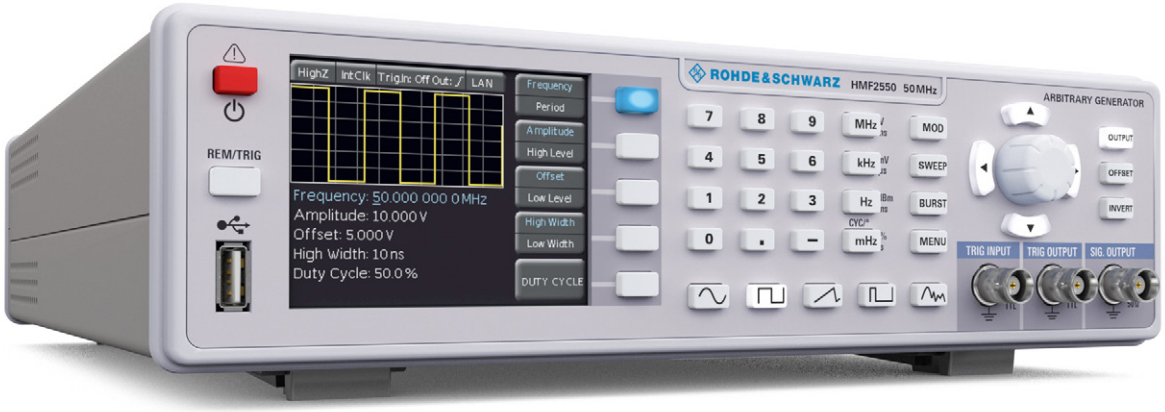
R&S®HMF2525
R&S®HMF2550
Arbitrary Function Generator
SCPI Programmers Manual



5800571802

Test & Measurement

SCPI Programmers Manual

Version 01

ROHDE&SCHWARZ

Content

1 Basics

This chapter provides basic information on operating an instrument via remote control.

1.1 Remote Control Interfaces

For remote control, RS-232 / USB (standard interface HO720), Ethernet / USB (optional interface HO730) or GPIB (optional interface HO740) can be used. The optional interfaces replaces the RS-232 / USB HO720 interface module on the rear panel.

NOTICE

Within this interface description, the term GPIB is used as a synonym for the IEC/IEEE bus interface.

SCPI (Standard Commands for Programmable Instruments) SCPI commands - messages - are used for remote control. Commands that are not taken from the SCPI standard follow the SCPI syntax rules.

NOTICE

The end character must be set to linefeed (LF) or carriage return + linefeed (CR-LF).

1.1.1 RS-232 Interface

If you use classic RS-232 you do not need any driver. In order to set the interface parameter at the HMF, please press the button MENU at the front panel and choose the menu item INTERFACE. Make sure the RS-232 interface is chosen. The menu item PARAMETER opens a menu where you can set and save all parameter for the RS-232 communication. Setting of the RS-232 must fit the setting of the corresponding PC COM port.

In general, there are exist two options for the RS-232 communication: with or without interface handshake. If you are working without handshake you have to integrate appropriate delays between the commands to make sure that all commands are executed correctly (approx. 50ms to 100ms). Without handshake you can have the problem that the interface buffer can overflow (e.g. missing commands). If you are working with interface handshake (set on both sides, HMF and PC) you don‘t need to integrate delays.

In the interface settings of the HMF you can set the interface handshake (button MENU). If you have a look into the menu item INTERFACE you can choose the menu item HANDSHAKE. There you can find the items NONE or CTS/RTS. CTS/RTS means the activation of the interface handshake. Please note that you choose the same settings in your appropriate software.

NOTICE

If you want to use the RS-232 interface we recommend to activate the interface handshake to avoid timing problems.

1.1.2 USB Interface

If you are using USB you need to install a USB driver which is available on the Rohde & Schwarz Website.

NOTICE

All descriptions regarding the USB interface are true for the HO720 interface card as well as for the optional HO730 USB part. All currently available USB driver are fully tested, functional and released for Windows XP™, Windows Vista™, Windows 7™ or Windows 8™, both as 32Bit or 64Bit versions.

The HMP USB interface has to be chosen in the MENU and does not need any setting.

