

MATLAB[®] Toolkit for R&S[®] NRP-Z Sensors Application Note

Products:

- | R&S[®] NRP-Z Sensors
- | R&S[®] NRP Power Meter
- | R&S[®] NRP2 Power Meter

The R&S MATLAB[®] Toolkit for R&S[®] NRP-Z Sensors provides routines for remote controlling these instruments via USB or the R&S[®] NRP Power Meter. This application note describes how to access the sensors using the R&S[®] VXI PnP driver. It also contains a set of scripts that perform commonly needed tasks, such as opening a sensor connection or performing measurements.

Table of Contents

1	Notices.....	3
2	Overview.....	3
3	System Requirements.....	4
3.1	Hardware.....	4
3.2	Software.....	4
4	Release Notes.....	4
5	Installation.....	5
5.1	Unpacking The Files.....	5
5.2	Setting The Path In MATLAB.....	5
6	USB Device IDs.....	6
7	Using the VXI PnP Driver.....	7
8	Toolkit Scripts.....	8
8.1	rs_nrpz_load_vxipnp.....	8
8.2	rs_nrpz_init.....	9
8.3	rs_nrpz_close.....	10
8.4	rs_nrpz_get_error.....	11
8.5	rs_nrpz_get_sensorname.....	12
8.6	rs_nrpz_setup_contav.....	13
8.7	rs_nrpz_read_measurement.....	14
9	Example Scripts.....	15
9.1	NRPZ_VXIPnP_ContAv.....	15
9.2	NRPZ_VXIPnP_Trace.....	15
9.3	NRPZ_Instrument_Sweep.m.....	16
9.4	NRP_BaseUnit.m.....	16

1 Notices

The following abbreviations are used throughout the application note.

- The MATLAB® Toolkit for R&S® NRP-Z Sensors is referred to as MATLAB Toolkit or toolkit.

Microsoft®, Windows®, MS Windows®, Windows NT®, and MS-DOS® are U.S. registered trademarks of Microsoft Corporation.

MATLAB® is a U.S. registered trademark of The Math Works, Inc.

Rohde & Schwarz® is a registered trademark of Rohde & Schwarz GmbH & Co. KG.

2 Overview

The R&S MATLAB® Toolkit for R&S® NRP-Z Sensors provides routines for remote controlling these instruments via USB or the R&S® NRP Power Meter. This application note describes how to access the sensors using the R&S® VXI PnP driver. It also contains a set of scripts that perform commonly needed tasks, such as opening a sensor connection or performing measurements.

The scripts provided with this toolkit are intended for use on a Microsoft Windows based system.

3 System Requirements

3.1 Hardware

- The MATLAB requirements (CPU, memory, hard drive)
- Pertaining to the NRP Power Meter remote control:
 - 100 MBit LAN -or-
 - GPIB interface -or-
 - USB 1.1 or USB 2.0 interface
- Pertaining to the NRP-Z Sensors:
 - USB 1.1 or USB 2.0 interface
 - R&S NRP-Z3, R&S NRP-Z4, R&S NRP-Z5 adapter

3.2 Software

This toolkit is intended for use on a Microsoft Windows operating system. The following configuration was tested by Rohde & Schwarz by the time of writing.

- Microsoft Windows
 - Windows 7 x64
 - MATLAB R2010a (64 bit)
 - MATLAB Instrument Control Toolbox
 - National Instruments VISA Version 4.6.2
 - R&S NRP Toolkit 2.1.10
 - rsnrpz VXI PnP driver 2.12.0

4 Release Notes

V 1.0 - Initial release

5 Installation

5.1 Unpacking The Files

This toolkit comes as a set of files bundled in a ZIP archive.

`'Rs_Nrpz_Matlab_Toolkit_<version number>.zip`

First, create a new folder under your MATLAB toolbox directory. This directory is a sub directory of your MATLAB installation path.

`.\MATLAB\R2010a\toolbox\RsNrpzMatlabToolkit`

The contents of the archive should now be unpacked into this directory.

5.2 Setting The Path In MATLAB

For a convenient use of the toolkit functions it is required to add the installation path to the MATLAB path list. This can be done from within MATLAB by selecting *File* → *Set Path...* from the menu bar. This opens the 'Set Path' dialog where the new path is added using the *Add Folder...* button.

