Remote Emulation with the R&S®SMW200A Vector Signal Generator

Application Note

Products:

| R&S®SMW200A

The R&S®SMW200A Vector Signal Generator offers a remote emulation feature that makes it possible to control the instrument by commands other than the built-in native SCPI commands. This feature allows the user to replace signal generators, e.g. from other manufacturers, with the R&S®SMW200A without having to change the remote control code.

This application note describes how to use the remote emulation feature in general. Furthermore, it describes in detail the remote emulation for each supported instrument, limitations of the individual emulations and the remaining differences between the emulated and the original commands.

ROHDE & SCHWARZ

Table of Contents

1 Abbreviations

The following abbreviations are used in this application note:

- AF3416   3416 signal generator from Aeroflex / IFR / Marconi
- E8257   E8257 signal generator from Agilent Technologies
- HP83620   83620 signal generator from Hewlett-Packard / Agilent Technologies
- HP83630   83630 signal generator from Hewlett-Packard / Agilent Technologies
- HP83640   83640 signal generator from Hewlett-Packard / Agilent Technologies
- HP83650   83650 signal generator from Hewlett-Packard / Agilent Technologies
- SMW   R&S$^{®}$SMW200A signal generator from Rohde & Schwarz

2 Overview

Measuring instruments used in commercial test systems for applications such as testing of mobile radio base stations typically have a much longer lifespan than the test system itself. Some five to seven years of usage is normal prior to replacement. On the other hand, test systems for use in aerospace & defense applications can have a typical lifespan of 25 years or even more. If standard measuring instruments with a lifespan that is significantly less than this time frame are used in such test systems, an obvious question is how to handle maintenance, repair, calibration and even replacement of the instruments that are no longer supported by their manufacturer. Moreover, the test program sets used in such test systems are generally certified, which makes it very time-consuming and costly to modify and reaccept them.

A feasible strategy to solve this problem is the migration with new measuring instruments that emulate the discontinued instruments.

This migration approach is saves time and is cost-effective. It involves replacing obsolete measuring instruments with new ones that emulate their functionality and programming.

The requirements for the new instruments revolve around the electrical and functional features of the instruments to be emulated as well as the test programs, which ideally should not require any modifications.

To fulfill the last requirement, the SMW offers the remote emulation feature.

Remote emulation means that the SMW will understand the programming commands of the emulated instrument and also emulate the behavior as fully as possible.

Most instruments from other manufacturers in the category of the SMW are emulated by the SMW. An overview of actually implemented remote emulations is given in the following table:

Remote emulations in the SMW			
Manufacturer	**Instrument**	**Language**	**Section**
Aeroflex IFR Marconi	AF3416	SCPI	5
Agilent Technologies Hewlett-Packard	E8257	SCPI	6
	HP83620		7
	HP83630		
	HP83640		
	HP83650		

3 Basics

3.1 Remote Control Languages

Instruments implement different kind of remote control languages. These languages are grouped into two basic categories:

● SCPI-compatible
● Non-SCPI-compatible

Command examples				
Function	**HP8642** **Non-SCPI**	**HP8657** **Non-SCPI**	**AF2023** **SCPI**	**SMW** **SCPI**
Resetting the instrument	PR	PR	*RST	*RST
Clearing the system status		CS	*CLS	*CLS
Setting the RF frequency	CW10MZ	FR10MZ	:CFRQ:VALUE 10MHZ	:FREQ 10MHZ
Setting the RF power	AP-10DB	PL-10DM	:RFLV:VALUE -10DBM	:POW -10DBM
Activating the RF output	ON	R3	:RFLV:ON	:OUTP 1

Older instruments often implement a simple, unstructured and non-SCPI-compatible language, whereas modern instruments implement usually a complex, well structured and SCPI-compatible language.
The SMW offers a solution for both kinds of languages using specialized processing units in non-SCPI-compatible languages:
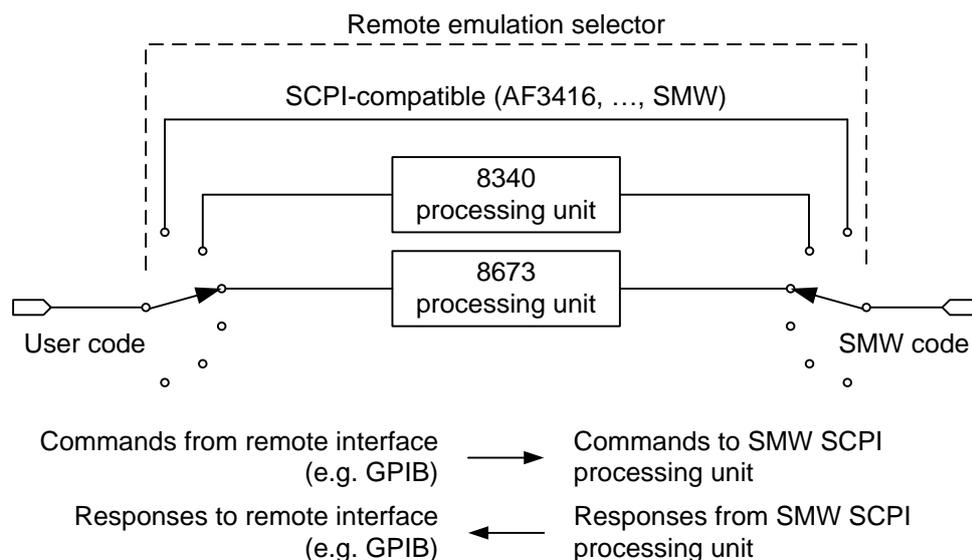
3.1.1 SCPI-Compatible Languages

Commands are routed directly from the remote interface to the SMW SCPI command processing unit; responses are routed in the reverse direction.

3.1.2 Non-SCPI-Compatible Languages

Commands are routed from the remote interface to the SMW SCPI command processing unit through a remote emulation specific processing unit; responses are routed in the reverse direction.
This specific processing unit parses the non-SCPI-compatible commands with reference to the corresponding syntax and translates them into SCPI-compatible ones. The unit also reformats the responses with respect to the requirements of the emulated instrument.

3.2 Remote Emulation Compatibility

An emulated instrument having fewer features than, or the same features as, the SMW can be replaced without special care.
However, replacing an emulated instrument having more features than the SMW or features that differ from those of the SMW requires additional care. The user must

- ensure that the SMW complies with the functional requirements
- verify that application code does not use features in the emulated instrument which are not available in the SMW

3.2.1 Command Compatibility

Most of the remote emulations in the SMW implement all commands of the original instrument. However, e.g. due to functional differences in hardware, in certain remote emulations the SMW

- does not support all commands
- does not support all parameters of a command
- implements a different behavior for a command

In the command tables of the different remote emulations, the status column gives compatibility information to a command:

Command compatibility status	
Status	**Comment**
✓	Command implementation is fully compatible.
✋	Command implementation is not fully compatible. The implementation in the SMW • does not support the same parameter(s) as the emulated instrument does • has different functionality than the emulated instrument • reports an invalid parameter or execution error if possible
○	Command is implemented without any functionality. The implementation in the SMW • ignores setting commands • returns default value in query commands • does not report errors • does not change any operating mode of the instrument • does not change any system state of the instrument
✗	Command is not implemented. The implementation in the SMW reports an unknown command error if possible.
+	Command has been added to enhance the functionality of the emulated instrument.

If the application software uses commands that are fully compatible, no special care has to be taken. The applications software can be used as is.
If the application software uses commands that are not fully compatible, the application software must be verified and normally also modified. If the required modifications to the application software are infeasible, the SMW cannot be used as replacement for another instrument.

3.2.2 IDN / OPT Strings

The remote emulation provides user-defined responses to *IDN? and *OPT? queries. This feature is of informational character only and has no impact on the functionality of the SMW.

3.2.3 Timing

The remote emulation cannot provide exact timing compatibility with the emulated instrument, since timing is a hardware-related property.

3.2.4 Sweep Operation

Some of the emulated instruments implement sweep operations with a constant total sweep time:
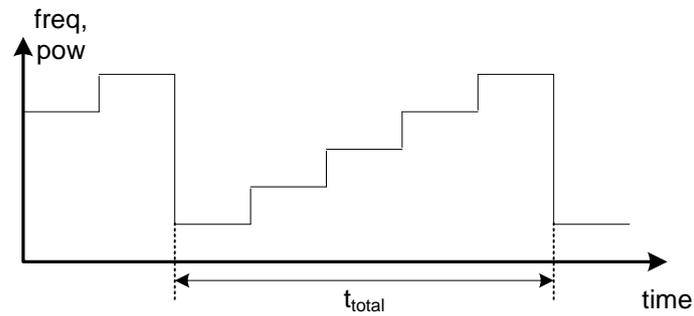
Changing the start, stop or steps parameter of the sweep operation does not have an impact on the total sweep time.

In the SMW, all sweep operations are based on the dwell time:
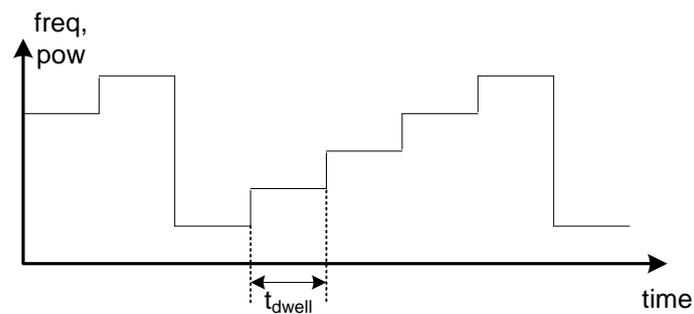
Changing the start, stop or steps parameter directly affects the total sweep time.

$$n_{steps} = \left| \frac{f_{stop} - f_{start}}{f_{step}} \right| + 1$$

$$t_{total} = n_{steps} \cdot t_{dwell}$$

3.2.5 Trigger Control

In signal generators with sweeping features, all sweep operating modes use the trigger control system. The figure below shows the SCPI trigger control system proposal:

pon or *RST or :ABOR

Idle

trigger system initiated
(:INIT[:IMM] or :INIT:CONT ON)

no longer initiated
(:INIT:CONT OFF)

