement cycle

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 sensor module returns to the idle state.



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

7.4 Selecting a Measurement Mode and Retrieving Results

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 Chapter 7.5.1, "Configuring a Continuous Average Measurement", on page 75.
- Trace ("XTIMe:POWer"): A sequence of measurements is performed, see Chapter 7.5.2, "Configuring a Trace Measurement", on page 77.

After mesasuring, you can query the measurement results with the correspondent FETCh command.

Example: Selecting a measurement mode

```
*RST

// Select continuous average mode
SENSe:FUNCtion "POWer:AVG"

INITiate
FETCh?
```

Remote commands:

[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`.

XTIMe:POWer
Trace
In this mode, power over time is measured. Therefore several 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<Channel>[:SCALar][:POWer][:AVG]?

Queries the measurement results of a particular channel of the sensor module.

Suffix:

<Channel> 1...3

Usage: Query only

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

Queries the measurement results of all channels of a sensor module.

The sensor module returns the results in a string, separated by commas.

Usage: Query only

CALCulate:FEED <mode>

When you query measurement data using FETCh<Channel>[:SCALar][:POWer][:AVG]?, the sensor module returns data of the measurand that was configured before, but it can also return data of different measurands. By default, the reading is the average power.

To determine the measurand for the FETCh<Channel>[:SCALar][:POWer][:AVG]? command, use the CALCulate:FEED before initiating the measurement.

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
"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:

```
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?
```

CALCulate:MATH[:EXPRession] <mode>

Selects the measurement quantity for the power of the incident electromagnetic wave towards the antenna module.

String	Meaning
`"PISotropic"`	Equivalent isotropically received power P_{ISO} in W or dBm (default): Equivalent detected power of an isotropic antenna with an ideal power detector at the phase center location of the R&S NRPM antenna module assuming radiation only from boresight direction. $P_{ISO}=P_i$
`"PDENsity"`	Power density S in W/m², calculated as: $S=(4\pi/\lambda^2)\cdot P_{ISO}$
`"EFDensity"`	Electric field strength E_{eff} in V/m, calculated as: $E_{eff}=\sqrt{(S\cdot Z_0)}$ with: Z_0=376,73 Ω.
`"MFDensity"`	Magnetic field strength H_{eff} in A/m, calculated as: $H_{eff}=\sqrt{(S/Z_0)}$ with: Z_0=376,73 Ω.
`"DETPower"`	Power at the internal RF detector in W or dBm: The measurement result without including antenna gain and frequency-dependent calibration factors is returned. $P_{Det}=(G_{RF,i}/k_{att,A,i}(f))\cdot P_{ISO}$

Parameters:

<mode> *RST: "PISotropic"

Example: `CALC:MATH "PDEN"`
Selects power density in W/m² as the equivalent resulting unit.

CALCulate:MATH[:EXPRession]:CATalog?

Lists all supported calculation functions.

The result is a list of strings, separated by comma.

Example: `CALC:MATH:CAT?`
Response: `"PISotropic","PDENsitiy","EFDensity","MFDensity","DETPower"`

Usage: Query only

[SENSe<Sensor>:]AUXiliary <mode>

Activates 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
By averaging the measured values in the sensor module, extreme values are lost.

RNDMax
In contrast to `MINMax`, the value of a randomly selected sample is returned instead of the min value. All evaluations occur using these values instead of the average values.

*RST: NONE

7.5 Configuring the Measurement Modes

The following chapter describes the settings needed for configuring a measurement mode.

7.5.1 Configuring a Continuous Average Measurement

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

Reducing noise and zero offset

The continuous average measurement can be performed with chopper stabilization to obtain more accurate results with reduced noise and zero offset. When chopper stabilization is used, a single measurement is performed over two sampling windows, the polarity of the detector output signal being reversed for the second window. By taking the difference of the output signals, the effect of the video path on noise and zero drift is minimized.

The smoothing filter can further reduce result fluctuations caused by modulation. But when activated it increases the inherent noise of the sensor module by approx. 20%, so it should remain deactivated if it is not 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 for the fluctuations caused by modulation to be 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 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

The measurement time is calculated as follows:

*MT = 2 * AC * APER + (2 * AC - 1) * 100 µs*

with:

MT: overall measurement time

AC: average count

APER: aperture time

100 µs is the time for switching the chopper phase.

Remote commands:

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

Sets the aperture, sampling window (time interval) for the continuous average mode. The aperture time defines the length of the unsynchronized time interval used to measure the average signal power.

Parameters:

<integration_time>	Range:	10.0e-6 to 2.00
	*RST:	0.02
	Default unit:	s

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

Activates 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: `SMOothing:STATe OFF`

7.5.2 Configuring a 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 <count>

Sets the filter length, the number of readings to be averaged for one measured value for trace mode. The higher the count the lower the noise and the longer it takes to obtain a measured value.

Parameters:

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

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

Sets the termination control mode for trace mode, defining how you retrieve the measurement results.

When a new measured value is shifted to the FIR filter, a new average value is available at the filter output. It is obtained from the new measured value and the other values stored in the filter. See also Chapter 7.3.2, "Controlling the Measurement Results", on page 66.

Parameters:

<mode> MOVing | REPeat

MOVing
Returns each new average value as a measurement result.
This mode is suitable for measurements, where tendencies in the result have to be recognized during the measurement procedure.

REPeat
Returns a new result after the FIR filter has been filled with new measured values.
This mode is suitable to avoid redundant results in measurements.

*RST: REPeat

Example: `TRACe:AVERage:TCONtrol REPeat`

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

Activates the averaging filter in trace mode.

Parameters:

<state> *RST: ON

[SENSe<Sensor>:]TRACe:DATA?

Returns the measured trace data in a pre-defined format. Unlike the `FETCh?` command, this command considers the settings of `[SENSe<Sensor>:]AUXiliary` on page 74 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 R&S NRPM sensor module can measure additional measurands like "Minimum", "Maximum" or "Random". These additional measurands are denoted as auxiliary measurands and can be selected with the `[SENSe<Sensor>:]AUXiliary` on page 74 command.

A trace measurement with the R&S NRPM can therefore be configured to return up to three measurands on each channel/antenna module. As a consequence, the resulting data which is provided by a `[SENSe<Sensor>:]TRACe:DATA` query can contain between 1 and 9 blocks of measurement data. For example, 1 resulting measurement data block if only one antenna module is used and only average power is to be measured on that channel. 9 resulting measurement data blocks if three antenna modules are used and average, min & max power is to be measured on each channel.

To obtain a single measurement data block, you must first extract the contents of the "Definite Length Arbitrary Block" which is returned from the `SENSe:TRACe:DATA` query. The format is defined in IEEE488.2.

In principle the response has the format as shown in Figure 7-1:

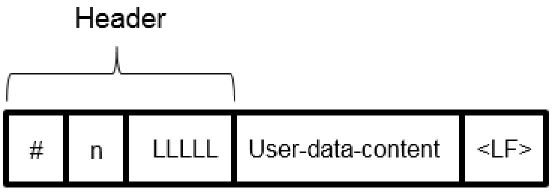


Figure 7-1: Response format

- The header consists of:
  – The character #.
  – A single digit ('n') which defines the number of the following digits taken as the size of the content.
  – A number consisting of as many digits as 'n' specified ('LLLLL'). This number determines the size of the content.
- The content ('user-data-content'), see also Figure 7-2. There are as many bytes as 'LLLLL' specified.
- A single linefeed character (symbolically shown as <LF>, Response format).

Example:

The "Arbitrary Block Response Data" for a user data that contains 45182 bytes would be:

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

The "Arbitrary Block Response Data" for a "user-data-content" `'THIS IS A TEST'` would be:

```
#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 one or more sections with trace measurement results, depending on the selection of auxiliary measurands.

Each section is made up of a certain sequence. The following example assumes that the number of data points has been chosen as 260.

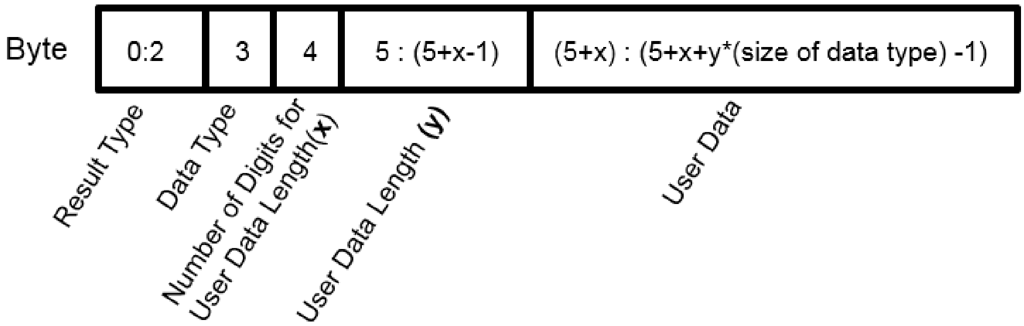


Figure 7-2: User-data-content format

y = number of values which follow the header
x = number of digits of y

Example:

```
C1Af3260xxxxyyyy
C1                    Channel 1
  A                   The letter 'A' to denote
                        the Average-Trace
   f                  The letter 'f' to denote
                        float format
    3                 3 bytes for length of the number
                        of points that follows
     260              260 float values (4 bytes each)
        xxxxyyyy...   260 float values...
```

This example shows one of up to nine measurement data blocks which could be contained in the "user-data-content". Referring to the example above, if the min and max traces are also selected (by `SENS:AUX MINMAX`) the subsequent data would contain:

```
C1mf3260mmmmnnnn
C1                    Channel 1
  m                   The letter 'm' to denote
                        the Min-Trace
   f                  The letter 'f' to denote
                        float format
    3                 3 bytes for length of the
                        number of points that follows
     260              260 float values (4 bytes each)
        mmmmnnnn...   260 float values...