6 USB Device IDs

The R&S NRP-Z Sensors are complete instruments that feature the detector, as well as the analog and digital signal processing in one housing. As a result the sensors provide an industry standard USB interface that can be connected to the NRP Power Meter or to a standard PC.

Sensors that are directly connected to a PC are identified by their vendor, the instrument type as well as the serial number. This is required because USB does typically does address specific ports.

The vendor ID for all Rohde & Schwarz devices is 0x0AAD. The product ID depends on the instrument and is listed below.

USB Device IDs	
Instrument	Device ID
NRP-Z11	0x0C
NRP-Z21	0x03
NRP-Z22	0x13
NRP-Z23	0x14
NRP-Z24	0x15
NRP-Z27	0x2F
NRP-Z28	0x51
NRP-Z31	0x2C
NRP-Z37	0x2D
NRP-Z51	0x16
NRP-Z52	0x17
NRP-Z55	0x18
NRP-Z56	0x19
NRP-Z57	0x70
NRP-Z81	0x23
NRP-Z85	0x83
NRP-Z86	0x95
NRP-Z91	0x21
NRP-Z92	0x62
NRP-Z98	0x52

The list represents all R&S NRP-Z Power Sensors that support USB remote control at the time of writing.

7 Using the VXI PnP Driver

This toolkit uses the C function interface of the VXI PnP driver for accessing the NRP-Z Sensors. These sensors are complete instruments that do not require a base unit and feature a USB interface that can directly be connected to the host PC. For performance reasons the protocol between the sensors and the PC is binary. On the PC side the instrument drivers convert SCPI commands or function calls into the binary protocol of the sensors.

The VXI PnP driver comes as a Microsoft Windows DLL which is referred to as a shared library in MATLAB. The MATLAB `callib` function provides easy access to the routines contained in the DLL file.

The following code lines demonstrate how routines from the VXI PnP driver DLL can be accessed via MATLAB.

Initially the DLL must be loaded into memory. This step is required once before any driver function can be used. The code line below assumes that the name of the driver DLL is `rsnrpz_32.dll` and a header file `rsnrpz.h` exists.

```
loadlibrary( 'rsnrpz_32', 'rsnrpz.h' );
```

The next example opens a sensor connection. The `rsnrpz_init()` function in the VXI PnP driver returns a session number and requires a string identifying the sensor that is to be opened. Before calling the function these two variables must be defined as pointers of the correct data type.

```
pNRPSession = libpointer( 'uint32Ptr', 0 );
sensorname = '*';
pSensorName = libpointer( 'uint8Ptr', [uint8( sensorname ) 0] );
```

The session pointer is defined as a pointer to variable of the type unsigned integer (32 bit). The sensor name is a pointer to a string (unsigned integer 8 bit) containing the name '*'. Using the * character instead of a full resource identifier string lets the driver function open the first available sensor.

```
[err, p8, session] = calllib( 'rsnrpz_32', 'rsnrpz_init', ...
                             pSensorName, 1, 1, pNRPSession );
```

The `callib` function accesses the `rsnrpz_init()` function of the VXI PnP driver. The first argument is the pointer to the sensor name. The next two parameters are boolean flags that enable resetting the sensor and querying its ID. The last parameter is the pointer to the session number that is returned by the function.

MATLAB requires that all return variables are listed in the same order as they appear as arguments on the left side of the `callib` function call. This is the error return code of `rsnrpz_init()`, as well as the pointer to the name and the session number.

8 Toolkit Scripts

This toolkit contains scripts that simplify access to the VXI PnP driver functions. These scripts demonstrate the use of the VXI PnP driver and can also be used as a starting point for own implementations. Using these scripts also avoids the need to use the `callib` functions directly.

The following overview lists all scripts that are part of this toolkit.