Initiated

still initiated

number of arm
loops completed

Wait for arm

Wait for arm

Wait for arm

arm events

Wait for arm

arm conditions satisfied

number of trigger
loops completed

trigger event

Wait for trigger

trigger conditions satisfied

„device action" completed

Device action
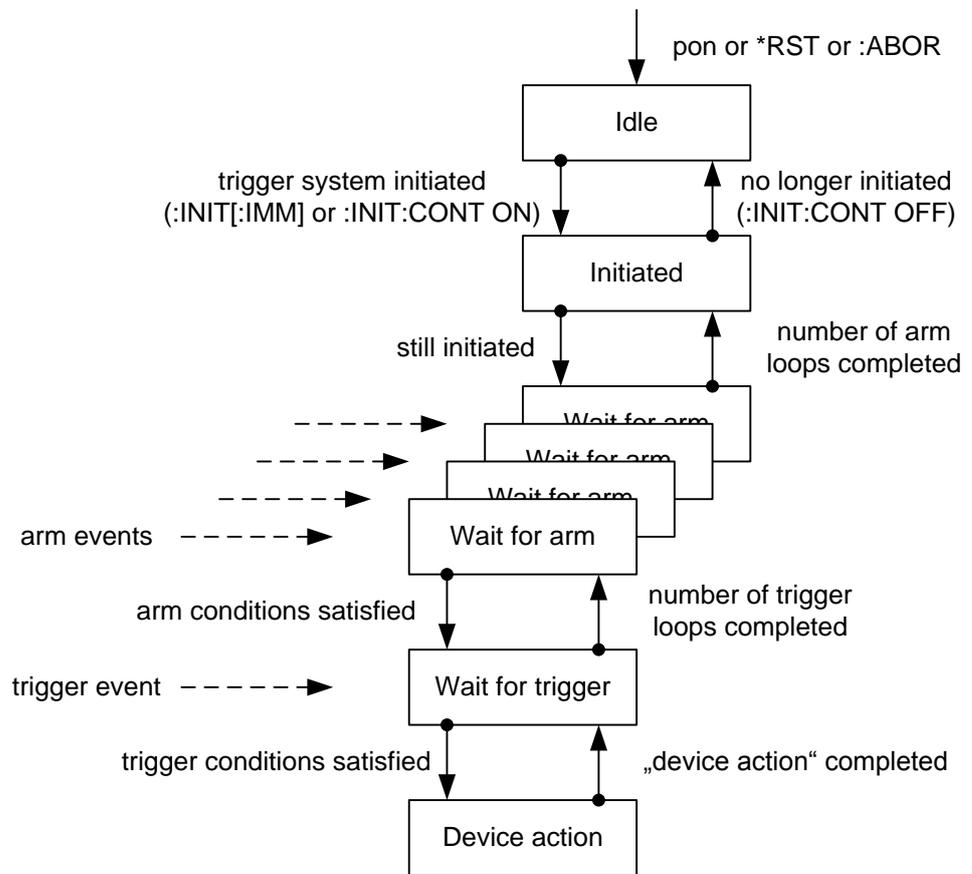
Some of the emulated instruments implement a simplified trigger control system:

pon or *RST or :ABOR

Idle

trigger system initiated
(:INIT[:IMM] or :INIT:CONT ON)

no longer initiated
(:INIT:CONT OFF)

Initiated

still initiated

number of trigger
loops completed

trigger event

Wait for trigger

trigger conditions satisfied

„device action"
completed

Device action
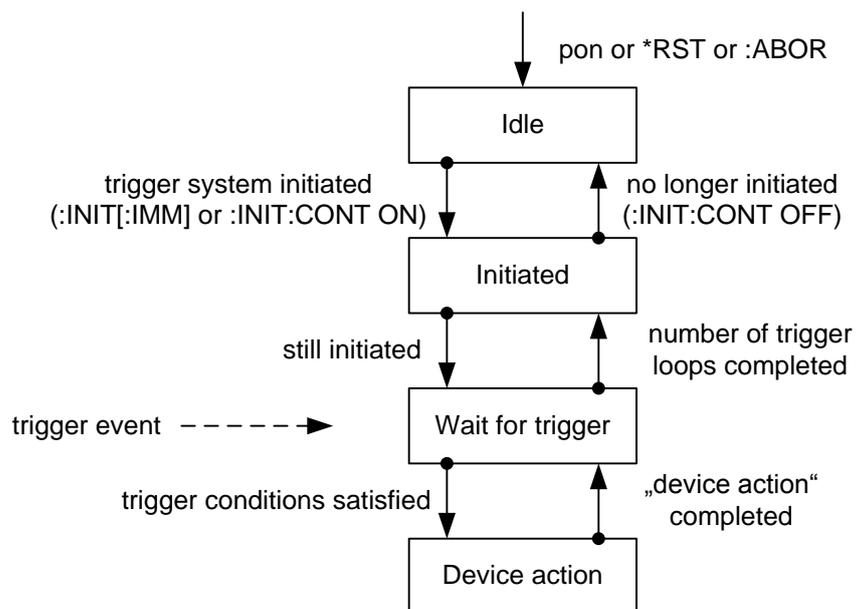
Other emulated instruments and also the SMW implement the trigger control system shown in the following figure:
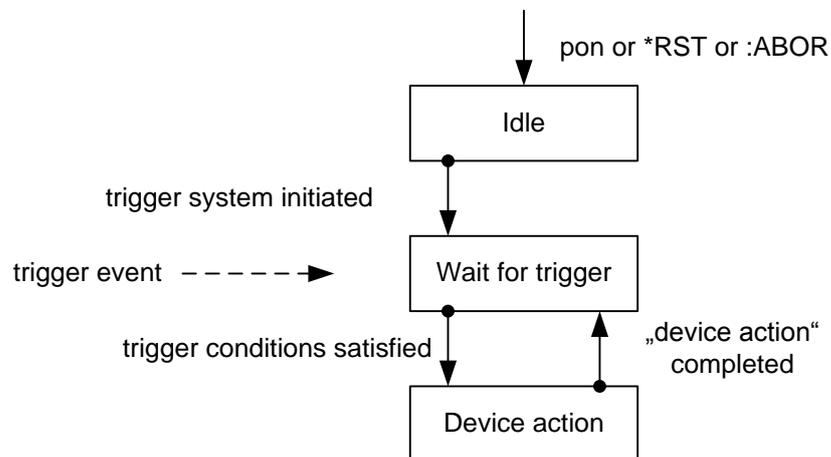
The main difference between the emulated instrument's and the SMW's trigger control system is that there is no "Initiated" state in the SMW. The SMW implementation assumes that the trigger system is initiated automatically in the "Idle" state. As a result, any sweep operation in the SMW, once activated, can only be controlled by internal or external trigger events.
This has an important impact on the user application. Application code that requires an "Initiated" state in the trigger control system must be adapted.

3.2.6 Status and Error Reporting

The SMW implements the minimal status and error reporting system required by the SCPI proposal. The following figure shows the status and error reporting model:

STAT:QUES — 0 0, 0 1, 0 2, 0 3, 0 4, 0 5, 0 6, 0 7, 0 8, 0 9, 0 10, 0 11, 0 12, 0 13, 0 14, 0 15 (+)

Output queue

Error/event queue

*ESR / *ESE
- Operation complete — 0
- 0 — 1
- Query error — 2
- Device-dependent error — 3
- Execution error — 4
- Command error — 5
- User request — 6
- Power on — 7
(+)

STAT:OPER
- Calibrating — 0
- 0 — 1
- 0 — 2
- Sweeping — 3
- 0 — 4
- 0 — 5
- 0 — 6
- 0 — 7
- 0 — 8
- 0 — 9
- 0 — 10
- 0 — 11
- 0 — 12
- 0 — 13
- 0 — 14
- 0 — 15
(+)

*STB: 0 0, 0 1, 2, 3, 4, 5, 6, 7
*SRE: 0, 1, 2, 3, 4, 5, 6, 7
(&)

Service request
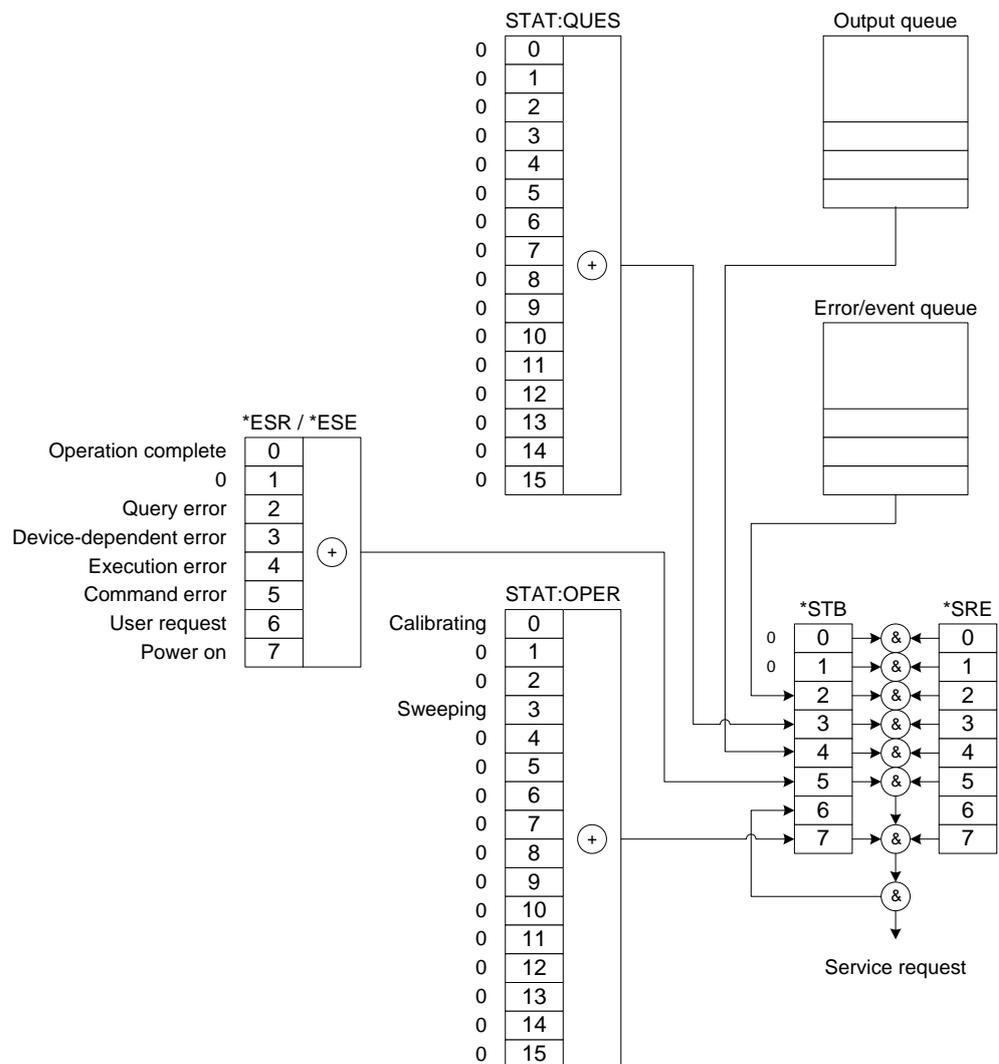
Some of the emulated instruments implement a more detailed status and error reporting system. Since the additional information stored in that system is not available in the SMW, application code that uses the additional information must be changed.