C1Mf3260gggghhhh
C1                    Channel 1
  M                   The letter 'M' to denote
                        the Max-Trace
   f                  The letter 'f' to denote
                        float format
    3                 3 bytes for length of the
                        number of points that follows
     260              260 float values (4 bytes each)
        gggghhhh...   260 float values...
```

For further information, see the `...[NRPM3]\Trace-M3\Visa\visaTraceBinaryData-M3` project from the accompanying programming examples (included). The source-code shows how to extract the measurement data from the "user-data-content".

Usage: Query only

[SENSe<Sensor>:]TRACe:MPWidth?

Queries the attainable time resolution of 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>

Sets the relative position of the trigger event in relation to the beginning of the trace measurement sequence. It is used to specify the start of recording for trace mode.

The start of recording is referenced to the delayed trigger point that is set with `TRIGger:DELay`. Negative values indicate that the start of recording occurs before the trigger point.

Parameters:

<time>	Range:	-3.0 to 3.0
	*RST:	0.0
	Default unit:	s

[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>

Activates realtime processing of the trace mode.

If disabled, each measurement from the sensor module 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, i.e. 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, i.e. the time to be covered by the trace sequence. This time period is divided into several 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

[SENSe<Sensor>:]TRACe:UPSample[:TYPE] <type>

Selects an output mode for the acquired trace data.

Parameters:

<type> SINC | HOLD

HOLD
Returns the trace data unchanged. The course of power over time is represented as sampled by the sensor module's data acquisition and processing logic.

SINC
Returns the trace data as the result of a SINC interpolation of the acquired samples. This setting is only reasonable if the selected number of trace points is higher than the number of samples which results from the internal sample rate in the selected trace time. The SINC interpolation reconstructs the original signal sequence best from the sampled values. Use it to get a smooth resulting curve.

*RST: SINC

Example:

`SENS:TRAC:TIME 100E-6`
Sets the trace time to 100 µs. With an internal sample rate of 2 MHz, the sensor module performs 200 measurements.
`SENS:TRAC:POIN 400`
`SENS1:TRACe:UPS:TYPE SINC`
If you select the number of trace points higher than the number of physical measurements, you achieve a smooth result curve using the `SINC` interpolation.

`SENS:TRAC:POIN 200`
`SENS1:TRACe:UPS:TYPE SINC`
If the trace points are up to the number of physical measurements, the `SINC` interpolation does not affect the trace results.

7.6 Configuring Basic Measurement Parameters

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

7.6.1 Configuring Auto Averaging

This chapter includes the commands required for automatic averaging in the continuous average measurements.

Remote commands:

[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 parameter, i.e the number of measured values that have to be averaged for forming the measurement result.

Averaging is only effective, when `[SENSe<Sensor>:]AVERage[:STATe]` is turned on.

Parameters:

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

Example: `AVERage:COUNt 4`

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

Selects the mode that determines the average count. Average count is often also called averaging factor, but it designates the same parameter: the number of measured values that have to be averaged for forming the measurement result.

Parameters:

<state>

ON
Auto averaging: the averaging factor is continuously determined and set depending on the power level 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: OFF

[SENSe<Sensor>:]AVERage:COUNt:AUTO:MTIMe <maximum_time>

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

Parameters:

<maximum_time>	Range:	0.01 to 999.99
	*RST:	4.00

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

Sets the maximum noise ratio in the measurement result.

This value is considered in the auto averaging calculation when [SENSe<Sensor>:]AVERage:COUNt:AUTO is ON and for [SENSe<Sensor>:]AVERage:COUNt:AUTO:TYPENSR.

Parameters:

<nsr>	Range:	100.000e-6 to 1.00
	*RST:	0.01

[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:

- A 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 case, previous measurement results 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: AVERage:RESet

Usage: Event

[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 considered for [SENSe<Sensor>:]AVERage:COUNt:AUTO ON and for [SENSe<Sensor>:]AVERage:COUNt:AUTO:TYPE RES.

Parameters:

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

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

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

When a new measured value is shifted to the FIR filter, a new average value is available at the filter output. It is obtained from the new measured value and the other values stored in the filter.

Parameters:

<mode> MOVing | REPeat

MOVing
Provides every new average value at the output as a measurement result.
This mode is suitable for measurements, where tendencies in the result have to be recognized during the measurement procedure.

REPeat
A new result is output after the FIR filter has been filled with new measured values.
This mode is suitable for measurements, where no redundant information has to be output.

*RST: REPeat

Example: `AVERage:TCONtrol REPeat`

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

Sets the automatic averaging filter mode.

Parameters:

<type> RESolution | NSRatio

RESolution
The usual mode for the sensor modules.

NSRatio
Predefines the compliance to an exactly defined noise component.

*RST: RESolution

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

Activates the averaging filter for the continuous average mode.

Parameters:

<state> *RST: ON

7.6.2 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.

The data of the connected antenna module determines the frequency limits.

Parameters:

<frequency> Range: antenna-specific
*RST: 10.0e9
Default unit: Hz

Example: `FREQuency 10000`

7.6.3 Configuring Corrections

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

7.6.3.1 Duty Cycle Corrections

The duty cycle is the percentage of one period in 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 <duty_cycle>

Effective in continuous average mode.

Sets the duty cycle for measuring pulse-modulated signals. The duty cycle defines the percentage of one period when the signal is active.

If the duty cycle is enabled, the R&S NRPM calculates the signal pulse power from the average power considering the duty cycle in percent.

Parameters:

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

Example: `CORRection:DCYCle 0.01`

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

Activates the duty cycle correction for the measured value.

Parameters:

<state> *RST: OFF

Example: `CORRection:DCYCle:STATe ON`

7.6.3.2 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 module (or the coupling attenuation of a directional coupler) is considered with a positive offset, i.e. the sensor module calculates the power at the input of the attenuator or the directional coupler. You can use a negative offset to compensate the influence of an upstream amplifier.

Remote commands:

[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

Example: `CORRection:OFFSet 2`