NOTICE

If the virtual COM port will be used, you have to install the virtual COM port part of the HO720 / HO730 USB driver. The virtual COM port (VCP) will be activated in the PC device explorer.

1.1.3 Ethernet (LAN) Interface

The settings of the parameter will be done after selecting the menu item Ethernet. You can set a fix IP adress or a dynamic IP setting via the DHCP function. Please ask your IT department for the correct setting at your network.

IP address
To set up the connection the IP address of the instrument is required. It is part of the resource string used by the program to identify and control the instrument. The resource string has the form:

TCPIP::‹IP_address›::‹IP_port›::INSTR

The default port number for SCPI socket communication is 5025. IP address and port number are listed In the „Ethernet Settings“ of the HMP series, see also: chapter 1.2.2,“Configuring LAN Parameters“.

Example (HO732): If the instrument has the IP address 192.1.2.3; the valid resource string is

TCPIP0::192.1.2.3::inst0::INSTR

If the LAN is supported by a DNS server, the host name can be used instead of the IP address. The DNS server (Domain Name System server) translates the host name to the IP address. The resource string has the form:

TCPIP::‹host_name›::‹IP_port›::INSTR

1.1.4 GPIB Interface (IEC/IEEE Bus Interface)

An optional GPIB interface is also available for the HMF series. This solution is particularly attractive for customers who already have an existing GPIB environment. With minimum efforts, an old instrument can be replaced by a model of the HMF series.

To be able to control the instrument via the GPIB bus, the instrument and the controller must be linked by a GPIB bus cable. A GPIB bus card, the card drivers and the program libraries for the programming language must be provided by the controller. The controller addresses the instrument with the GPIB instrument address.

Characteristics
The GPIB interface is described by the following characteristics:

- Up to 15 instruments can be connected
- The total cable length is restricted to a maximum of 15m; the cable length between two instruments should not exceed 2m.
- A wired „OR“-connection is used if several instruments are connected in parallel.

GPIB Instrument Address
In order to operate the instrument via remote control, it must be addressed using the GPIB address. The remote control address is factory-set to 20, but it can be changed in the network environment settings or in the HMF MENU under INTERFACE -> PARAMETER. For remote control, a GPIB address from 0 to 30 are allowed. The GPIB address is maintained after a reset of the instrument settings.

1.2 Setting Up a Network (LAN) Connection

1.2.1 Connecting the Instrument to the Network

NOTICE

Risk of network failure
Before connecting the instrument to the network or configuring the network, consult your network administrator. Errors may affect the entire network.

The network card can be operated with a 10 Mbps Ethernet IEEE 802.3 or a 100 Mbps Ethernet IEEE 802.3u interface.

NOTICE

To establish a network connection, connect a commercial RJ-45 cable to one of the LAN ports of the instrument and to a PC.

1.2.2 Configuring LAN Parameters

Depending on the network capacities, the TCP/IP address information for the instrument can be obtained in different ways.

- If the network supports dynamic TCP/IP configuration using the Dynamic Host Configuration Protocol (DHCP), and a DHCP server is available, all address information can be assigned automatically.
- Otherwise, the address must be set manually. Automatic Private IP Addressing (APIPA) is not supported.

By default, the instrument is configured to use dynamic TCP/IP configuration and obtain all address information automatically. This means that it is safe to establish a physical connection to the LAN without any previous instrument configuration.

NOTICE

Risk of network errors
Connection errors can affect the entire network. If your network does not support DHCP, or if you choose to disable dynamic TCP/IP configuration, you must assign valid address information before connecting the instrument to the LAN. Contact your network administrator to obtain a valid IP address.

Configuring LAN parameters

- Press the MENU key and choose INTERFACE.
- Press the knob, choose ETHERNET (SELECT INTERFACE) and PARAMETER.
- Define the IP and port settings of the instrument.
- Select the IP Port - the port number for SCPI socket communication.

NOTICE

By default the instrument is not set to DHCP. If the instrument is set to DHCP and cannot find a DHCP server, it takes about two minutes until the Ethernet menu is available.