3.3 Preset / Reset

Changing the remote emulation does not automatically trigger a reset operation to the instrument. Therefore it is strongly recommended to manually execute a reset to the SMW after changing the remote emulation.
To apply the default of a particular remote emulation, the user must send the corresponding command via the remote control interface, e.g. the *RST command in SCPI-compatible languages.

3.4 Power Down / Power Up

The selected remote emulation and the user-defined responses to *IDN? and *OPT? queries are saved when the instrument is switched off.
When the instrument is switched on again, it starts up with the same settings that were active before it was switched off.

4 Activating a Remote Emulation

In order to use a specific remote emulation, it must first be activated by the user. Activation is done either

- manually using the SMW front panel
- remotely using SCPI commands

4.1 Manual Operation

On the SMW front panel, press the SETUP key to open the Menu tree:
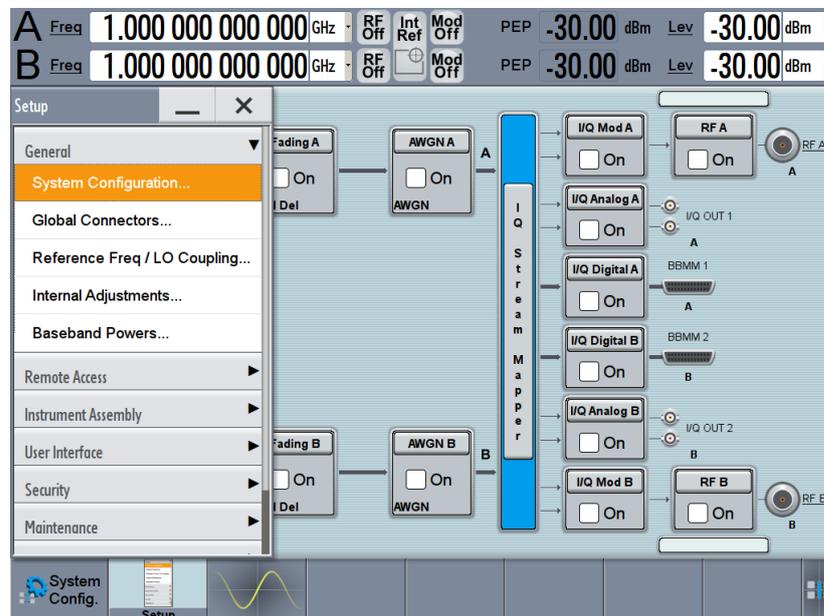
In the Menu tree, select Instrument Emulations… and open the Remote Access dialog:
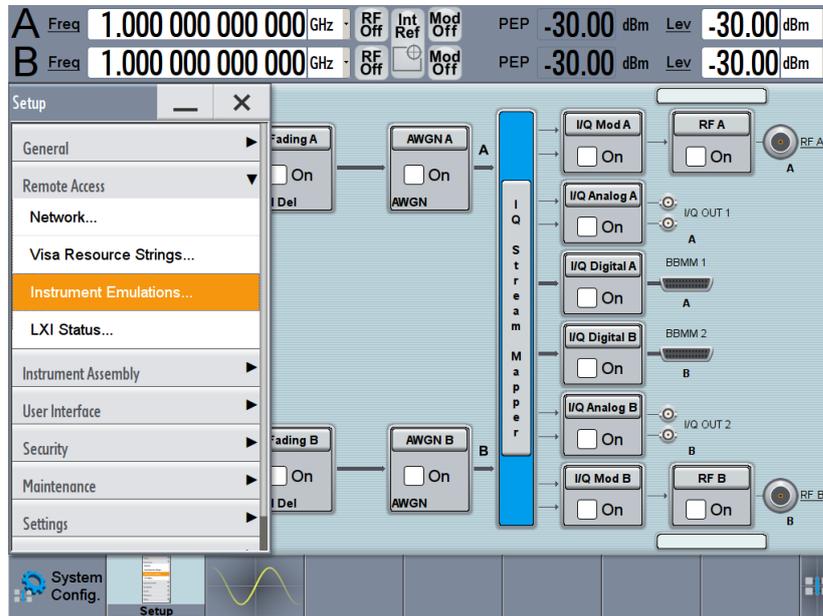
In the Remote Access dialog, set up the remote emulation specific parameters:
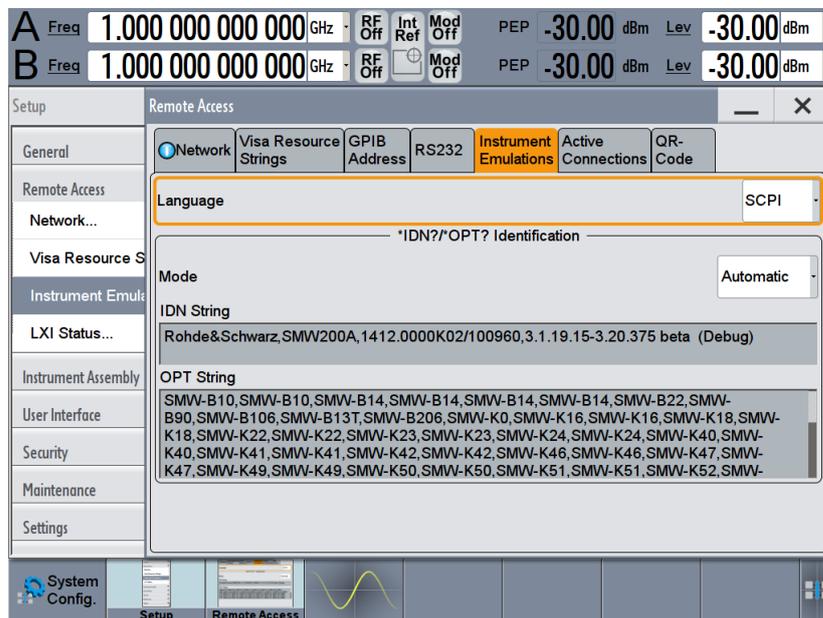
Select the Language and open the drop-down list. Pick an item from the list and confirm the selection:
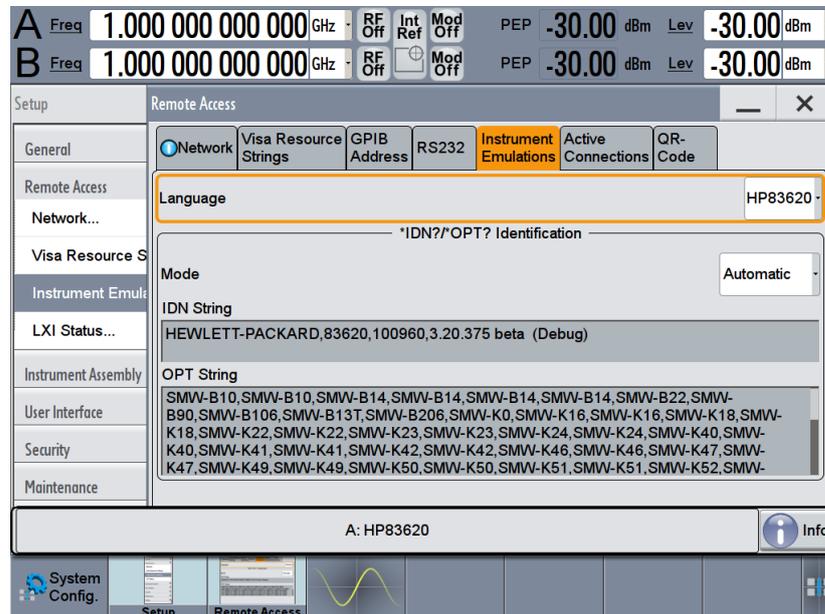
Select the Mode and open the drop-down list. Pick either "Automatic" or "User defined" from the list and confirm the selection. In "Automatic" mode, the response text to *IDN? and *OPT? queries is generated by the signal generator itself. In "User defined" mode, the response text to *IDN? and *OPT? queries must be entered in the corresponding IDN string and/or OPT string text boxes:
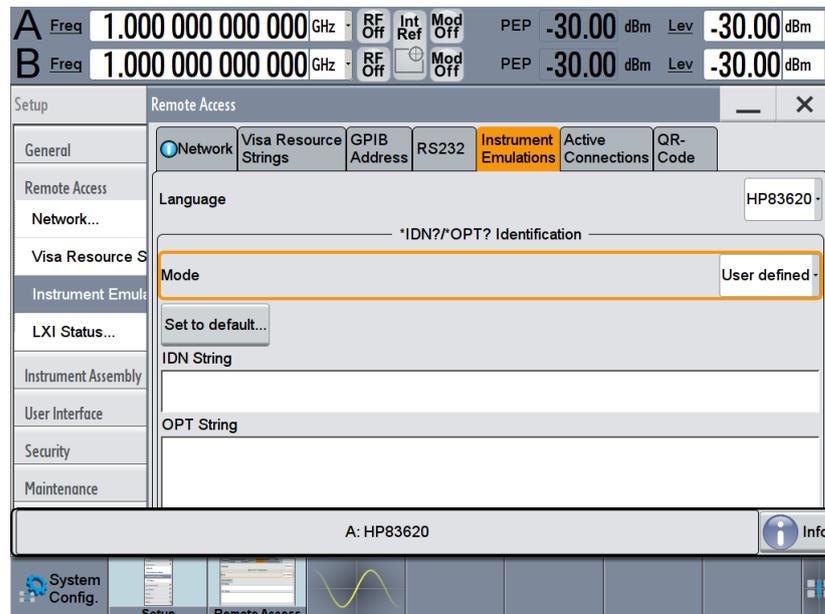
In "User defined" mode, press the "Set to default…" button to preset the corresponding IDN string and/or OPT string text boxes with the default settings from the selected remote emulation:
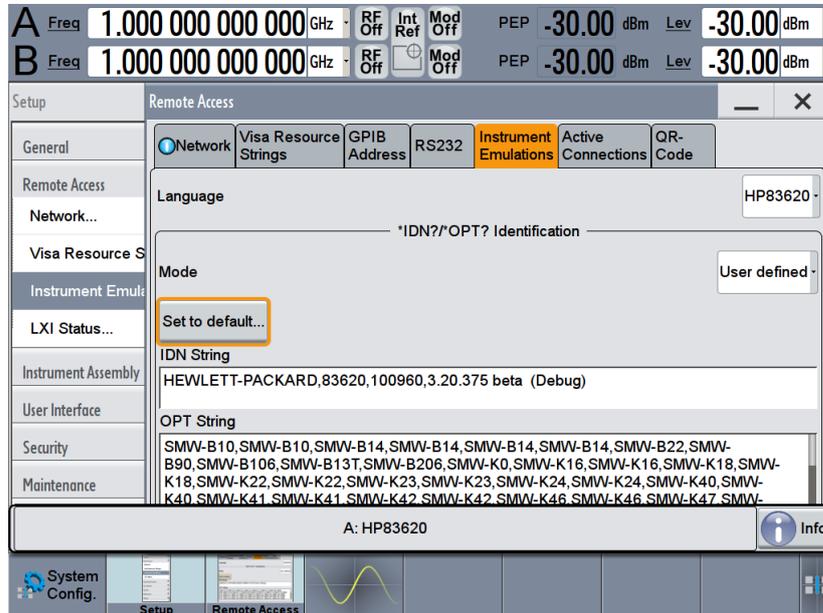
Then select the IDN String text box and edit the response text to *IDN? queries. When finished, confirm the text:
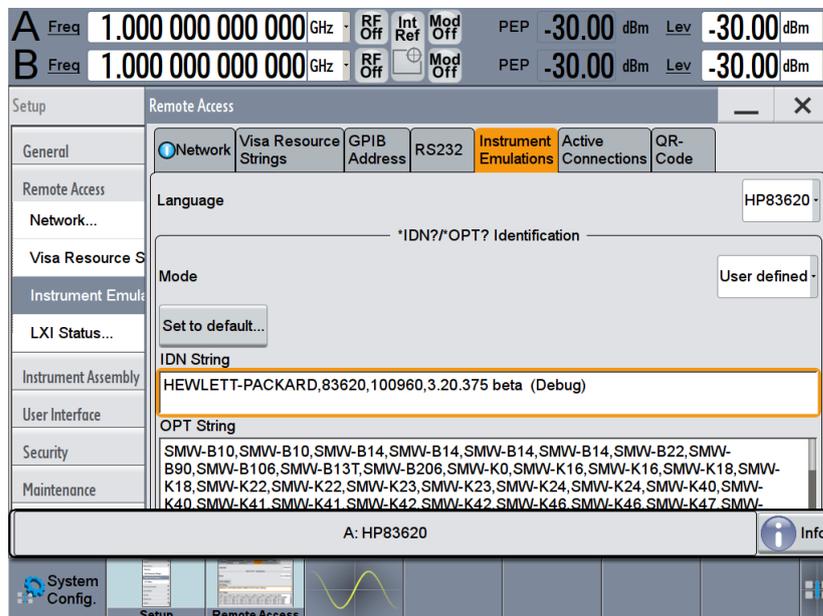
Finally select the OPT String text box and edit the response text to *OPT? queries. When finished, confirm the text:
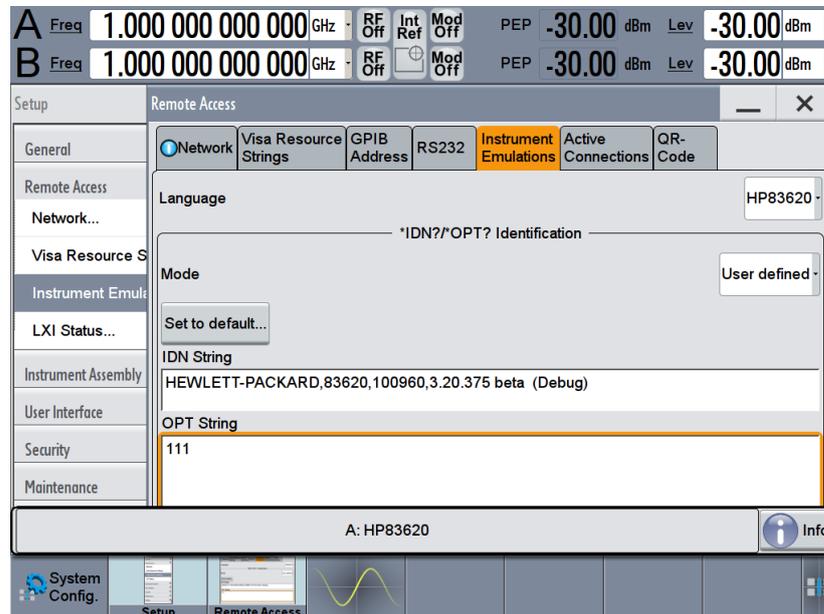
Exit the Remote Channel Settings dialog by pressing the ESC key.