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

Activates the offset correction.

Parameters:

<state> *RST: OFF

Example: `CORRection:OFFSet:STATe ON`

7.7 Starting and Ending a Measurement

ABORt

Immediately interrupts the current measurement.

Depending on the selected measurement mode, the trigger system of the sensor module exits the measuring state and switches to:

- Idle state
  If a single measurement was previously initiated with command `INITiate[:IMMediate]`.
- Waiting for trigger
  If a continuous measurement was initiated with `INITiate:CONTinuous ON`.
  The sensor module starts the next measurement when a trigger event occurs that meets the set trigger conditions.

Usage: Event

INITiate:ALL
INITiate[:IMMediate]

Starts a single measurement cycle. The sensor module changes from the idle state to the wait for trigger state. When the trigger condition is fulfilled, the sensor module starts meausring. Depending on the number of trigger events that are required, e.g. for averaging, the sensor module enters the wait for trigger state several times. When the entire measurement is completed, the measurement readings are available, and the sensor module returns to the idle state.

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

Example: See Chapter 8.2, "Performing Measurements in Continuous Average Mode", on page 103.

Usage: Event

INITiate:CONTinuous <state>

Activates the continuous measurement mode. In continuous measurement mode, the sensor module does not change to idle state after a measurement cycle has been completed, but enters the wait for trigger state. When a trigger event occurs, it starts the next measurement cycle. This mode is also known as free-running mode, although each measurement cycle depends on the trigger conditions.

Parameters:

<state>

ON
Measures continuously. If a measurement is completed, the sensor module enters the wait for trigger state again.

OFF
Stops the continuous measurement mode. The sensor module switches to idle state.

*RST: OFF

Example: See Chapter 8.2, "Performing Measurements in Continuous Average Mode", on page 103.

7.8 Configuring the Trigger

Further information:

- Chapter 7.3, "Controlling the Measurement", on page 64

Remote commands:

TRIGger:ATRigger:DELay <delay>

For `TRIGger:ATRigger:STATe ON`, sets the delay between the artificial trigger event and the start of the measurement.

Parameters:

<delay>

Range: 0.1 to 5.0
*RST: 0.3
Default unit: s

TRIGger:ATRigger:EXECuted?

Queries the number of measurements which were triggered automatically, provided TRIGger:ATRigger:STATe is activated.

In scalar measurements, this number can only be 0 or 1. If you execute a buffered measurement the 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>

Activates the artificial trigger.

An artificial trigger is generated if the delay time set with TRIGger:ATRigger:DELay has elapsed after the start of measurement and no trigger event has occurred.

Parameters:

<state> *RST: OFF

Example: `TRIG:ATR:STAT ON`

TRIGger:COUNt <count>

Sets the number of measurement cycles to be performed when the measurement is started with TRIGger:IMMediate. This number equals the number of results which can be obtained from the sensor module after a single INITiate[:IMMediate]. As long as the defined number of measurements are not yet executed, the sensor module automatically initiates another measurement internally after the current result is available.

Use this command in particular in conjunction with buffered measurements, for example, to fill a buffer with a predefined size with measurements which have been triggered externally or with the command *TRG without the overhead of sending multiple measurement starts (INITiate[:IMMediate]).

Parameters:

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

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>

Activates the automatic setting of the delay time.

If activated, the measurement starts only after the R&S NRPM has settled. is not

The function determines the delay value automatically. It is ignored if the set `TRIGger:DELay` time is longer than the automatically determined value.

Parameters:

<state> *RST: OFF

TRIGger:DTIMe <dropout_time>

Sets the drop-out time.

With a positive (negative) trigger slope, the dropout time is the minimum time for which the signal must be below (above) the power level defined by`TRIGger:LEVel` and `TRIGger:HYSTeresis` before triggering can occur again. As with the "Holdoff" parameter, unwanted trigger events can be excluded. The set dropout time only affects the INTernal trigger source.

The dropout time parameter is useful when dealing with, for example, GSM signals with several active slots `TRIGger:DTIMe`. When performing a measurement in sync with the signal, a trigger event is to be produced at A, but not at B or C.

As the RF power between the slots is below the threshold defined by `TRIGger:LEVel` and `TRIGger:HYSTeresis`, the trigger hysteresis alone cannot prevent triggering at B or at C.

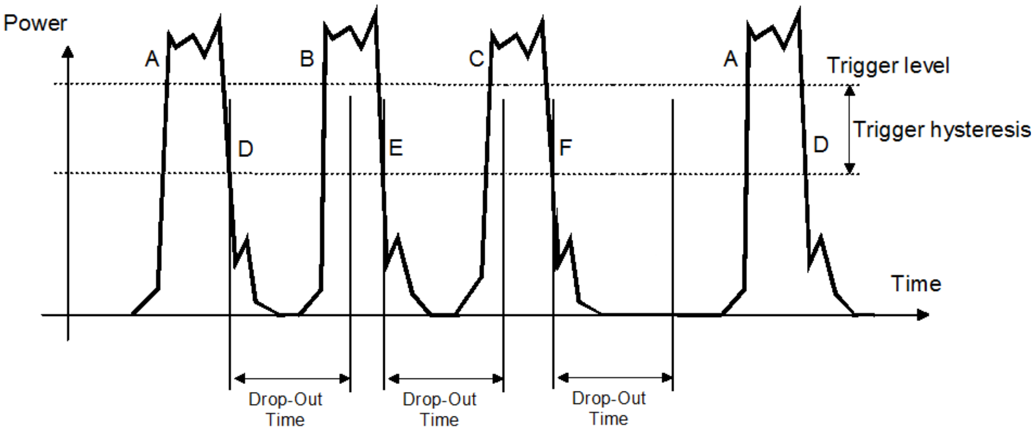


Figure 7-3: Significance of the drop-out time parameter

As the mechanism associated with the dropout time parameter is reactivated whenever the trigger threshold is crossed, unambiguous triggering can also be obtained for many complex signals. By contrast, all triggering is suppressed during the hold-off time. For the example described, this would mean that although stable triggering conditions could be obtained with a suitable hold-off time (regular triggering at the same point), it would not be possible to set exclusive triggering at A.

Parameters:

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

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

Sets the termination resistance of the second external trigger input (`EXTernal2`). You can select between `HIGH` (~ 10 kOhm) and `LOW` (50 Ohms) to fit the impedance of the trigger source and thus minimize reflections on the trigger signals.

Suffix:

<2...2> 2...2

Parameters:

<impedance> HIGH | LOW
*RST: HIGH

TRIGger:HOLDoff <holdoff>

Sets the hold off time, a time period after a trigger event. During the hold off time all further trigger events are ignored, see Figure 7-4.

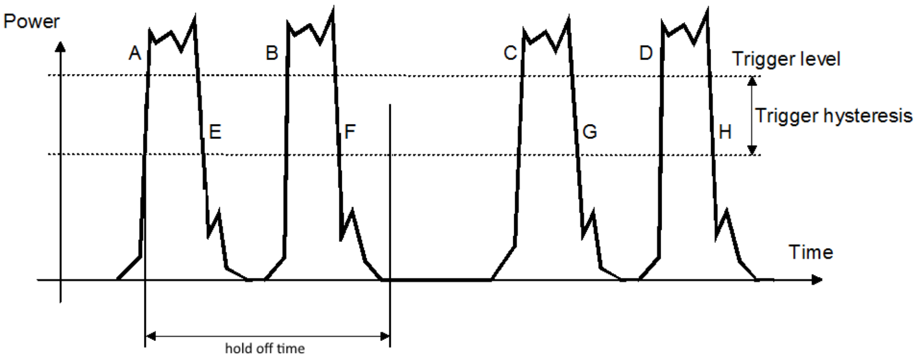


Figure 7-4: Effect of the trigger holdoff time

Parameters:

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

TRIGger:HYSTeresis <hysteresis>

Sets the hysteresis thresholds. 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.

You can use the hysteresis to eliminate the effects of noise in the signal on the edge detector of the trigger system.

Parameters:

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

TRIGger:IMMediate

Triggers a generic trigger event that causes the sensor module to exit the WAIT_FOR_TRIGGER state immediately, irrespective of the trigger source and the trigger delay and start the measurement. The command is the only means of starting a measurement when the trigger source is set to HOLD. Only one measurement cycle is executed irrespective of the averaging factor.

Usage: Event

TRIGger:LEVel <level>

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

This setting is effective only for the internal `TRIGger:SOURce`.

Parameters:

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

TRIGger:MASTer:PORT <master_port>

Selects the port where the R&S NRPM provides its internal trigger signal at the output, provided it is trigger master (see `TRIGger:MASTer:STATe`).

You can assign either the port `EXTernal[1]` or `EXTernal2` for the trigger master. If you want to trigger the master sensor module externally with its own trigger signal, assign the trigger source to the other port accordingly, e.g:

```
TRIGger:MASTer:PORT    EXT1
TRIGger:SOURce         EXT2
TRIGger:MASTer:STATe   ON
```

or

```
TRIGger:MASTer:PORT    EXT2
TRIGger:SOURce         EXT1
TRIGger:MASTer:STATe   ON
```

Parameters:

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

TRIGger:MASTer:STATe <state>

Activates the trigger master mode of the sensor module. In this state the sensor module can output a digital trigger signal in synchronization with its own trigger event.

If activated, select the output port 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, since the R&S NRPM provides two external trigger connectors. If you trigger the 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>

Available only if `TRIGger:SOURce INTernal/EXTernal`.

Determines what is used for triggering, depending on the trigger source:

- `TRIGger:SOURce INTernal`: uses the rising or falling edge of the envelope power.
- `TRIGger:SOURce EXTernal`: uses the increasing voltage.

Parameters:

<slope> POSitive | NEGative
*RST: POSitive

TRIGger:SOURce <source>

Selects the source for the trigger event detector.

Parameters:

<source> HOLD | IMMediate | INTernal | INT1 | INT2 | INT3 | INTernal1 | INTernal2 | INTernal3 | BUS | EXTernal | EXT1 | EXTernal1 | EXT2 | EXTernal2

BUS
Starts triggering with the commands *TRG or TRIGger:IMMediate, where TRIGger:IMMediate shortens the measurement. In this case, the other trigger settings are meaningless.

EXTernal
Initiates triggering via the hardware trigger bus, e.g via the base unit. Waiting for a trigger event can be skipped by TRIGger:IMMediate.
EXT, EXT1, EXTernal and EXTernal1 denote the same, an external trigger applied through the round 8 pin connector.
EXT2 and EXTernal2 refer to external triggering initiated by the dedicated SMB type connector, TRIG2 I/O, in the rear of the sensor module.

HOLD
Starts triggering with the command TRIGger:IMMediate.

IMMediate
Starts the measurement immediately.

INTernal
Starts triggering by the measurement signal.
When this signal exceeds (TRIGger:SLOPe POSitive) or drops below (TRIGger:SLOPe NEGative) the power set with TRIGger:LEVel, the measurement starts after the TRIGger:DELay has elapsed. Similar to trigger source EXT, waiting for a trigger event can also be skipped by TRIGger:IMMediate.

*RST: IMMediate

TRIGger:SYNC:PORT <sync_port>

Selects the external connection for the sensor module's sync output, see also 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 activated, blocks the external trigger bus as long as the sensor module remains in the measurement state.

This function makes sure, that a new measurement only starts after all sensor modules have completed their last measurement.

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

Parameters:

<state>	*RST:	OFF

7.9 Using the Status Register

Further information:

- Chapter A.2, "Status Reporting System", on page 116

Contents:

7.9.1 General Status Register Commands

Remote commands:

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 module specific errors. Negative error numbers are error messages defined by SCPI.

If the error queue is empty, the querry returns 0 ("No error").

Usage: Query only

7.9.2 Reading Out the CONDition Part

For more information on the CONDition part see Chapter A.2.2, "Structure of an SCPI Status Register", on page 118.

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?

These commands read out the `CONDition` section of the status register.

The commands do not delete the contents of the `CONDition` section.

Usage: Query only

7.9.3 Reading Out the EVENt Part

For more information on the EVENt part see Chapter A.2.2, "Structure of an SCPI Status Register", on page 118.

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[:EVENt]?
STATus:QUEStionable:CALibration[:SUMMary][:EVENt]?
STATus:QUEStionable:POWer[:SUMMary][:EVENt]?

These commands read out the `EVENt` section of the status register.

At the same time, the commands delete the contents of the `EVENt` section.

Usage: Query only

7.9.4 Controlling the ENABle Part

For more information on the ENABLe part see Chapter A.2.2, "Structure of an SCPI Status Register", on page 118.

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>

These commands control the `ENABle` part of a register.

The `ENABle` part allows true conditions in the `EVENt` part of the status register to be reported in the summary bit. If a bit is 1 in the enable register and its associated event bit transitions to true, a positive transition will occur in the summary bit reported to the next higher level.

Parameters:

<value>	*RST:	0

7.9.5 Controlling the Negative Transition Part

For more information on the negative transition part see Chapter A.2.2, "Structure of an SCPI Status Register", on page 118.

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>

These commands control the Negative TRansition part of a register.

Setting a bit causes a 1 to 0 transition in the corresponding bit of the associated register. The transition also writes a 1 into the associated bit of the corresponding `EVENt` register.

Parameters:

<value>	*RST:	0

7.9.6 Controlling the Positive Transition Part

For more information on the positive transition part see Chapter A.2.2, "Structure of an SCPI Status Register", on page 118.

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>

These commands control the Positive TRansition part of a register.

Setting a bit causes a 0 to 1 transition in the corresponding bit of the associated register. The transition also writes a 1 into the associated bit of the corresponding `EVENt` register.

Parameters:

<value>	*RST:	65535

7.10 Testing the R&S NRPM OTA Power Measurement Solution

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

TEST:SENSor? [<Item>]

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

Note: Do not apply a signal while the selftest is running because this can cause erroneously output error messages for the test steps *Offset Voltages* and/or *Noise Voltages*.

Query parameters:

<Item>	String

Usage: Query only

7.11 Calibrating/Zeroing the R&S NRPM3 Sensor Module

Zeroing removes offset voltages from the analog circuitry of the sensor modules, 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.

We recommend that you zero in regular intrervals (at least once a day), if :

- The temperature has varied by more than 5 K.
- The sensor module 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<1...4>:DATA <caldata>

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

Suffix:

<1...4>	1...4

Parameters:

<caldata>	<block_data>

CALibration<1...4>: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.

Suffix:

<1...4>	1...4

Usage: Query only

CALibration<Channel>:ZERO:AUTO <state>

Performs zeroing using the signal at the sensor module input.

Disconnect the sensor module from all power sources, since any signal at the RF input of the sensor module affects the calibration.

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

The setting command accepts only the parameter ONCE; OFF and ON are ignored.

The query returns the value ON if a calibration is in progress, otherwise the value OFF.

Suffix:

<Channel> 1...4
The channel suffix is ignored.

Parameters:

<state> *RST: OFF

8 Programming Examples

This section provides programming examples for R&S NRPM OTA power measurement tasks. It includes examples for continuous average and trace power measurements in the common SCPI syntax.

In addition, Rohde & Schwarz provides archives with source code, project and auxiliary files, and programming examples under C/C++ and Python. The samples can be intregrated into customer-specific applications.

The programming examples are part of the R&S NRP Toolkit, see Chapter 3.4.3, "Optional Software", on page 20 and e.g. "To install the R&S NRP Toolkit on MS Windows" on page 22.

8.1 Performing a Simple Measurement

The simplest way to obtain a result is to use the following sequence of SCPI commands:

Example:

```
*RST          // sets the continuous average mode and all parameters to default
INITiate      // initiates the measurement
FETCh?        // delivers measurement result to the output queue
```

When the measurement is complete, you can retrieve the result from the output queue.

8.2 Performing Measurements in Continuous Average Mode

This section describes programming examples for measuring continuous average power with one R&S NRPM. The examples demonstrate the sequences in SCPI syntax, pseudo code, and also show the corresponding source codes for using VISA protocol.

Example: SCPI sequence for measuring the power of three antenna modules

The command sequence measures the continuous average power of the three channels of an R&S NRPM.

```
// Query the resource identifier and reset the sensor module to default
*IDN?
// Response: ROHDE&SCHWARZ,NRPM3,100001,16.09.20.01
*RST
// Enable all channels of the sensor module
SENSe:CHAnnel1:ENABle 1
SENSe:CHAnnel2:ENABle 1
SENSe:CHAnnel3:ENABle 1
// Select measurement mode and set the time interval to 10 µs
SENSe:FUNCtion "POWer:AVG"
SENSe:POWer:AVG:APERture 10e-6
//Set the averaging filter length to 1
SENSe:AVERage:COUNt 1
// Select the trigger source
TRIGger:SOURce IMMediate
// Activate the continuous measurement
INITiate:CONTinuous 1
// Query the measurement results
FETCh:ALL?
// Response:
// 6.118910E-10,1.185138E-10,4.612524E-10
// Read out all errors / Clear error queue
SYSTem:ERRor?
SYSTem:SERRor?
// Stop the continuous measurement.
INITiate:CONTinuous OFF
```

Example: Pseudo code for measuring the power of two antenna modules

This example, written in pseudo code, shows a reduced set of basic steps to set up and execute a continuous average measurement.

```
// resource = "NRPM3-100001"
// Open the device
sensor = Open( resource )
// Query the resource identifier and reset the sensor module to default
Query( sensor, "*IDN?" )
Write( sensor, "*RST" )
// Measure only on channels 1 & 3
Write( sensor, "SENS:CHAN1:ENAB ON" )
Write( sensor, "SENS:CHAN2:ENAB OFF" )
Write( sensor, "SENS:CHAN3:ENAB ON" )
Write( sensor, "SENS:FUNC \"POW:AVG\"" )
Write( sensor, "TRIG:SOUR IMM" )
Query( sensor, "SYST:ERR?" )
Query( sensor, "SYST:SERR?" )
Write( sensor, "INIT:IMM" )
Query( sensor, "FETCH:ALL?" )
Close( sensor )
```

Example: C/C++ (VISA) source code

This sequence shows the VISA source code, based on the *pseudo code* example above.

```
// Determine the resource identifier, e.g.
// resource = "NRPM3-100001"
// resource identifier = USB0::0x0aad::0x0195::100001
// Open the device
viOpen( defaultRM, “USB0::0x0aad::0x0195::100001”, VI_NULL, VI_NULL,&session );
// Select the measurement mode
viWrite( session, “SENS:FUNC \”POW:AVG\”\n”, 20, &uiSent );
// Initiate a single measurement
viWrite( session, “INIT:IMM\n”, 9, &uiSent );
// Read the measurement result
viWrite( session, “FETCH:ALL?\n”, 7, &uiSent );
viRead( session, buffer, sizeof(buffer), &uiCnt );
// buffer = “1.468872E-09,3.682521E-10,3.829144E-09”
// Close the device
viClose( session );
```

Example: Python (VISA) source code

This sequence shows the Python source code, based on the *pseudo code* example above.