Checking LAN and SCPI connection

- Check the LAN connection using ping: `ping xxx.yyy.zzz.xxx`.
- If the PC can access the instrument, enter the IP address into your PC internet browser:
  http//:xxx.yyy.zzz.xxx
  The „Instrument Home“ page appears. It provides information of the instrument and the LAN connection.

1.3 Switching to Remote Control

When you switch on the instrument, it is always in manual operation state („local" state) and can be operated via the front panel. When you send a command via PC, it will be received and executed by the instrument. The display remains on, manual operation via the front panel is always possible.

1.4 Messages and Command Structure

1.4.1 Messages

Instrument messages are employed in the same way for all interfaces, if not indicated otherwise in the description.
See also:
- Structure and syntax of the instrument messages: chapter 1.4.2, „SCPI Command Structure".
- Detailed description of all messages: chapter 2, „Command Reference".

There are different types of instrument messages:
- Commands
- Instrument responses

Commands
Commands (program messages) are messages which the controller sends to the instrument. They operate the instrument functions and request information. The commands are subdivided according to two criteria:

According to the instrument effect:
- Setting commands cause instrument settings such as a reset of the instrument or setting the frequency.
- Queries cause data to be provided for remote control, e.g. for identification of the instrument or polling a parameter value. Queries are formed by appending a question mark to the command header.

According to their definition in standards:
- The function and syntax of the Common commands are precisely defined in standard IEEE 488.2. They are employed identically on all instruments (if implemented). They refer to functions such as management of the standardized status registers, reset and self test.
- Instrument control commands refer to functions depending on the features of the instrument such as voltage settings. Many of these commands have also been standardized by the SCPI committee. These commands are marked as „SCPI compliant" in the command reference chapters. Commands without this SCPI label are device-specific, however, their syntax follows SCPI rules as permitted by the standard.

Instrument responses
Instrument responses (response messages and service requests) are messages which the instrument is sent to the controller after a query. They can contain measurement results, instrument settings and information on the instrument status.

GPIB Interface Messages

Interface messages are transmitted to the instrument on the data lines with the attention line (ATN) being active (LOW). They are used for communication between the controller and the instrument and can only be sent by a PC which has the function of a GPIB bus controller. GPIB interface messages can be further subdivided into:

- **Universal commands:** act on all instruments connected to the GPIB bus without previous addressing; universal commands are encoded in the range 10 through 1F hex. They affect all instruments connected to the bus and do not require addressing.
- **Addressed commands:** only act on instruments previously addressed as listeners; addressed commands are encoded in the range 00 through 0F hex. They only affect instruments addressed as listeners.

1.4.2 SCPI Command Structure

SCPI commands consist of a so-called header and, in most cases, one or more parameters. The header and the parameters are separated by a „white space". The headers may 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.

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 content of the event status register.
***IDN?**	Identification Query	Queries the instrument identification string.

Table 1.4: Examples of Common Commands

Long and short form

The mnemonics feature a long form and a short form. The short form is marked by upper case letters, the long form corresponds to the complete word. Either the short form or the long form can be entered; other abbreviations are not permitted.

Example: VOLTage:OFFSet? is equivalent to VOLT:OFF?

NOTICE

Case-insensitivity

Upper case and lower case notation only serves to distinguish the two forms in the manual, the instrument itself is case-insensitive.

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 the used parameter. **Example:** FUNCtion {SINusoid | SQUare | RAMP | PULSe | ARBitrary} FUNC SIN sets the sine wave function. FUNC SQU sets the square wave function.
{ }	Parameters in curly brackets are optional and can be inserted once or several times, or omitted. **Example:** PERiod {<Period in sec> | MINimum | MAXimum} The following are valid commands: PER MIN PER MAX

Table 1.5: Special characters

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 required parameters and the allowed value range are specified in the command description.

Numeric values

Numeric values can be entered in the following form. Values exceeding the resolution of the instrument are rounded up or down.

Example:

VOLT 5V = VOLT 5
VOLT 800mV = VOLT 0.8

Special numeric values
The text 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.

Example:
VOLT MAX
VOLT MAX?, Response: 2.00000E+01

Queries for special numeric values
The numeric values associated to MAXimum/MINimum can be queried by adding the corresponding mnemonics to the command. They must be entered following the quotation mark.