4.2 Remote Operation

Use the following commands to modify the remote emulation parameters:

Commands to modify remote emulation relevant settings	
Command	Comment
:SYSTem:IDENt <value>	Sets the state of the identification mode.
	If the identification mode is set to USER, the value provided with the command :SYSTem:IDN <value> is returned on an *IDN? query and the value provided with the command :SYSTem:OPT <value> is returned on *OPT? query.
	If the identification mode is set to AUTO, the factory default setting is returned on a *IDN? query or an *OPT? query.
	The value range of character-type parameter <value> is
	• AUTO
	• USER
:SYSTem:IDENt?	Gets the state of the identification mode.
:SYSTem:IDN <value>	Sets the user-defined response to an *IDN? query.
	The string-type parameter <value> allows up to 128 characters. The parameter has to be enclosed in single or double quotes.
:SYSTem:IDN?	Gets the user-defined response to an *IDN? query.

Commands to modify remote emulation relevant settings	
Command	**Comment**
:SYSTem:OPT <value>	Sets the user-defined response to a *OPT? query.
	The string-type parameter <value> allows up to 128 characters. The parameter has to be enclosed in single or double quotes.
:SYSTem:OPT?	Gets the user-defined response to a *OPT? query.
:SYSTem:LANGuage <value>	Activates the remote emulation to be used for further communications.
	The value range of the string-type parameter <value> is:
	• "AF3416"
	• "E8257" , "HP83620", "HP83630", "HP83640", "HP83650"
	• "EXIT"
	Attention:
	The remote emulation is changed immediately after parsing this command. Succeeding commands such as *WAI, *OPC or *OPC? are not allowed, since these commands may not be a part of the newly selected command set.
	Therefore, this command must be the one and only command in a program message unit.
	After sending this command, a delay of two seconds must be applied to the application software before the next command is sent.
	The parameter value "EXIT" must be used to return to the native SCPI command set of the instrument.
:SYSTem:LANGuage?	Gets the current active remote emulation.

Note:

The upper-case and lower-case notation serves to distinguish between the long and the short form of a command. The instrument itself does not distinguish between upper-case and lower-case notation.

5 Emulating the IFR 3416

The following tables show the current implementation status of each command. Commands not shown in these tables are not supported.

IEEE488.2 functions	
Command syntax	**Status**
*CLS	✔
*ESE value *ESE? **Remark:** Layout of ESE register corresponds to Rohde & Schwarz signal generator.	✋
*ESR? **Remark:** Layout of ESE register corresponds to Rohde & Schwarz signal generator.	✋
*IDN? **Remark:** If the response does not match the requirements, a user-specific response to *IDN? and *OPT? can be applied on the instrument's front panel.	✋
*OPC *OPC?	✔
*OPT? **Remark:** If the response does not match the requirements, a user-specific response to *IDN? and *OPT? can be applied on the instrument's front panel.	✋
*RST	✔
*SRE value *SRE? **Remark:** Layout of ESE register corresponds to Rohde & Schwarz signal generator.	✋
*STB? **Remark:** Layout of ESE register corresponds to Rohde & Schwarz signal generator.	✋
*TST?	✔

Device-specific functions	
Command syntax	**Status**
DIAGnostic:INFormation:OTIMe?	✔
DIAGnostic:INFormation:OPTions:SOURce? **Remark:** Return value is always 0,0,0,0,0,0.	✋

Device-specific functions	
Command syntax	**Status**
DIAGnostic:INFormation:OPTions:SYSTem? **Remark:** Return value is always 0,0,0,0,0,0.	🖐
DISPlay:ANNotation[:ALL] value DISPlay:ANNotation[:ALL]?	✓
DISPlay:ANNotation:FREQuency value DISPlay:ANNotation:FREQuency?	✓
DISPlay:ANNotation:POWer value DISPlay:ANNotation:POWer?	✓
OUTPut:MODulation:AM[1]|2[:STATe] value OUTPut:MODulation:AM[1]|2[:STATe]?	✓
OUTPut:MODulation:FM[1]|2[:STATe] value OUTPut:MODulation:FM[1]|2[:STATe]?	✓
OUTPut:MODulation:IQ[:STATe] value OUTPut:MODulation:IQ[:STATe]?	✓
OUTPut:MODulation:PM[1]|2[:STATe] value OUTPut:MODulation:PM[1]|2[:STATe]?	✓
OUTPut:MODulation:PULM[:STATe] value OUTPut:MODulation:PULM[:STATe]?	✓
OUTPut:MODulation:RESet	✓
OUTPut:MODulation[:STATe] value OUTPut:MODulation[:STATe]	✓
OUTPut[:POWer]:PROTection:CLEar	✓
OUTPut[:POWer]:PROTection:TRIPped?	✓
OUTPut[:POWer][:STATe] value OUTPut[:POWer][:STATe]?	✓
ROSCillator:INTernal:ADJust[:VALue] value ROSCillator:INTernal:ADJust[:VALue]?	✓
ROSCillator:SOURce value ROSCillator:SOURce? **Remark:** INT, EXT10DIR are supported.	🖐
[SOURce:]FREQuency[:CW]|:FIXed value [SOURce:]FREQuency[:CW]|:FIXed?	✓
[SOURce:]FREQuency[:CW]|:FIXed:STEP[:INCRement] value [SOURce:]FREQuency[:CW]|:FIXed:STEP[:INCRement]?	✓

Device-specific functions	
Command syntax	**Status**
[SOURce:]FREQuency:MODE value [SOURce:]FREQuency:MODE?	✓
[SOURce:]FREQuency:PHASe[:ADJust] value [SOURce:]FREQuency:PHASe[:ADJust]?	✓
[SOURce:]FREQuency:SWEep:DWELl value [SOURce:]FREQuency:SWEep:DWELl?	✓
[SOURce:]FREQuency:SWEep:MANual value [SOURce:]FREQuency:SWEep:MANual?	✓
[SOURce:]FREQuency:SWEep:SPACing value [SOURce:]FREQuency:SWEep:SPACing?	✓
[SOURce:]FREQuency:SWEep:STARt value [SOURce:]FREQuency:SWEep:STARt?	✓
[SOURce:]FREQuency:SWEep:STEP[:LINear] value [SOURce:]FREQuency:SWEep:STEP[:LINear]?	✓
[SOURce:]FREQuency:SWEep:STEP:LOGarithmic value [SOURce:]FREQuency:SWEep:STEP:LOGarithmic?	✓
[SOURce:]FREQuency:SWEep:STOP value [SOURce:]FREQuency:SWEep:STOP?	✓
[SOURce:]LIST:ABORt	✓
[SOURce:]LIST:CALCulate	○
[SOURce:]LIST:CLEar value	✓
[SOURce:]LIST:CLEar:ALL	✓
[SOURce:]LIST:CLEar:TEND value	✓
[SOURce:]LIST:DELete value	✓
[SOURce:]LIST:DWELl value [SOURce:]LIST:DWELl?	✓
[SOURce:]LIST:FREQuency value, value [,value] [SOURce:]LIST:FREQuency?	✓
[SOURce:]LIST:INITiate	✓
[SOURce:]LIST:INSert value, value, value	✓
[SOURce:]LIST:OPERation value [SOURce:]LIST:OPERation?	✓
[SOURce:]LIST:POWer value, value [,value] [SOURce:]LIST:POWer?	✓
[SOURce:]LIST:RESet	✓