Function Overview	
Function / .m File	Description
<code>rs_nrpz_load_vxipnp</code>	Loads the VXI PnP driver initially
<code>rs_nrpz_init</code>	Open a sensor connection
<code>rs_nrpz_close</code>	Close a sensor connection
<code>rs_nrpz_get_error</code>	Get error text for a VXI PnP error code
<code>rs_nrpz_get_sensorname</code>	Read name information from sensor
<code>rs_nrpz_setup_contav</code>	Setup a continuous average measurement
<code>rs_nrpz_read_measurement</code>	Initiate a measurement and wait for result

8.1 `rs_nrpz_load_vxipnp`

This script loads the VXI PnP driver DLL into memory. The actual driver DLL depends on the system architecture and is automatically detected. The script returns the DLL name for use with subsequent functions.

```
[ <Status, <LibName> ] = rs_nrpz_load_vxipnp()
```

Return values

<code><Status></code>	1 if successful
<code><LibName></code>	The library name

Example code

```
[status, libname] = rs_nrpz_load_vxipnp();
if( status==0 ); ... end;
```


8.2 rs_nrpz_init

This script opens a sensor connection and initializes the sensor to a default state. The return parameter is the session number that is required for further sensor communication.

```
[ <Status>, <Session> ] = rs_nrpz_init( <LibName>,  
                                       '<SensorName>' );
```

Parameters

<LibName>	The library name returned by rs_nrpz_load_vxipnp()
<SensorName>	The sensor resource string contains the vendor and product ID, as well as the device serial number. 'USB::0x0AAD::<product ID>::<serial_no>'

Return value

<Status>	1 if successful
<Session>	The sensor session.

Example code

```
[status, session] = rs_nrpz_init( libname, '*' );  
if( status==0 ); ... end;  
  
[status, session] = rs_nrpz_init( libname,  
                                 'USB::0x0AAD::0x23::100123' );  
if( status==0 ); ... end;
```

8.3 rs_nrpz_close

This script closes a sensor connection. After the sensor is closed the session number cannot be used for further communication anymore.

```
<Status> = rs_nrpz_close( <LibName>, <Session> );
```

Parameters

<LibName> The library name returned by `rs_nrpz_load_vxipnp()`
<Session> The session number returned by `rs_nrpz_init()`

Return values

<Status> 1 if successful

Example code

```
status = rs_nrpz_close( libname, session );
```

8.4 rs_nrpz_get_error

This script returns the textual error message for a VXI PnP driver error code.

```
[<Status>, <ErrorText>] = rs_nrpz_get_error( <LibName>,  
                                           <Session>, <errorcode> );
```

Parameters

<LibName>	The library name returned by <code>rs_nrpz_load_vxipnp()</code>
<Session>	The session number returned by <code>rs_nrpz_init()</code>
<errorcode>	The error code returned by a VXI PnP driver function

Return values

<Status>	1 if successful
<ErrorText>	Descriptive text for the driver error code

Example code

```
[status, text] = rs_nrpz_get_error( libname,  
                                   session, err );  
disp( [ 'Driver Error: ' text ] );
```

8.5 rs_nrpz_get_sensorname

This script reads the sensor name from a previously opened sensor. This functionality is useful if the sensor was opened using '*' as the resource identifier.

```
[<Status>, <Name>] = rs_nrpz_get_sensorname( <LibName>,  
                                             <Session> );
```

Parameters

<LibName>	The library name returned by <code>rs_nrpz_load_vxipnp()</code>
<Session>	The session number returned by <code>rs_nrpz_init()</code>

Return value

<Status>	1 if successful
<Name>	Sensor name, e.g. 'NRP-Z81'

Example code

```
[status, name] = rs_nrpz_get_sensorname( libname,  
                                         session );  
disp( [ 'Sensor: ' name ] );
```

8.6 rs_nrpz_setup_contav

This script configures the sensor for a continuous average power measurement. Major parameters, such as the carrier frequency, level offset, and average filter count are provided as arguments. This script may serve as starting point for custom implementations of continuous power measurements.

```
<Status> = rs_nrpz_setup_contav( <LibName>,  
                                <Session>, <Frequency>,  
                                <LevelOffset>, <AvCount> );
```

Parameters

<LibName>	The library name returned by <code>rs_nrpz_load_vxipnp()</code>
<Session>	The session number returned by <code>rs_nrpz_init()</code>
<Frequency>	The signal frequency in Hz
<LevelOffset>	The level offset in dB
<AvCount>	The average filter count

Return value

<Status>	1 if successful
<Name>	Sensor name, e.g. 'NRP-Z81'

Example code

```
status = rs_nrpz_setup_contav( libname,  
                               session, 1e9, 0, 32 );  
if( status==0 ); ... end;
```