Example:
VOLT? MAX
Returns the maximum numeric value.

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:
OUTPut ON
OUTPut?, Response: ON

Overview of Syntax Elements
The following table provides an overview of the syntax elements:

:	The colon separates the mnemonics of a command.
,	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.
	A „white space" (ASCII-Code 0 to 9, 11 to 32 decimal, e.g. blank) separates the header from the parameters.

Table 1.6: Syntax Elements

Responses to Queries
A query is defined for each setting command. 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:
VOLTage:UNIT VOLT
VOLTage:UNIT?, Response: VOLT

1.5 Command Sequence and Synchronization

A sequential command finishes the execution before the next command is starting. In order to make sure that commands are actually carried out in a certain order, each command must be sent in a separate command line.

NOTICE

As a general rule, send commands and queries in different program messages.

1.5.1 Preventing Overlapping Execution

Command	Action	Programming the controller
***OPC**	Sets the Operation Complete bit in the ESR after all previous commands have been executed.	• Setting bit 0 in the ESE • Setting bit 5 in the SRE • Waiting for service request (SRQ)
***OPC?**	Stops command processing until 1 is returned. This is only the case after the Operation Complete bit has been set in the ESR. This bit indicates that the previous setting has been completed.	Sending *OPC? directly after the command whose processing should be terminated before other commands can be executed.
***WAI**	Stops further command processing until all commands have been executed before *WAI.	Sending *WAI directly after the command whose processing should be terminated before other commands are executed

Table 1.7: Synchronization using *OPC, *OPC? and *WAI

To prevent an overlapping execution of commands the commands *OPC, *OPC? or *WAI can be used. All three commands cause a certain action only to be carried out after the hardware has been set. The controller can be forced to wait for the corresponding action.

NOTICE

The HMF series does not support parallel processing of remote commands. If OPC? returns a „1“, the device is able to process new commands.

1.6 Contents of the Status Registers

The SCPI standard contains an event handling system for all available interfaces that can be used to be informed about the processes within the oscilloscope. According to the standard the oscilloscope replies only after receiving a query but the event handling enables the device to inform the user that an extraordinary event took place.

Event Status Register (ESR) and Event Status Enable Register (ESE)
The ESR is defined in IEEE 488.2. The event status register can be read out using command *ESR?. If a bit is set in the ESE and the associated bit in the ESR changes from 0 to 1, the ESB bit in the STB is set. The ESE register can be set using the command *ESE and read using the command *ESE?.

Status Byte (STB) and Service Request Enable Register (SRE)
The STatus Byte (STB) is already defined in IEEE 488.2. It provides a rough overview of the instrument status by collecting the pieces of information of the lower registers. A special feature is that bit 6 acts as the sum bit of the remaining bits of the status byte. The STB is read using the command *STB or a serial poll.

The STatus Byte (STB) is linked to the Service Request Enable (SRE) register. 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. The SRE can be set using the command *SRE and read using the command *SRE?.

The SRER defines which interfaces may ask for permission to transmit. The GPIB interface is the only one which has a hardware connection to the user for the transmission request (RQS bit); the status of this line will tell whether an event happened.

All other interfaces (RS-232, USB, Ethernet) do not support this. If the user want to use the event handling feature, he will have to read the status, e.g. by polling, from the instrument in every case. Eventually, the status byte SBR will yield the desired information.

Error Queue
Each error state in the instrument leads to an entry in the error queue. The entries of the error queue are detailed plain text error messages that can be looked up in the error log or queried via remote control using SYSTem:ERRor[:NEXT]?. Each call of SYSTem:ERRor[:NEXT]? provides one entry from the error queue. If no error messages are stored, the instrument responds with 0, „No error“.

For further description of the error queue and the device error codes, please refer to chapter 2.

2 Command Reference

This chapter provides the description of all remote commands available for HMC8012. The commands are sorted according to the menu structure of the instrument. A list of commands in alphabetical order ist given in the „List of Commands“ at the end of this documentation.

2.1 Common Commands

Common commands are described in the IEEE 488.2 (IEC 625-2) standard. These commands have the same effect and are employed in the same way on different devices.The headers of these commands consist of ” * “ followed by three letters. Many common commands are related to the Status Reporting System.