Device-specific functions	
Command syntax	**Status**
[SOURce:]LIST:STARt value [SOURce:]LIST:STARt?	✔
[SOURce:]LIST:STOP value [SOURce:]LIST:STOP?	✔
[SOURce:]LIST:TRIGger[:MODE] value [SOURce:]LIST:TRIGger[:MODE]? **Remark:** OFF, STARt, STEP are supported.	🖐
[SOURce:]LIST:TRIGger:SLOPe value [SOURce:]LIST:TRIGger:SLOPe?	✔
[SOURce:]LIST:VALue value,value,value [SOURce:]LIST:VALue?	✔
[SOURce:][MODulation:]AM[1]|2[:DEPth] value [SOURce:][MODulation:]AM[1]|2[:DEPth]?	✔
[SOURce:][MODulation:]AM[1]|2[:DEPth]:STEP[:INCRement] value [SOURce:][MODulation:]AM[1]|2[:DEPth]:STEP[:INCRement]?	✔
[SOURce:][MODulation:]AM[1]|2:EXTernal:COUPling value [SOURce:][MODulation:]AM[1]|2:EXTernal:COUPling?	✔
[SOURce:][MODulation:]AM[1]|2:EXTernal:IMPedance value [SOURce:][MODulation:]AM[1]|2:EXTernal:IMPedance?	✔
[SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency[:FIXed] value [SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency[:FIXed]?	✔
[SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency[:FIXed]:STEP[:INCRement] value [SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency[:FIXed]:STEP[:INCRement]?	✔
[SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:MODE value [SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:MODE? **Remark:** AM[1] is supported.	🖐
[SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:DWELl value [SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:DWELl? **Remark:** AM[1] is supported.	🖐
[SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:MANual value [SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:MANual? **Remark:** AM[1] is supported.	🖐

Device-specific functions	
Command syntax	**Status**
[SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:SPACing value [SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:SPACing? **Remark:** AM[1] is supported.	✋
[SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:STARt value [SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:STARt? **Remark:** AM[1] is supported.	✋
[SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:STEP[:LINear] value [SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:STEP[:LINear]? **Remark:** AM[1] is supported.	✋
[SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:STEP:LOGarithmic value [SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:STEP:LOGarithmic? **Remark:** AM[1] is supported.	✋
[SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:STOP value [SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:STOP? **Remark:** AM[1] is supported.	✋
[SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:SPACing value [SOURce:][MODulation:]AM[1]|2:INTernal:FREQuency:SWEep:SPACing? **Remark:** AM[1] is supported.	✋
[SOURce:][MODulation:]AM[1]|2:INTernal:SHAPe value [SOURce:][MODulation:]AM[1]|2:INTernal:SHAPe? **Remark:** SINE, SQUare, TRIangle are supported.	✋
[SOURce:][MODulation:]AM[1]|2:SOURce value [SOURce:][MODulation:]AM[1]|2:SOURce?	✓
[SOURce:][MODulation:]AM[1]|2:STATe value [SOURce:][MODulation:]AM[1]|2:STATe?	✓
[SOURce:][MODulation:]FM[1]|2[:DEViation] value [SOURce:][MODulation:]FM[1]|2[:DEViation]?	✓
[SOURce:][MODulation:]FM[1]|2[:DEViation]:STEP[:INCRement] value [SOURce:][MODulation:]FM[1]|2[:DEViation]:STEP[:INCRement]?	✓
[SOURce:][MODulation:]FM[1]|2:EXTernal:COUPling value [SOURce:][MODulation:]FM[1]|2:EXTernal:COUPling?	✓
[SOURce:][MODulation:]FM[1]|2:EXTernal:DNULI	✓

Device-specific functions	
Command syntax	**Status**
[SOURce:][MODulation:]FM[1]|2:EXTernal:IMPedance value [SOURce:][MODulation:]FM[1]|2:EXTernal:IMPedance?	✓
[SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency[:FIXed] value [SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency[:FIXed]?	✓
[SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency[:FIXed]:STEP[:INCRement] value [SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency[:FIXed]:STEP[:INCRement]?	✓
[SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:MODE value [SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:MODE? **Remark:** FM[1] is supported.	🖐
[SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:DWELl value [SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:DWELl? **Remark:** FM[1] is supported.	🖐
[SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:MANual value [SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:MANual? **Remark:** FM[1] is supported.	🖐
[SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:SPACing value [SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:SPACing? **Remark:** FM[1] is supported.	🖐
[SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:STARt value [SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:STARt? **Remark:** FM[1] is supported.	🖐
[SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:STEP[:LINear] value [SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:STEP[:LINear]? **Remark:** FM[1] is supported.	🖐
[SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:STEP:LOGarithmic value [SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:STEP:LOGarithmic? **Remark:** FM[1] is supported.	🖐
[SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:STOP value [SOURce:][MODulation:]FM[1]|2:INTernal:FREQuency:SWEep:STOP? **Remark:** FM[1] is supported.	🖐

Device-specific functions	
Command syntax	**Status**
[SOURce:][MODulation:]FM[1]|2:INTernal:SHAPe value [SOURce:][MODulation:]FM[1]|2:INTernal:SHAPe? **Remark:** SINE, SQUare, TRIangle are supported.	✋
[SOURce:][MODulation:]FM[1]|2:SOURce value [SOURce:][MODulation:]FM[1]|2:SOURce?	✔
[SOURce:][MODulation:]FM[1]|2:STATe value [SOURce:][MODulation:]FM[1]|2:STATe?	✔
[SOURce:][MODulation:]PM[1]|2[:DEViation] value [SOURce:][MODulation:]PM[1]|2[:DEViation]?	✔
[SOURce:][MODulation:]PM[1]|2[:DEViation]:STEP[:INCRement] value [SOURce:][MODulation:]PM[1]|2[:DEViation]:STEP[:INCRement]?	✔
[SOURce:][MODulation:]PM[1]|2:EXTernal:COUPling value [SOURce:][MODulation:]PM[1]|2:EXTernal:COUPling?	✔
[SOURce:][MODulation:]PM[1]|2:EXTernal:IMPedance value [SOURce:][MODulation:]PM[1]|2:EXTernal:IMPedance?	✔
[SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency[:FIXed] value [SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency[:FIXed]?	✔
[SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency[:FIXed]:STEP[:INCRement] value [SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency[:FIXed]:STEP[:INCRement]?	✔
[SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:MODE value [SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:MODE? **Remark:** PM[1] is supported.	✋
[SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:DWELl value [SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:DWELl? **Remark:** PM[1] is supported.	✋
[SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:MANual value [SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:MANual? **Remark:** PM[1] is supported.	✋
[SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:SPACing value [SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:SPACing? **Remark:** PM[1] is supported.	✋

Device-specific functions	
Command syntax	**Status**
[SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:STARt value [SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:STARt? **Remark:** PM[1] is supported.	✋
[SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:STEP[:LINear] value [SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:STEP[:LINear]? **Remark:** PM[1] is supported.	✋
[SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:STEP:LOGarithmic value [SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:STEP:LOGarithmic? **Remark:** PM[1] is supported.	✋
[SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:STOP value [SOURce:][MODulation:]PM[1]|2:INTernal:FREQuency:SWEep:STOP? **Remark:** PM[1] is supported.	✋
[SOURce:][MODulation:]PM[1]|2:INTernal:SHAPe value [SOURce:][MODulation:]PM[1]|2:INTernal:SHAPe? **Remark:** SINE, SQUare, TRIangle are supported.	✋
[SOURce:][MODulation:]PM[1]|2:SOURce value [SOURce:][MODulation:]PM[1]|2:SOURce?	✔
[SOURce:][MODulation:]PM[1]|2:STATe value [SOURce:][MODulation:]PM[1]|2:STATe?	✔
[SOURce:][MODulation:]PULM:SOURce value [SOURce:][MODulation:]PULM:SOURce?	✔
[SOURce:][MODulation:]PULM:STATe value [SOURce:][MODulation:]PULM:STATe?	✔
[SOURce:]POWer:ALC:STATe value [SOURce:]POWer:ALC:STATe?	✔
[SOURce:]POWer[:LEVel][:IMMediate][:AMPlitude] value [SOURce:]POWer[:LEVel][:IMMediate][:AMPlitude]?	✔
[SOURce:]POWer[:LEVel][:IMMediate][:AMPlitude]:STEP[:INCRement] value [SOURce:]POWer[:LEVel][:IMMediate][:AMPlitude]:STEP[:INCRement]?	✔
[SOURce:]POWer[:LEVel][:IMMediate][:AMPlitude]:OFFSet:ATTenuation value [SOURce:]POWer[:LEVel][:IMMediate][:AMPlitude]:OFFSet:ATTenuation?	✔
[SOURce:]POWer[:LEVel][:IMMediate][:AMPlitude]:OFFSet[:GAIN] value [SOURce:]POWer[:LEVel][:IMMediate][:AMPlitude]:OFFSet[:GAIN]?	✔