```
####################################################################
## Name:        contAv-M3.py
## Purpose:     Measuring RF power with the R&S NRPM3 OTA sensor module
## Description: This example demonstrates the use of an R&S NRPM3
##              sensor module measuring continuously average power
##              on up to 3 channels
## Author:      Juergen D. Geltinger
## Created:     2016-01-14
## Modified by: Juergen D. Geltinger
## Modified:    2016-07-21
## Copyright:   (c) Rohde & Schwarz, Munich
####################################################################

# This is a Visa sample, so import that module
import visa
from time import sleep
from math import fabs, log10

####################################################################
# Find a R&S NRPM3 power sensor modules. That sensor module is
# characterized by a Product ID of 0x0195
####################################################################
def OpenFirstNRPM3():
   rm = visa.ResourceManager()
   resources=list(rm.list_resources())
   for s in resources:
      # NRPM3 RF sensor module has a USB Product ID of '0x0195'
      if -1 != s.find("0195::"):
         print
         print "Opening NRPM3 sensor '" + s + "'..."
         sensor = rm.open_resource( s )
         if sensor != None:
            sensor.timeout = 20000
            # Setting Aperture Time
            sensor.write( "sens:pow:avg:aper 10e-6" )
            # Setting Average Filter Length
            sensor.write( "sens:aver:count 16" )
            print "Querying *IDN?..."
            print sensor.ask( "*idn?" )
            # Enable the available/connected antenna modules
            result = ""
            for antenna in range(3):
               cmd = "SENS:CHAN{}:PRES?".format( antenna + 1 )
               res = sensor.ask(cmd)
               if int(res) == 1:
                  cmd = "SENS:CHAN{}:ENAB ON".format( antenna + 1 )
```

```
                else:
                    cmd = "SENS:CHAN{}:ENAB OFF".format( antenna + 1 )
                sensor.write(cmd)
            print "SYST:ERR?  --> " + sensor.ask("SYST:ERR?")
            print "SYST:SERR? --> " + sensor.ask("SYST:SERR?")
            return sensor
            break
    return None

##################################################################
# Convert a power value of Watt unit to dBm unit
##################################################################
def Watt2dBm( dW ):
    if fabs( dW ) < 1.0e-19:
        return -160.0
    return 10.0 * log10( fabs( dW ) ) + 30.0

##################################################################
##################################################################
##################################################################
sensor = None
sensor = OpenFirstNRPM3()
if sensor != None:
    for meas in range(1,10):
        sensor.write("init:imm")
        result = sensor.ask("fetch:all?")
        antenna = [float(x) for x in result.split(",")]
        if antenna[0] < 9.9e37:
            print ('Antenna A = %4.1f dBm' % Watt2dBm( antenna[0] ) )
        if antenna[1] < 9.9e37:
            print ('Antenna B = %4.1f dBm' % Watt2dBm( antenna[1] ) )
        if antenna[2] < 9.9e37:
            print ('Antenna C = %4.1f dBm' % Watt2dBm( antenna[2] ) )
        print
    sensor.close()
else:
    print "No NRPM3 sensor found"
```

8.3 Performing Measurements in Trace Mode

This section describes programming examples for measuring power over time with the R&S NRPM OTA Power Measurement Solution. The examples demonstrate the sequences in SCPI syntax, pseudo code, and also show the source codes for using VISA protocol.

Example: SCPI sequence for measuring the power of three antenna modules

The command sequence measures the envelope power over time in the three channels of the R&S NRPM antenna module.

```
// Query the resource identifier and
// reset the sensor module to default
*IDN?
// Response: ROHDE&SCHWARZ,NRPM3,100001,16.09.20.01
*RST
// Enable all channels of the sensor module
SENSe:CHAnnel1:ENABle 1
SENSe:CHAnnel2:ENABle 1
SENSe:CHAnnel3:ENABle 1
// Set the trace measurement mode
SENSe:FUNCtion "XTIME:POWer"
// 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 10e-3
// 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 10
// Enable and configure the averaging filter
SENSe:TRACe:AVERage:COUNt 8
SENSe:TRACe:AVERage:STATe ON
// Configure the trigger
TRIGger:SOURce INTernal
TRIGger:SLOPe POSitive
TRIGger:DTIMe 0.001
TRIGger:HYSTeresis 0.1
TRIGger:LEVel 100e-9
// Activate the measurement
INITiate
// Query the measurement results
FETCh:ALL?
// Response (10 points per channel):
// 1.247340E-07,9.701300E-11,1.249202E-07,1.255408E-10,1.248314E-07,
// 9.718711E-11,1.248812E-07,6.739847E-11,1.248604E-07,2.118106E-10,
// 1.644239E-09,4.748356E-11,1.676801E-09,6.181332E-12,1.670082E-09,
// -1.933849E-11,1.719137E-09,-4.777736E-12,1.678197E-09,3.814278E-12,
// -4.753650E-11,-7.962022E-12,-4.782197E-11,-3.001128E-11,-8.387346E-12,
// -5.910127E-11,5.714555E-11,-5.767460E-12,7.732340E-11,-1.816896E-11
// Read out all errors / Clear error queue
SYSTem:ERRor?
SYSTem:SERRor?
// Stop the continuous measurement.
INITiate:CONTinuous OFF
```

Example: Pseudo code for measuring the power of two antenna modules

This example, written in pseudo code, shows how to set up and execute a trace power measurement.

```
// resource = "NRPM3-100001"
// Open the device
sensor = Open( resource )
// Query the resource identifier and reset the sensor module to default
Query( sensor, "*IDN?" )
Write( sensor, "*RST" )
// Measure only on channels 1 & 3
Write( sensor, "SENS:CHAN1:ENAB ON" )
Write( sensor, "SENS:CHAN2:ENAB OFF" )
Write( sensor, "SENS:CHAN3:ENAB ON" )
Write( sensor, "SENS:FUNC \"XTIM:POW\"" )
Write( sensor, "TRIG:SOUR IMM" )
Query( sensor, "SYST:ERR?" )
Query( sensor, "SYST:SERR?" )
Write( sensor, "INIT:IMM" )
Query( sensor, "FETCH:ALL?" )
Close( sensor )
```

Example: C/C++ (VISA) source code

This sequence shows the VISA source code, based on the *pseudo code* example above.