8.7 rs_nrpz_read_measurement

This script initiates a measurement and waits for the numeric result. This script may serve as a starting point for custom implementations.

```
[ <Status>, <Result> ] = rs_nrpz_read_measurement( <LibName>,  
                                                <Session> );
```

Parameters

<LibName>	The library name returned by <code>rs_nrpz_load_vxipnp()</code>
<Session>	The session number returned by <code>rs_nrpz_init()</code>

Return value

<Status>	1 if successful
<Result>	Numeric measurement result

Example code

```
[status, result] = rs_nrpz_read_measurement( libname,  
                                             session );  
  
if( status==0 ); ... end;  
disp( [ 'Result: ' num2str(result) ] );
```

9 Example Scripts

The example scripts provided in this toolkit use the functions described in the previous chapter. Some scripts also require the instrument control functionality provided in the 1GP60 application note.

Script Overview	
Script / .m File	Functionality
NRPZ_VXIPnP_ContAv.m	Open sensor and run a few continuous average power measurements.
NRPZ_VXIPnP_Trace.m	Open sensor and run a trace measurement.
NRPZ_Instrument_Sweep.m	Control a power sensor that is connected to a R&S signal generator. Requires the scripts provided with the 1GP60 application note.
NRP_BaseUnit.m	Control a power sensor via the NRP base unit. Requires the scripts provided with the 1GP60 application note.

9.1 NRPZ_VXIPnP_ContAv

This script demonstrates how to access the NRP-Z Sensor using the R&S VXI PnP instrument driver. The power sensor is directly connected to the PC via USB. The installation of the NRP Toolkit and the VXI PnP instrument driver is required. Both drivers are available free of charge from the Rohde&Schwarz web site www.rohde-schwarz.com.

The example configures the sensor in continuous average power measurement mode. In this mode the sensor continuously performs average power measurements and returns scalar results to the PC.

9.2 NRPZ_VXIPnP_Trace

This script demonstrates how to access the NRP-Z Sensor using the R&S VXI PnP instrument driver. The power sensor is directly connected to the PC via USB. The installation of the NRP Toolkit and the VXI PnP instrument driver is required. Both drivers are available free of charge from the Rohde&Schwarz web site www.rohde-schwarz.com.

The example configures the sensor in the trace measurement mode. This mode is available for all universal and wideband power sensors and captures the RF envelope. The result is an array of scalar readings that can be displayed graphically.

9.3 NRPZ_Instrument_Sweep.m

This script controls a signal generator and a power sensor that is connected to this signal generator. It makes use of the instrument control routines provided with the 1GP60 application note.

The script demonstrates how to implement a frequency sweep with the generator and take simultaneous power readings for each frequency point. This functionality may be useful for characterizing devices, such as passive filters.

9.4 NRP_BaseUnit.m

This simple script performs a series of average power measurements using a sensor that is connected to the NRP Power Meter. The script automatically detects the sensor connected to port A or B.

As a prerequisite for running this example the scripts provided with the 1GP60 application note must be installed.

About Rohde & Schwarz

Rohde & Schwarz is an independent group of companies specializing in electronics. It is a leading supplier of solutions in the fields of test and measurement, broadcasting, radiomonitoring and radiolocation, as well as secure communications. Established more than 75 years ago, Rohde & Schwarz has a global presence and a dedicated service network in over 70 countries. Company headquarters are in Munich, Germany.

Environmental commitment

- Energy-efficient products
- Continuous improvement in environmental sustainability
- ISO 14001-certified environmental management system

Certified Quality System

ISO 9001

Regional contact

Europe, Africa, Middle East

+49 89 4129 123 45

customersupport@rohde-schwarz.com

North America

1-888-TEST-RSA (1-888-837-8772)

customer.support@rsa.rohde-schwarz.com

Latin America

+1-410-910-7988

customersupport.la@rohde-schwarz.com

Asia/Pacific

+65 65 13 04 88

customersupport.asia@rohde-schwarz.com

This application note and the supplied programs may only be used subject to the conditions of use set forth in the download area of the Rohde & Schwarz website.

R&S® is a registered trademark of Rohde & Schwarz GmbH & Co. KG. Trade names are trademarks of the owners.

Rohde & Schwarz GmbH & Co. KG

Mühl Dorfstraße 15 | D - 81671 München

Phone + 49 89 4129 - 0 | Fax + 49 89 4129 - 13777

www.rohde-schwarz.com