Device-specific functions	
Command syntax	**Status**
[SOURce:]POWer[:LEVel][:IMMediate][:AMPlitude]:OFFSet:LOSS value [SOURce:]POWer[:LEVel][:IMMediate][:AMPlitude]:OFFSet:LOSS?	✓
[SOURce:]POWer[:LEVel][:IMMediate][:AMPlitude]:OFFSet:STATe value [SOURce:]POWer[:LEVel][:IMMediate][:AMPlitude]:OFFSet:STATe?	✓
[SOURce:]POWer:LIMit[:IMMediate][:AMPlitude] value [SOURce:]POWer:LIMit[:IMMediate][:AMPlitude]?	✓
[SOURce:]POWer:MODE value [SOURce:]POWer:MODE?	✓
[SOURce:]POWer:SWEep:DWELl value [SOURce:]POWer:SWEep:DWELl?	✓
[SOURce:]POWer:SWEep:MANual value [SOURce:]POWer:SWEep:MANual?	✓
[SOURce:]POWer:SWEep:STARt value [SOURce:]POWer:SWEep:STARt?	✓
[SOURce:]POWer:SWEep:STEP value [SOURce:]POWer:SWEep:STEP?	✓
[SOURce:]POWer:SWEep:STOP value [SOURce:]POWer:SWEep:STOP?	✓
[SOURce:]SWEep:ABORt	✓
[SOURce:]SWEep:INITiate	✓
[SOURce:]SWEep:OPERation value [SOURce:]SWEep:OPERation?	✓
[SOURce:]SWEep:RESet	✓
[SOURce:]SWEep:TRIGger[:MODE] value [SOURce:]SWEep:TRIGger[:MODE]? **Remark:** OFF, STARt, STEP are supported.	✋
[SOURce:]SWEep:TRIGger:SLOPe value [SOURce:]SWEep:TRIGger:SLOPe?	✓
STATus:OPERation:CONDition?	✓
STATus:OPERation:ENABle value STATus:OPERation:ENABle?	✓
STATus:OPERation[:EVENt]?	✓
STATus:OPERation:NTRansition value STATus:OPERation:NTRansition?	✓
STATus:OPERation:PTRansition value STATus:OPERation:PTRansition?	✓

Device-specific functions	
Command syntax	**Status**
STATus:PRESet	✓
STATus:QUEStionable:CONDition?	✓
STATus:QUEStionable:ENABle value STATus:QUEStionable:ENABle?	✓
STATus:QUEStionable[:EVENt]?	✓
STATus:QUEStionable:NTRansition value STATus:QUEStionable:NTRansition?	✓
STATus:QUEStionable:PTRansition value STATus:QUEStionable:PTRansition?	✓
SYSTem:COMMunicate:ETHernet:ADDRess value, value SYSTem:COMMunicate:ETHernet:ADDRess?	✓
SYSTem:COMMunicate:ETHernet:AUTO value SYSTem:COMMunicate:ETHernet:AUTO?	✓
SYSTem:COMMunicate:ETHernet:HNAMe value SYSTem:COMMunicate:ETHernet:HNAMe?	✓
SYSTem:COMMunicate:ETHernet:MADDress value SYSTem:COMMunicate:ETHernet:MADDress?	✓
SYSTem:COMMunicate:GPIB[:SELF]:ADDRess value SYSTem:COMMunicate:GPIB[:SELF]:ADDRess?	✓
SYSTem:COMMunicate:SERial:BAUD value SYSTem:COMMunicate:SERial:BAUD? **Remark:** Returns the original error code of the Rohde & Schwarz signal generator.	✋
SYSTem:COMMunicate:SERial:CONTrol:HANDshake value SYSTem:COMMunicate:SERial:CONTrol:HANDshake? **Remark:** HW, SW are supported.	✋
SYSTem:COMMunicate:SERial:PARity value SYSTem:COMMunicate:SERial:PARity?	✓
SYSTem:COMMunicate:SERial:SBITs value SYSTem:COMMunicate:SERial:SBITs?	✓
SYSTem:ERRor:ALL?	✓
SYSTem:ERRor:CODE:ALL?	✓
SYSTem:ERRor:CODE[:NEXT]?	✓
SYSTem:ERRor:COUNt?	✓
SYSTem:ERRor[:NEXT]?	✓

Device-specific functions	
Command syntax	**Status**
SYSTem:KLOCk value SYSTem:KLOCk?	✓
SYSTem:LANGuage value SYSTem:LANGuage? **Remark:** "EXIT" returns to native SCPI mode.	✋
SYSTem:PRESet	✓
SYSTem:SETTings:FULL:RECall value	✓
SYSTem:SETTings:FULL:SAVE value	✓
UNIT:POWer value UNIT:POWer?	✓
UNIT:VOLTtype value UNIT:VOLTtype?	✓

6 Emulating the Agilent E8257

The remote emulation E8257 is available only if one of the following options is installed:

- SMW-B112   RF Path 100 kHz to 12.75 GHz
- SMW-B120   RF Path 100 kHz to 20 GHz
- SMW-B131   RF Path 100 kHz to 31.8 GHz
- SMW-B140   RF Path 100 kHz to 40 GHz

The following tables show the current implementation status of each command. Commands not shown in these tables are not supported.

IEEE488.2 functions	
Command syntax	**Status**
*CLS	✔
*ESE value *ESE? **Remark:** Layout of ESE register corresponds to Rohde & Schwarz signal generator.	🖐
*ESR? **Remark:** Layout of ESE register corresponds to Rohde & Schwarz signal generator.	🖐
*IDN? **Remark:** If the response does not match the requirements, a user-specific response to *IDN? and *OPT? can be applied on the instrument's front panel.	🖐
*OPC *OPC?	✔
*OPT? **Remark:** If the response does not match the requirements, a user-specific response to *IDN? and *OPT? can be applied on the instrument's front panel.	🖐
*RST	✔
*SRE value *SRE? **Remark:** Layout of ESE register corresponds to Rohde & Schwarz signal generator.	🖐
*STB? **Remark:** Layout of ESE register corresponds to Rohde & Schwarz signal generator.	🖐
*TRG	✔
*TST?	✔

IEEE488.2 functions	
Command syntax	**Status**
*WAI	✓

Device-specific functions	
Command syntax	**Status**
ABORt	✓
CALibration:DCFM	✓
CALibration:IQ:FULL	✓
DISPlay:ANNotation:AMPLitude[:STATe] value DISPlay:ANNotation:AMPLitude[:STATe]?	✓
DISPlay:ANNotation:FREQuency[:STATe] value DISPlay:ANNotation:FREQuency[:STATe]?	✓
DISPlay:REMote value DISPlay:REMote?	✓
INITiate:CONTinuous[:ALL] value INITiate:CONTinuous[:ALL]?	○
INITiate:IMMediate[:ALL]	○
MEMory:CATalog[:ALL]?	✓
MEMory:CATalog:LIST?	✓
MEMory:CATalog:ULFT?	✓
MEMory:COPY[:NAME] value, value	✓
MEMory:DATA value,value MEMory:DATA? value	✓
MEMory:DATA:UNPRotected value, value MEMory:DATA:UNPRotected? Value **Remark:** Implementation identical to MEMory:DATA[?].	🖐
MEMory:DELete:ALL	✓
MEMory:DELete:LIST	✓
MEMory:DELete[:NAME] value	✓
MEMory:DELete:ULFT	✓
MEMory:FREE[:ALL]	✓
MEMory:LOAD:LIST value	✓
MEMory:MOVE	✓
MEMory:STORE:LIST value	✓
MMEMory:CATalog[:ALL]? value	✓

Device-specific functions	
Command syntax	**Status**
MMEMory:COPY value, value	✓
MMEMory:DATA value, value MMEMory:DATA? value	✓
MMEMory:DELete[:NAME] value	✓
MMEMory:LOAD:LIST value	✓
MMEMory:LOAD:MOVE value, value	✓
MMEMory:STORe:LIST value	✓
OUTPut:MODulation[:STATe] value OUTPut:MODulation[:STATe]?	✓
OUTPut[:STATe] value OUTPut[:STATe]?	✓
[SOURce:]AM[1] [:DEPth][:LINear] value [SOURce:]AM[1] [:DEPth][:LINear]?	✓
[SOURce:]AM[1] [:DEPth]:STEP[:INCRement] value [SOURce:]AM[1] [:DEPth]:STEP[:INCRement]?	✓
[SOURce:]AM[1]:EXTernal[1]:COUPling value [SOURce:]AM[1]:EXTernal[1]:COUPling?	✓
[SOURce:]AM[1]:INTernal[1]:FREQuency value [SOURce:]AM[1]:INTernal[1]:FREQuency?	✓
[SOURce:]AM[1]:INTernal[1]:FREQuency:STEP[:INCRement] value [SOURce:]AM[1]:INTernal[1]:FREQuency:STEP[:INCRement]?	✓
[SOURce:]AM[1]:INTernal[1]:FUNCtion:SHAPe value [SOURce:]AM[1]:INTernal[1]:FUNCtion:SHAPe? **Remark:** SINE is supported.	🖑
[SOURce:]AM[1]:INTernal[1]:SWEep:TIME value [SOURce:]AM[1]:INTernal[1]:SWEep:TIME? **Remark:** Dwell time is assumed.	🖑
[SOURce:]AM[1]:INTernal[1]:SWEep:TRIGger value [SOURce:]AM[1]:INTernal[1]:SWEep:TRIGger? **Remark:** BUS, EXTernal, IMMediate are supported.	🖑
[SOURce:]AM[1]:SOURce value [SOURce:]AM[1]:SOURce? **Remark:** EXTernal[1], INTernal[1] are supported.	🖑

Device-specific functions	
Command syntax	**Status**
[SOURce:]AM[1]:STATe value [SOURce:]AM[1]:STATe?	✔
[SOURce:]CORRection:FLATness:LOAD value	✔
[SOURce:]CORRection:FLATness:PAIR value, value {value, value} [SOURce:]CORRection:FLATness:PAIR?	✔
[SOURce:]CORRection:FLATness:POINts?	✔
[SOURce:]CORRection:FLATness:PRESet	✔
[SOURce:]CORRection:FLATness:STORe value	✔
[SOURce:]CORRection[:STATe] value [SOURce:]CORRection[:STATe]?	✔
[SOURce:]FM[1][:DEViation] value [SOURce:]FM[1][:DEViation]? **Remark:** DOWN, MINImum, Numeric, UP are supported.	🖐
[SOURce:]FM[1][:DEViation]:STEP[:INCRement] value [SOURce:]FM[1][:DEViation]:STEP[:INCRement]?	✔
[SOURce:]FM[1]|2:EXTernal[12:COUPling value [SOURce:]FM[1]|2:EXTernal[1]:COUPling?	✔
[SOURce:]FM[1]:INTernal[1]:FREQuency value [SOURce:]FM[1]:INTernal[1]:FREQuency?	✔
[SOURce:]FM[1]:INTernal[1]:FREQuency:STEP[:INCRement] value [SOURce:]FM[1]:INTernal[1]:FREQuency:STEP[:INCRement]?	✔
[SOURce:]FM[1]:INTernal[1]:FUNCtion:SHAPe value [SOURce:]FM[1]:INTernal[1]:FUNCtion:SHAPe? **Remark:** SINE is supported.	🖐
[SOURce:]FM[1]:INTernal[1]:SWEep:TIME value [SOURce:]FM[1]:INTernal[1]:SWEep:TIME? **Remark:** Dwell time is assumed.	🖐
[SOURce:]FM[1]:INTernal[1]:SWEep:TRIGger value [SOURce:]FM[1]:INTernal[1]:SWEep:TRIGger? **Remark:** BUS, EXTernal, IMMediate are supported.	🖐
[SOURce:]FM[1]:SOURce value [SOURce:]FM[1]:SOURce? **Remark:** EXTernal[1], INTernal[1] are supported.	🖐