```
// Determine the resource identifier, e.g.
// resource = "NRPM3-100001"
// resource identifier = USB0::0x0aad::0x0195::100001
// Open the device
viOpen( defaultRM, "USB0::0x0aad::0x0195::100001", VI_NULL, VI_NULL,&session );
// Select the trace measurement mode
viWrite( session, "SENS:FUNC \"XTIM:POW\"\n", 21, &uiSent );
// Set the overall length (10 ms) of the trace
// select the number of measurment points (500 points)
viWrite( session, "SENS:TRAC:TIME 10.0e-3\n", 23, &uiSent );
viWrite( session, "SENS:TRAC:POINTS 500\n", 21, &uiSent );
// Initiate a single measurement
viWrite( session, "INIT:IMM\n", 9, &uiSent );
// Read the measurement result
viWrite( session, "FETCH1?\n", 7, &uiSent );
viRead( session, resA, sizeof(resA), &uiCnt );
viWrite( session, "FETCH2?\n", 7, &uiSent );
viRead( session, resB, sizeof(resB), &uiCnt );
viWrite( session, "FETCH3?\n", 7, &uiSent );
viRead( session, resC, sizeof(resC), &uiCnt );
// process resA, resB and resC. Every buffer contains
// the measurement data of one channel
// Close the device
viClose( session );
```

Annex

A Remote Control Basics

A.1 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 R&S NRPM3 sensor modules support 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.

A.1.1 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.

A.1.2 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 A.1.3, "SCPI Parameters", on page 112.

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**: `INITiate[:IMMediate]` `INIT:IMM` is equivalent to `INIT`
{ }	Parameters in curly brackets are optional and can be inserted once or several times, or omitted.

A.1.3 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. In the case of physical quantities, the unit can be entered. Allowed unit prefixes are G (giga), MA (mega), MOHM and MHZ are also allowed), K (kilo), M (milli), U (micro) and N (nano). If the unit is missing, the basic unit is used.

Units

Only basic units are allowed and recognized.

Special numeric values

The texts listed below are interpreted as special numeric values. In the case of 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 a 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. In the case of a query, the short form of the text is provided.

Example:

Setting command: `TRIGger:SLOPe POSitive`

Query: `TRIG:SLOP?`

Response: `POS`

Character strings

Strings must always be entered 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.

A.1.4 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. • Binary: #B10110 • Octal: #O7612 • Hexa: #HF3A7 • Block: #21312
	A "white space" (ASCII-Code 0 to 9, 11 to 32 decimal, e.g. blank) separates the header from the parameters.

A.1.5 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`

A.1.6 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`

A.2 Status Reporting System

The status reporting system stores all information on the current operating state of the sensor module, 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.

A.2.1 Hierarchy of the Status Registers

Fig.A-1 shows the hierarchical structure of information in the status registers.

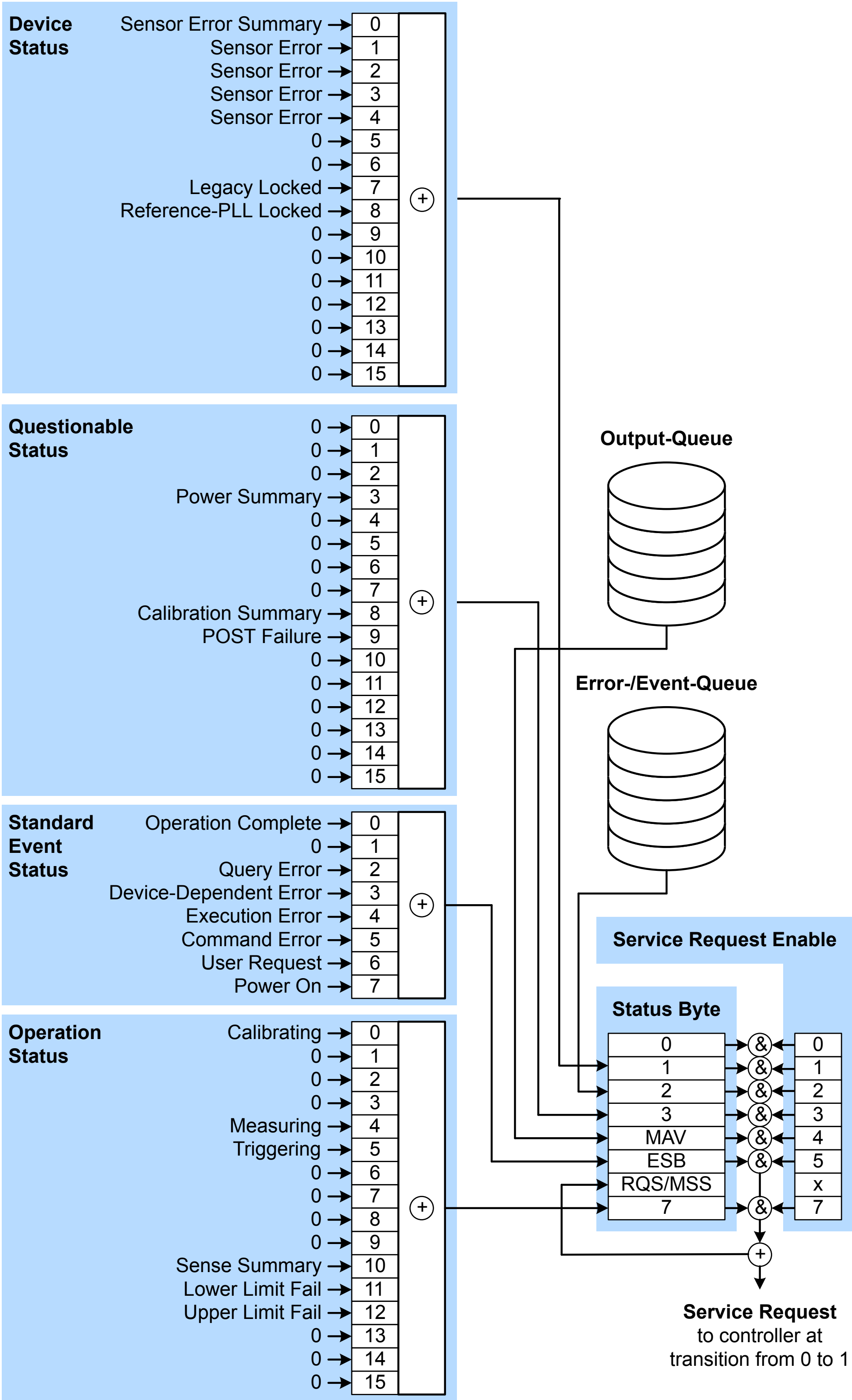


Figure A-1: Graphical overview of the status registers hierarchy

The highest level is formed by the status byte register (STB) and the associated service request enable (SRE) register.

The STB receives its information from the standard event status register (ESR) and the associated Standard Event Status Enable (ESE) Register, as well as from the SCPI-defined operation status register, and the questionable status register, which contain detailed information on the device, and from the device status register.

A.2.2 Structure of an SCPI Status Register

Each SCPI register consists of five 16-bit registers which have different functions (see Figure A-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 zero in all registers. This prevents problems some controllers have with the processing of unsigned integers.

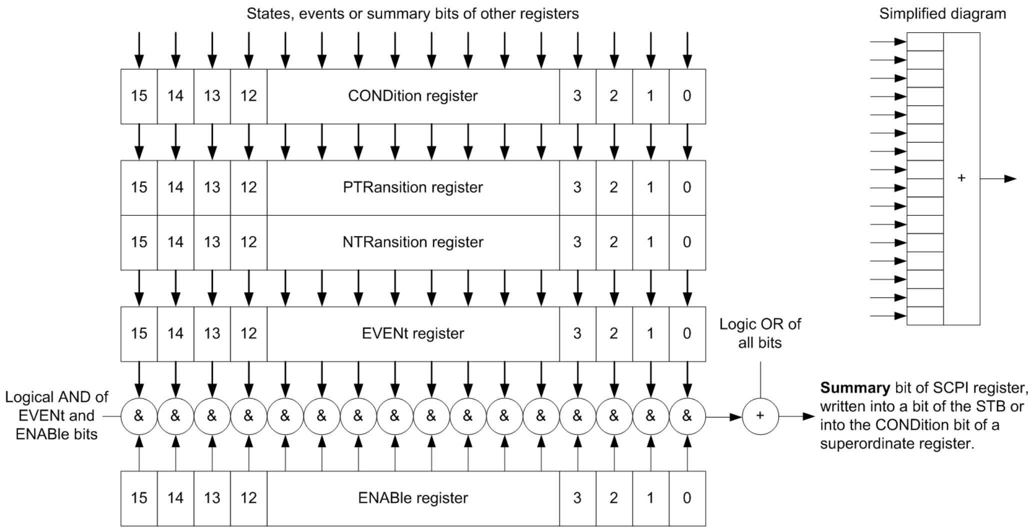


Figure A-2: Standard SCPI status register

CONDition status register part

The five parts of an 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.

This part can only be read by the user. 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.

This part can be written into and read by the user 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.

A.2.3 Status Byte (STB) and Service Request Enable Register (SRE)

The STB is already defined in IEEE 488.2. It gives a rough overview of the sensor module status, collecting information from the lower-level registers. It is comparable with the CONDition register of an 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 is read by the query *STB? or a serial poll. The SRE is associated with the STB. The function of the SRE corresponds to that of the ENABle register of the SCPI registers. Each bit of the STB is assigned a bit in the SRE. Bit 6 of the SRE is ignored. If a bit is set in the SRE and the associated bit in the STB changes from 0 to 1, a service request (SRQ) is generated on the IEC/IEEE bus, which triggers an interrupt in the controller configured for this purpose, and can be further processed by the controller.

The SRE can be set by the command `*SRE` and read by the query `*SRE?`.

Table A-1: Meaning of bits used in the status byte

Bit No.	Meaning
0	Not used
1	**Device status register summary bit** Depending on the configuration of the sensor modules status register, this bit is set when a sensor module is connected or disconnected or when an error has occurred in a sensor module. See Chapter A.2.5, "Device Status Register", on page 121.
2	**Error queue not empty** The bit is set if the error queue has an entry. If this bit is enabled by the SRE, 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 bit** This bit is set if an `EVENt` bit is set in the `QUEStionable` status register and the associated `ENABLe` 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. See Chapter A.2.6, "Questionable Status Register", on page 122.
4	**MAV bit** (Message available) This bit is set if a readable message is in the output queue. This bit may be used to automate reading of data from the sensor module into the controller.
5	**ESB: Standard event status register summary bit** This bit is set if 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. See Chapter A.2.7, "Standard Event Status and Enable Register (ESR, ESE)", on page 124.

Bit No.	Meaning
6	**MSS: Master status summary bit** This bit is set if the sensor module triggers a service request. This is the case 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 bit** This bit is set if an `EVENt` bit is set in the operation status register and the associated `ENABLe` bit is set to 1. A set bit denotes that an action is being performed by the sensor module. Information on the type of action can be obtained by querying the operation status register. See Chapter A.2.8, "Operation Status Register", on page 125.