Device-specific functions	
Command syntax	**Status**
[SOURce:]FM[1]:STATe value [SOURce:]FM[1]:STATe?	✓
[SOURce:]FREQuency[:CW] value [SOURce:]FREQuency[:CW]?	✓
[SOURce:]FREQuency[:CW]:STEP[:INCRement] value [SOURce:]FREQuency[:CW]:STEP[:INCRement]?	✓
[SOURce:]FREQuency:FIXed value [SOURce:]FREQuency:FIXed?	✓
[SOURce:]FREQuency:MODE [SOURce:]FREQuency:MODE?	✓
[SOURce:]FREQuency:OFFSet value [SOURce:]FREQuency:OFFSet?	✓
[SOURce:]FREQuency:OFFSet:STATe value [SOURce:]FREQuency:OFFSet:STATe?	✓
[SOURce:]FREQuency:REFerence value [SOURce:]FREQuency:REFerence?	✓
[SOURce:]FREQuency:REFerence:STATe value [SOURce:]FREQuency:REFerence:STATe?	✓
[SOURce:]FREQuency:STARt value [SOURce:]FREQuency:STARt?	✓
[SOURce:]FREQuency:STOP value [SOURce:]FREQuency:STOP?	✓
[SOURce:]LFOutput:AMPLitude value [SOURce:]LFOutput:AMPLitude?	✓
[SOURce:]LFOutput:FUNCtion[1]:FREQuency value [SOURce:]LFOutput:FUNCtion[1]:FREQuency?	✓
[SOURce:]LFOutput:FUNCtion[1]:FREQuency:STEP[:INCRement] value [SOURce:]LFOutput:FUNCtion[1]:FREQuency:STEP[:INCRement]?	✓
[SOURce:]LFOutput:FUNCtion[1]:PERiod value [SOURce:]LFOutput:FUNCtion[1]:PERiod?	✓
[SOURce:]LFOutput:FUNCtion[1]:PERiod:STEP[:INCRement] value [SOURce:]LFOutput:FUNCtion[1]:PERiod:STEP[:INCRement]?	✓
[SOURce:]LFOutput:FUNCtion[1]:PWIDth value [SOURce:]LFOutput:FUNCtion[1]:PWIDth?	✓
[SOURce:]LFOutput:FUNCtion[1]:PWIDth:STEP[:INCRement] value [SOURce:]LFOutput:FUNCtion[1]:PWIDth:STEP[:INCRement]?	✓

Device-specific functions	
Command syntax	**Status**
[SOURce:]LFOutput:FUNCtion[1]:SHAPe value [SOURce:]LFOutput:FUNCtion[1]:SHAPe? **Remark:** SINE is supported.	🖐
[SOURce:]LFOutput:FUNCtion[1]:SWEep:TRIGger value [SOURce:]LFOutput:FUNCtion[1]:SWEep:TRIGger? **Remark:** BUS, EXTernal, IMMediate are supported.	🖐
[SOURce:]LFOutput:FUNCtion[1]:SOURce value [SOURce:]LFOutput:FUNCtion[1]:SOURce? **Remark:** INTernal[1] is supported.	🖐
[SOURce:]LFOutput:FUNCtion[1]:STATe value [SOURce:]LFOutput:FUNCtion[1]:STATe?	✔
[SOURce:]LIST:DIRection?	◯
[SOURce:]LIST:DWELl value {,value} [SOURce:]LIST:DWELl? **Remark:** First parameter value is evaluated, rest is ignored.	🖐
[SOURce:]LIST:DWELl:POINts?	✔
[SOURce:]LIST:DWELl:TYPE value [SOURce:]LIST:DWELl:TYPE?	✔
[SOURce:]LIST:FREQuency value {,value} [SOURce:]LIST:FREQuency?	✔
[SOURce:]LIST:FREQuency:POINts?	✔
[SOURce:]LIST:MODE value [SOURce:]LIST:MODE?	✔
[SOURce:]LIST:POWer value {,value} [SOURce:]LIST:POWer?	✔
[SOURce:]LIST:POWer:POINts?	✔
[SOURce:]LIST:TYPE value [SOURce:]LIST:TYPE?	✔
[SOURce:]LIST:TYPE:LIST:INITialize:FSTep	✔
[SOURce:]LIST:TYPE:LIST:INITialize:PREset	✔
[SOURce:]PHASe[:ADJust] value [SOURce:]PHASe[:ADJust]?	✔
[SOURce:]PHASe:REFerence	✔

Device-specific functions	
Command syntax	**Status**
[SOURce:]PM[1][:DEViation] value [SOURce:]PM[1][:DEViation]? **Remark:** DOWN, MINImum, Numeric, UP are supported.	🖐
[SOURce:]PM[1][:DEViation]:STEP[:INCRement] value [SOURce:]PM[1][:DEViation]:STEP[:INCRement]?	✓
[SOURce:]PM[1]:EXTernal[1]:COUPling value [SOURce:]PM[1]:EXTernal[1]:COUPling?	✓
[SOURce:]PM[1]:INTernal[1]:FREQuency value [SOURce:]PM[1]:INTernal[1]:FREQuency?	✓
[SOURce:]PM[1]:INTernal[1]:FREQuency:STEP[:INCRement] value [SOURce:]PM[1]:INTernal[1]:FREQuency:STEP[:INCRement]?	✓
[SOURce:]PM[1]:INTernal[1]:FUNCtion:SHAPe value [SOURce:]PM[1]:INTernal[1]:FUNCtion:SHAPe? **Remark:** SINE is supported.	🖐
[SOURce:]PM[1]:INTernal[1]:SWEep:TIME value [SOURce:]PM[1]:INTernal[1]:SWEep:TIME? **Remark:** Dwell time is assumed.	🖐
[SOURce:]PM[1]:INTernal[1]:SWEep:TRIGger value [SOURce:]PM[1]:INTernal[1]:SWEep:TRIGger? **Remark:** BUS, EXTernal, IMMediate are supported.	🖐
[SOURce:]PM[1]:SOURce value [SOURce:]PM[1]:SOURce? **Remark:** EXTernal[1], INTernal[1] are supported.	🖐
[SOURce:]FM[1]:STATe value [SOURce:]FM[1]:STATe?	✓
[SOURce:]POWer:ALC[:STATe] value [SOURce:]POWer:ALC[:STATe]?	✓
[SOURce:]POWer:ATTenuation value [SOURce:]POWer:ATTenuation?	✓
[SOURce:]POWer:ATTenuation:AUTO value [SOURce:]POWer:ATTenuation:AUTO?	✓
[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude] value [SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]?	✓

Device-specific functions	
Command syntax	**Status**
[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]:STEP[:INCRement] value [SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]:STEP[:INCRement]?	✓
[SOURce:]POWer[:LEVel][:IMMediate]:OFFSet value [SOURce:]POWer[:LEVel][:IMMediate]:OFFSet?	✓
[SOURce:]POWer:MODE value [SOURce:]POWer:MODE?	✓
[SOURce:]POWer:REFerence value [SOURce:]POWer:REFerence?	✓
[SOURce:]POWer:REFerence:STATe value [SOURce:]POWer:REFerence:STATe?	✓
[SOURce:]POWer:STARt value [SOURce:]POWer:STARt?	✓
[SOURce:]POWer:STOP value [SOURce:]POWer:STOP?	✓
[SOURce:]PULM:EXTernal:POLarity value [SOURce:]PULM:EXTernal:POLarity?	✓
[SOURce:]PULM:INTernal[1]:FREQuency value [SOURce:]PULM:INTernal[1]:FREQuency?	✓
[SOURce:]PULM:INTernal[1]:FREQuency:STEP[:INCRement] value [SOURce:]PULM:INTernal[1]:FREQuency:STEP[:INCRement]?	✓
[SOURce:]PULM:INTernal[1]:FUNCtion:SHAPe value [SOURce:]PULM:INTernal[1]:FUNCtion:SHAPe? **Remark:** SQUare is supported.	✋
[SOURce:]PULM:INTernal[1]:PERiod value [SOURce:]PULM:INTernal[1]:PERiod?	✓
[SOURce:]PULM:INTernal[1]:PERiod:STEP[:INCRement] value [SOURce:]PULM:INTernal[1]:PERiod:STEP[:INCRement]?	✓
[SOURce:]PULM:INTernal[1]:PWIDth value [SOURce:]PULM:INTernal[1]:PWIDth?	✓
[SOURce:]PULM:INTernal[1]:PWIDth:STEP[:INCRement] value [SOURce:]PULM:INTernal[1]:PWIDth:STEP[:INCRement]?	✓
[SOURce:]PULM:INTernal[1]:SOURce value [SOURce:]PULM:INTernal[1]:SOURce? **Remark:** EXTernal[1], INTernal are supported.	✋

Device-specific functions	
Command syntax	**Status**
[SOURce:]PULM:INTernal[1]:STATe value [SOURce:]PULM:INTernal[1]:STATe?	✓
[SOURce:]:ROSCillator:SOURce value [SOURce:]:ROSCillator:SOURce?	✓
[SOURce:]SWEep:DWELl value [SOURce:]SWEep:DWELl?	✓
[SOURce:]SWEep:POINts value [SOURce:]SWEep:POINts?	✓
STATus:OPERation:CONDition?	✓
STATus:OPERation:ENABle value STATus:OPERation:ENABle?	✓
STATus:OPERation[:EVENt]?	✓
STATus:OPERation:NTRansition value STATus:OPERation: NTRansition?	✓
STATus:OPERation:PTRansition value STATus:OPERation: PTRansition?	✓
STATus:PRESet	✓
STATus:QUEStionable:CONDition?	✓
STATus:QUEStionable:ENABle value STATus:QUEStionable:ENABle?	✓
STATus:QUEStionable[:EVENt]?	✓
STATus:QUEStionable:NTRansition value STATus:QUEStionable: NTRansition?	✓
STATus:QUEStionable:PTRansition value STATus:QUEStionable: PTRansition?	✓
SYSTem:CAPability?	◯
SYSTem:COMMunicate:GPIB:ADDRess value SYSTem:COMMunicate:GPIB:ADDRess?	✓
SYSTem:COMMunicate:GTLocal	✓
SYSTem:COMMunicate:LAN:CONFig value SYSTem:COMMunicate:LAN:CONFig?	✓
SYSTem:COMMunicate:LAN:GATeway value SYSTem:COMMunicate:LAN:GATeway?	✓
SYSTem:COMMunicate:LAN:HOSTname value SYSTem:COMMunicate:LAN:HOSTname?	✓

Device-specific functions	
Command syntax	**Status**
SYSTem:COMMunicate:LAN:IP value SYSTem:COMMunicate:LAN:IP?	✔
SYSTem:COMMunicate:LAN:SUBNet value SYSTem:COMMunicate:LAN:SUBNet?	✔
SYSTem:COMMunicate:SERial:BAUD value SYSTem:COMMunicate:SERial:BAUD?	✔
SYSTem:DATE value SYSTem:DATE?	✔
SYSTem:ERRor:CODE[:NEXT]? **Remark:** Returns the original error code of the Rohde & Schwarz signal generator.	✋
SYSTem:ERRor[:NEXT]? **Remark:** Returns the original error message of the Rohde & Schwarz signal generator.	✋
SYSTem:IDN value SYSTem:IDN?	✔
SYSTem:OPT value SYSTem:OPT?	✔
SYSTem:PDOWn	✔
SYSTem:PRESet	✔
SYSTem:PRESet:ALL	✔
SYSTem:SECurity:DISPlay value SYSTem:SECurity:DISPlay?	✔
SYSTem:TIME value SYSTem:TIME?	✔
SYSTem:VERSion?	✔
TRIGger[:SEQuence][:IMMediate]	✔
TRIGger[:SEQuence]:SLOPe value TRIGger[:SEQuence]:SLOPe?	✔
TRIGger[:SEQuence]:SOURce value TRIGger[:SEQuence]:SOURce? **Remark:** BUS, EXTernal, IMMediate are supported.	✋
UNIT:POWer value UNIT:POWer? **Remark:** DB, DBM, DBV, DBMV, DBUV, V, MV, UV are supported.	✋

7 Emulating the Hewlett-Packard 83620, 83630, 83640, 83650

The remote emulations HP83620, HP83630, HP83640, HP83650 are available only if one of the following options is installed:

- SMW-B112   RF Path 100 kHz to 12.75 GHz
- SMW-B120   RF Path 100 kHz to 20 GHz
- SMW-B131   RF Path 100 kHz to 31.8 GHz
- SMW-B140   RF Path 100 kHz to 40 GHz

The following tables show the current implementation status of each command. Commands not shown in these tables are not supported.

IEEE488.2 functions	
Command syntax	**Status**
*CLS	✔
*ESE value *ESE? **Remark:** Layout of ESE register corresponds to Rohde & Schwarz signal generator.	✋
*ESR? **Remark:** Layout of ESE register corresponds to Rohde & Schwarz signal generator.	✋
*IDN? **Remark:** If the response does not match the requirements, a user-specific response to *IDN? and *OPT? can be applied on the instrument's front panel.	✋
*OPC *OPC?	✔
*OPT? **Remark:** If the response does not match the requirements, a user-specific response to *IDN? and *OPT? can be applied on the instrument's front panel.	✋
*RCL value	✔
*RST	✔
*SAV value	✔
*SRE value *SRE? **Remark:** Layout of ESE register corresponds to Rohde & Schwarz signal generator.	✋

IEEE488.2 functions	
Command syntax	**Status**
*STB? **Remark:** Layout of ESE register corresponds to Rohde & Schwarz signal generator.	✋
*TRG	✓
*TST?	✓
*WAI	✓

Device-specific functions	
Command syntax	**Status**
ABORt	✓
AM[:DEPth] value AM[:DEPth]?	✓
AM:INTernal:FREQuency value AM:INTernal:FREQuency?	✓
AM:INTernal:FREQuency:STEP[:INCRement] value AM:INTernal:FREQuency:STEP[:INCRement]?	✓
AM:INTernal:FUNCtion value AM:INTernal:FUNCtion? **Remark:** RAMP is not supported.	✋
AM:MODE value AM:MODE? **Remark:** DEEP is not supported.	✋
AM:SOURce value AM:SOURce?	✓
AM:STATe value AM:STATe?	✓
AM:TYPE value AM:TYPE? **Remark:** EXPonential is not supported.	✋
DISPlay[:STATe] value DISPlay[:STATe]?	✓
FM:COUPling value FM:COUPling?	✓

Device-specific functions	
Command syntax	**Status**
FM[:DEViation] value FM[:DEViation]?	✓
FM:INTernal:FREQuency value FM:INTernal:FREQuency?	✓
FM:INTernal:FUNCtion value FM:INTernal:FUNCtion? **Remark:** RAMP is not supported.	🖑
FM:SOURce value FM:SOURce?	✓
FM:STATe value FM:STATe?	✓
FREQuency:CENTer value FREQuency:CENTer?	✓
FREQuency[:CW] value FREQuency[:CW]?	✓
FREQuency[:CW]:AUTO value FREQuency[:CW]:AUTO?	✓
FREQuency:MANual value FREQuency:MANual?	✓
FREQuency:MODE value FREQuency:MODE? **Remark:** CW and SWEep are supported.	🖑
FREQuency:MULTiplier value FREQuency:MULTiplier?	✓
FREQuency:OFFSet value FREQuency:OFFSet?	✓
FREQuency:SPAN value FREQuency:SPAN?	✓
FREQuency:STARt value FREQuency:STARt?	✓
FREQuency:STEP[:INCRement] value FREQuency:STEP[:INCRement]?	✓
FREQuency:STOP value FREQuency:STOP?	✓

Device-specific functions	
Command syntax	**Status**
INITiate:CONTinuous value INITiate:CONTinuous?	✓
INITiate[:IMMediate]	✓
MODulation:OUTPut:SOURce value MODulation:OUTPut:SOURce?	✓
MODulation:OUTPut:STATe value MODulation:OUTPut:STATe?	✓
MODulation:STATe value MODulation:STATe?	✓
OUTPut[:STATe] value OUTPut[:STATe]?	✓
POWer:CENTer value POWer:CENTer?	✓
POWer[:LEVel] value POWer[:LEVel]?	✓
POWer:MODE value POWer:MODE? **Remark:** CW and SWEep are supported.	🖐
POWer:OFFSet value POWer:OFFSet?	✓
POWer:SPAN value POWer:SPAN?	✓
POWer:STARt value POWer:STARt?	✓
POWer:STEP[:INCRement] value POWer:STEP[:INCRement]?	✓
POWer:STOP value POWer:STOP?	✓
PULSe:FREQuency value PULSe:FREQuency?	✓
PULSe:PERiod value PULSe:PERiod?	✓
PULSe:WIDTh value PULSe:WIDTh?	✓
PULM:EXTernal:POLarity value PULM:EXTernal:POLarity?	✓

Device-specific functions	
Command syntax	**Status**
PULM:INTernal:FREQuency value PULM:INTernal:FREQuency?	✓
PULM:INTernal:PERiod value PULM:INTernal:PERiod?	✓
PULM:INTernal:TRIGger:SOURce value PULM:INTernal:TRIGger:SOURce?	✓
PULM:INTernal:WIDTh value PULM:INTernal:WIDTh?	✓
PULM:SOURce value PULM:SOURce? **Remark:** SCALar is not supported.	🖐
PULM:STATe value PULM:STATe?	✓
STATus:OPERation:CONDition?	✓
STATus:OPERation:ENABle value STATus:OPERation:ENABle?	✓
STATus:OPERation[:EVENt]?	✓
STATus:OPERation:NTRansition value STATus:OPERation:NTRansition?	✓
STATus:OPERation:PTRansition value STATus:OPERation:PTRansition?	✓
STATus:PRESet	✓
STATus:QUEStionable:CONDition?	✓
STATus:QUEStionable:ENABle value STATus:QUEStionable:ENABle?	✓
STATus:QUEStionable[:EVENt]?	✓
STATus:QUEStionable:NTRansition value STATus:QUEStionable:NTRansition?	✓
STATus:QUEStionable:PTRansition value STATus:QUEStionable:PTRansition?	✓
SWEep:DWELl value SWEep:DWELl?	✓
SWEep:DWELl:AUTO value SWEep:DWELl:AUTO?	✓

Device-specific functions	
Command syntax	**Status**
SWEep:GENeration value SWEep:GENeration? **Remark:** ANALog is not supported.	🖐
SWEep:MODE value SWEep:MODE?	✔
SWEep:POINts value SWEep:POINts?	✔
SWEep:TIME value SWEep:TIME?	✔
SWEep:TRIGger:SOURce value SWEep:TRIGger:SOURce?	✔
SYSTem:COMMunicate:GPIB:ADDRess value SYSTem:COMMunicate:GPIB:ADDRess?	✔
SYSTem:ERRor? **Remark:** Returns the original error code of the Rohde & Schwarz signal generator.	🖐
SYSTem:LANGuage value SYSTem:LANGuage? **Remark:** "SCPI" and "EXIT" are supported.	🖐
SYSTem:PRESet	✔
SYSTem:VERSion?	✔
TRIGger[:IMMediate]	✔
TRIGger:SOURce value TRIGger:SOURce?	✔
TRIGger[:IMMediate]	✔
TSWeep	✔
UNIT:POWer value UNIT:POWer?	✔

8 References

[1]   Rohde & Schwarz, R&S®SMW200A Vector Signal Generator User Manual

9 Ordering Information

Please visit the product websites at www.rohde-schwarz.com for comprehensive ordering information ("Options") on the following Rohde & Schwarz instruments:

- R&S®SMW200A vector signal generator

Rohde & Schwarz

The Rohde & Schwarz electronics group offers innovative solutions in the following business fields: test and measurement, broadcast and media, secure communications, cybersecurity, radiomonitoring and radiolocation. Founded more than 80 years ago, this independent company has an extensive sales and service network and is present in more than 70 countries.

The electronics group is among the world market leaders in its established business fields. The company is headquartered in Munich, Germany. It also has regional headquarters in Singapore and Columbia, Maryland, USA, to manage its operations in these regions.

Regional contact

Europe, Africa, Middle East
+49 89 4129 12345
customersupport@rohde-schwarz.com

North America
1 888 TEST RSA (1 888 837 87 72)
customer.support@rsa.rohde-schwarz.com

Latin America
+1 410 910 79 88
customersupport.la@rohde-schwarz.com

Asia Pacific
+65 65 13 04 88
customersupport.asia@rohde-schwarz.com

China
+86 800 810 82 28 |+86 400 650 58 96
customersupport.china@rohde-schwarz.com

Sustainable product design

▪ Environmental compatibility and eco-footprint

▪ Energy efficiency and low emissions

▪ Longevity and optimized total cost of ownership

Certified Quality Management
ISO 9001

Certified Environmental Management
ISO 14001

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.