A.2.4 IST Flag and Parallel Poll Enable Register (PPE)

Similar to the SRQ, the IST flag combines the complete status information in a single bit. It can be queried by a parallel poll or by the `*IST?` command.

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 SRE. The IST flag is obtained by `ORing` all results together.

The PPE can be set by the `*PRE` command and read by the `*PRE?` query.

A.2.5 Device Status Register

Contains information on the state of the static errors.

Querying the register:

- `STATus:DEVice:CONDition?`
- `STATus:DEVice[:EVENt]?`

Querying the static errors:

- `SYSTem:SERRor?`

Table A-2: Meaning of bits used in the device status register

Bit No.	Meaning
0	**Sum of SERR bits** The 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** A static error exists therefore the Status LED of the sensor module is blinking slowly in red.

Bit No.	Meaning
4	**SERR critical** A critical static error exists therefore the Status LED of the sensor module is blinking fast in red.
5 to 6	Not used
8	**Reference PLL locked state** This bit signals whether the PLL for the clock reference is synchronized. The bit is useful when selecting an external clock source. The following states are possible: ● Internal clock (`[SENSe<Sensor>:]ROSCillator:SOURce INTernal INT`): – `1` (always) ● External clock (`[SENSe<Sensor>:]ROSCillator:SOURce EXTernal`): – `1` if the sensor module was able to synchronize with external clock – `0` if the sensor module could not synchronize with external clock
9 to 15	Not used

A.2.6 Questionable Status Register

Contains information on questionable sensor module states. Such states may occur when the sensor module is not operated in compliance with its specifications.

Querying the register:

- `STATus:QUEStionable:CONDition?`
- `STATus:QUEStionable[:EVENt]?`

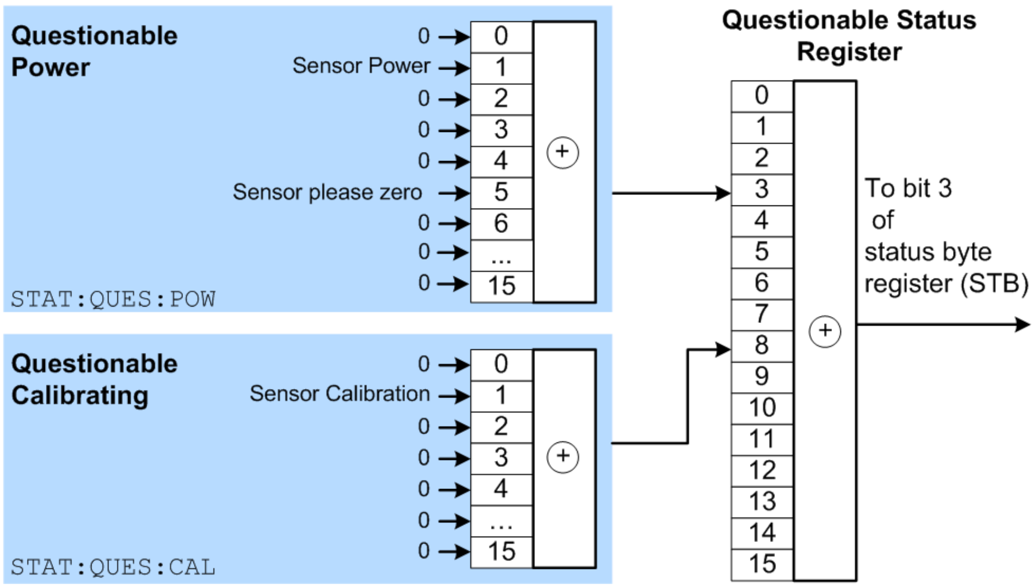


Table A-3: Meaning of bits used in the questionable status register

Bit No.	Meaning
0 to 2	Not used
3	**Questionable power status register summary bit** Corresponds to the summary bit of the questionable power status register. See Chapter A.2.6.1, "Questionable Power Status Register", on page 123.
4 to 7	Not used
8	**Questionable calibration status register summary bit** Corresponds to the summary bit of the questionable calibration status register. See Chapter A.2.6.2, "Questionable Calibration Status Register", on page 123.
9	**POST failure** The built-in test of the R&S NRPM carried out automatically upon power-up has generated an error.
10 to 15	Not used

A.2.6.1 Questionable Power Status Register

The `CONDition` register contains information whether the measured power values are questionable.

Querying the register:

- `STATus:QUEStionable:POWer:CONDition?`
- `STATus:QUEStionable:POWer[:SUMMary][:EVENt]?`

Table A-4: Meaning of bits used in the questionable power status register

Bit No.	Meaning
0	Not used
1	**sensor module power** The measurement data of the sensor module is corrupt.
2 to 4	Not used
5	**sensor module please zero** The zero correction for the sensor module is no longer correct and should be repeated.
6 to 15	Not used

A.2.6.2 Questionable Calibration Status Register

The `EVENt` register and the `CONDition` register contain information whether the zero offset of a sensor module is still valid.

Querying the register:

- `STATus:QUEStionable:CALibration:CONDition?`
- `STATus:QUEStionable:CALibration[:SUMMary][:EVENt]?`

Table A-5: Meaning of bits used in the questionable calibration status register

Bit No.	Meaning
0	Not used
1	**sensor module calibration** Zeroing of the sensor module was not successful.
2 to 15	Not used

A.2.7 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 an SCPI register. The standard event status register can be read out by the query `*ESR`.

The `ESE` forms the associated `ENABle` register. It can be set and read out with the command/query `*ESE`.

Table A-6: Meaning of bits used in the standard event status register

Bit No.	Meaning
0	**Operation complete** When the `*OPC` command is received, this bit is set if all previous commands have been executed.
1	Not used
2	**Query error** This bit is set in either of the two following cases: the controller wants to read data from the sensor module but has not sent a query, or it sends new commands to the sensor module before it retrieves existing requested data. A frequent cause is a faulty query which cannot be executed.
3	**Device-dependent error** This bit is set if a sensor module 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** This bit is set if 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** This bit is set if 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** This bit is set when the sensor module is switched over to manual control.
7	**Power on** This bit is set when the sensor module is switched on.

A.2.8 Operation Status Register

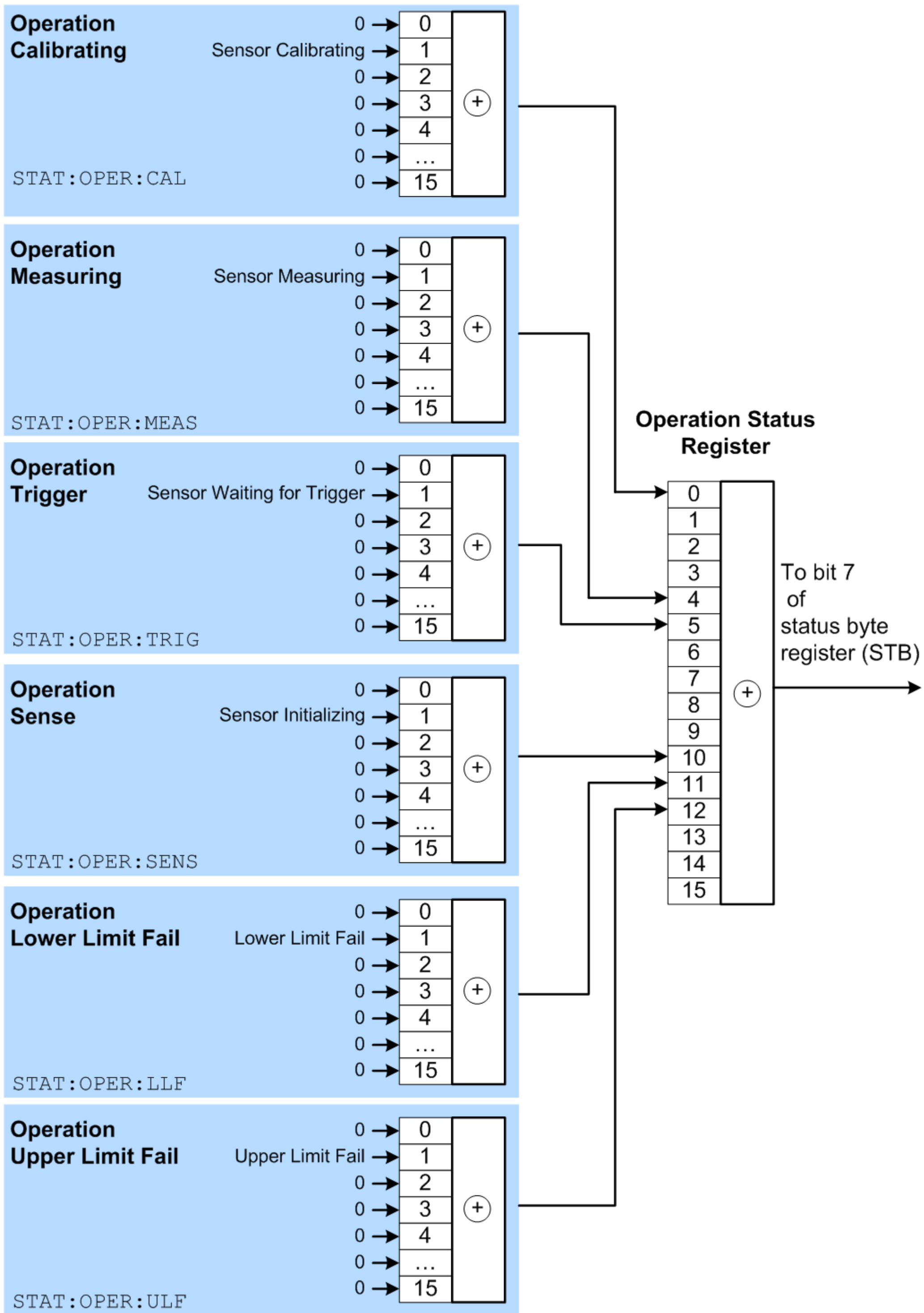
The CONDition register contains information on the operations currently being performed by the sensor module, while the EVENt register contains information on the operations performed by the since the last readout of the register.

Querying the register:

- STATus:OPERation:CONDition?
- STATus:OPERation[:EVENt]?

Table A-7: Meaning of bits used in the operation status register

Bit No.	Meaning
0	**Calibrating status register summary bit** This bit is set if the sensor module is being calibrated. See Chapter A.2.8.1, "Operation Calibrating Status Register", on page 126.
1 to 3	Not used
4	**Measuring status register summary bit** This bit is set if the sensor module is performing a measurement. See Chapter A.2.8.2, "Operation Measuring Status Register", on page 127.
5	**Trigger status register summary bit** This bit is set if the sensor module is in the waiting for trigger state, i.e. waiting for a trigger event. See Chapter A.2.8.3, "Operation Trigger Status Register", on page 127.
6	Not used
7 to 9	Not used
10	**Sense status register summary bit** This bit is set if a sensor module is initialized. See Chapter A.2.8.4, "Operation Sense Status Register", on page 128.
11	**Lower limit fail status register** This bit is set if a displayed value has dropped below a lower limit value. See Chapter A.2.8.5, "Operation Lower Limit Fail Status Register", on page 128.
12	**Upper limit fail status register** This bit is set if a displayed value has exceeded an upper limit value. See Chapter A.2.8.6, "Operation Upper Limit Fail Status Register", on page 129.
13 to 14	Not used
15	Bit 15 will never be used.



A.2.8.1 Operation Calibrating Status Register

The `CONDition` register contains information about whether a sensor module is currently being calibrated and, depending on the configuration of the transition register.

The EVENt register indicates whether a calibration was started or completed since the last readout of this register.

Querying the register:

- STATus:OPERation:CALibrating:CONDition?
- STATus:OPERation:CALibrating[:SUMMary][:EVENt]?

Table A-8: Meaning of bits used in the operation calibrating status register

Bit No.	Meaning
0	Not used
1	**sensor module calibrating** The sensor module is being calibrated
5 to 15	Not used

A.2.8.2 Operation Measuring Status Register

The CONDition register contains information about whether a measurement is being performed by a sensor module and, depending on the configuration of the transition register. The EVENt register indicates whether a measurement was started or completed since the last readout of this register.

Querying the register:

- STATus:OPERation:MEASuring:CONDition?
- STATus:OPERation:MEASuring[:SUMMary][:EVENt]?

Table A-9: Meaning of bits used in the operation measuring status register

Bit No.	Meaning
0	Not used
1	**sensor module measuring** The sensor module is performing a measurement.
5 to 15	Not used

A.2.8.3 Operation Trigger Status Register

The CONDition register contains information about whether a sensor module is currently in the waiting for trigger state, i.e. expecting a trigger event and, depending on the configuration of the transition register. TheEVENt register indicates whether the waiting for trigger state was entered or quit by a sensor module since the last readout of the register.

Querying the register:

- STATus:OPERation:TRIGger:CONDition?
- STATus:OPERation:TRIGger[:SUMMary][:EVENt]?

Table A-10: Meaning of bits used in the operation trigger status register

Bit No.	Meaning
0	Not used
1	**sensor module waiting for trigger** The sensor module is in the waiting for trigger state and is waiting for a trigger event. When the trigger event occurs, the sensor module changes into the measuring state.
5 to 15	Not used

A.2.8.4 Operation Sense Status Register

The `CONDition` register contains information about whether a sensor module is currently being initialized and, depending on the configuration of the transition register. The `EVENt` register indicates whether a sensor module initialization was started or completed since the last readout of this register.

This status is assumed by a sensor module if one of the following conditions is met:

- The supply voltage is switched on (power up)
- The sensor module was just connected
- A reset was performed using:
  - `*RST`
  - `SYSTem:PRESet`

Querying the register:

- `STATus:OPERation:SENSe:CONDition?`
- `STATus:OPERation:SENSe[:SUMMary][:EVENt]?`

Table A-11: Meaning of bits used in the operation sense status register

Bit No.	Meaning
0	Not used
1	**sensor module initializing** The sensor module is being initialized.
5 to 15	Not used

A.2.8.5 Operation Lower Limit Fail Status Register

The `CONDition` register contains information about whether a displayed value is currently below a configured lower limit. The `EVENt` register indicates whether a measured value dropped below a limit value since the last readout of the Operation Lower Limit Fail Status Register. Details of the behavior are defined by the transition register.

Querying the register:

- `STATus:OPERation:LLFail:CONDition?`
- `STATus:OPERation:LLFail[:SUMMary][:EVENt]?`

Table A-12: Meaning of bits used in the operation lower limit fail status register

Bit No.	Meaning
0	Not used
1	**Lower limit fail** The measured value drops below the lower limit value.
5 to 15	Not used

A.2.8.6 Operation Upper Limit Fail Status Register

The `CONDition` register contains information about whether a displayed value is currently above a configured upper limit. The `EVENt` register indicates whether a limit value was exceeded since the last readout of the Operation Upper Limit Fail Status Register.

Querying the register:

- `STATus:OPERation:ULFail:CONDition?`
- `STATus:OPERation:ULFail[:SUMMary][:EVENt]?`

Table A-13: Meaning of bits used in the operation lower limit fail status register

Bit No.	Meaning
0	Not used
1	**Upper limit fail** The measured value exceeds the upper limit value.
5 to 15	Not used

Glossary: List of the often Used Terms and Abbreviations

A

anechoic chamber: Non-reflective, non-echoing box (RF test box) or chamber (RF test chamber) that absorbs reflections of radio frequencies, used to test antennas, radars or electromagnetic interference.

API: Application Programming Interface

Average count: Number of averaged readings

Average factor: see Average count

B

Beamforming: Steering a beam in the direction of the receiver.

C

Computer name: An unambiguous indication of the instrument in a LAN that uses a DNS server.
The default computer name follows the syntax `NRPM3-<serial number>`, e.g. *NRPM3-100099*.
Synonym: Hostname
See Serial number.

D

DUT: Device Under Test

G

Glossary: List of the often used terms and abbreviations

GUI: Graphical User Interface

H

Hostname: Computer name

O

OS: Operating System

OTA: Over The Air

P

PC: Personal Computer, desktop or laptop PC

Power: A term describing the signal level in the RF domain or defining the length of the I/Q vector in the Baseband domain.

product page: A designation of the R&S NRPM product page.

R

R&S NRP-Toolkit: Software package, which provides drivers for Linux, Mac OS X or MS Windows.

Remote control: The operation of the R&S NRPM by remote control commands or programs to perform automated tests. The instrument is connected to a system controller via LAN/VXI-11, GPIB or USB using Virtual Instrument Software Architecture (VISA). The instrument is controlled either directly, or supported by instrument drivers.

RF: Radio Frequency

RF test box: Anechoic test box with very low reflectivity. Provides a shielded test environment for OTA measurements of radio frequencies.

RF test chamber: Larger anechoic RF test chamber up to RF test room sizes with very low reflectivity. Provides a shielded test environment for OTA measurements of radio frequencies.

rsu: Rohde & Schwarz Update. A file format which contains updated device software. Processed by a firmware update program.

S

Shielded box: See anechoic chamber.

T

Trigger: Internally generated or externally supplied signal which starts signal generation at a particular point in time.

Trigger event: A trigger event is caused by the received trigger signal or executed manual trigger.

U

USB: Universal Serial Bus. A standard type of connection for many different kinds of devices.

USBTMC: (USB Test & Measurement Class)
A protocol built on top of USB for communication with USB devices. Using VISA library, it supports service request, triggers, and other specific operations, similar to GPIB.

V

VISA: Virtual Instrument Software Architecture

A standardized software interface library providing input and output functions to communicate with instruments.

W

WPTC: (Wireless Performance Test Chamber)
Anechoic chamber with very low reflectivity. Provides a shielded test environment for OTA measurements of radio frequencies.

List of Commands

Index

Q

R

S

T

U

V

W