Available common commands:

***CLS**
CLear Status

Sets the status byte (STB), the standard event register (ESR) and the EVENt part of the QUEStionable to zero. The command does not alter the mask and transition parts of the registers. It clears the output buffer.

Usage: Setting only

***ESE <Value>**
Event Status Enable

Sets the event status enable register to the specified value. The query *ESE? returns the contents of the event status enable register in decimal form.

Parameters:
<Value> Range: 0 to 255

***ESR?**
Event Status Read

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

Return values:
<Contents> Range: 0 to 255

Usage: Query only

***IDN?**
IDeNtification

Returns the instrument identification string.

Return values:
<ID> HAMEG,‹device type›,‹serial number›,‹firmwareversion›

Example: HAMEG,HMF2550,011299686,HW90000000,SW02.300

Usage: Query only

***OPC**
OPeration Complete

Sets bit 0 in the event status register when all preceding commands have been executed. This bit can be used to initiate a service request. The query *OPC? writes a „1“ into the output buffer as soon as all preceding commands have been executed. This is used for command synchronization.

NOTICE

The HMF series does not support parallel processing of remote commands. If OPC? returns a „1“, the device is able to process new commands.

***RST**
ReSeT

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

Usage: Setting only

NOTICE

We recommend to start a program by *RST in order to set the instrument to a defined status prior to starting a program.

***SRE <Contents>**
Service Request Enable

Sets the service request enable register to the indicated value. This command determines under which conditions a service request is triggered. The query *SRE? returns a decimal value of the service request enable register which corresponds to the binary-weighted sum of all bits.

Parameters:
<Contents> Contents of the service request enable register in decimal form.
Bit 6 (MSS mask bit) is always 0.

Range: 0 to 255

NOTICE

The SRE is an enable register. Consequently, there are no denotations about the bits. This register conduce for the „OR" combination of the bits in the status byte.

***STB?**
STatus Byte query

Returns the contents of the status byte in decimal form.

Usage: Query only

***WAI**
WAIt to continue

Prevents servicing of the subsequent commands until all preceding commands have been executed.

Usage: Event

2.2 System related commands

SYSTem:NAME
Defines an instrument name.

Parameters:
<Name> String with max. 20 characters

Example: SYST:NAME “MyHMF“

SYSTem:NAME?
Queries the defined instrument name.

Return values: String with max. 20 characters

SYSTem:DATE <Year>,<Month>,<Day>
Specifies the internal date for the instrument.

Parameters:
<Year> Default unit: a

<Month> Range: 1 to 12

<Day> Range: 1 to 31
Default unit: d

Example: SYSTem:DATE 2014,10,1
Sets the device date to october 1st in the year 2014
SYSTem:DATE?
Returns 2014,10,1

Usage: SCPI confirmed

SYSTem:DATE?
Queries the system date.

Return values: e.g. 2015,9,3

SYSTem:TIME <Hour>,<Minute>,<Second>
Specifies the internal time for the instrument.

Parameters:
<Hour> Range: 0 to 23
Default unit: h

<Minute> Range: 0 to 59
Default unit: min

<Second> Range: 0 to 59
Default unit: s

Example: SYSTem:TIME 12,15,0
Sets the time to quarter past twelve.
SYSTem:TIME?
Returns 12,15,0

Usage: SCPI confirmed

SYSTem:TIME?
Queries the instrument time.

Return values: e.g. 14,46,45

SYSTem:HARDware?
Queries the hardware-ID of the instrument.

Usage: Query only

SYSTem:SOFTware?
Queries the software revision oft he instrument.

Usage: Query only

SYSTem:SNUM?
Queries the serial number of the instrument.

Usage: Query only

SYSTem:TREE?
Returns a list of implemented remote commands.

Usage: Query only

SYSTem:ELISt?
Queries the error list.

Usage: Query only

SYSTem:ERRor[:NEXT]?
Queries an error and removes it from the queue. Positive error numbers are instrument-dependent. Negative error numbers are reserved by the SCPI standard. If the queue is empty the response is 0, "No error".

Return values:
<error> **0**, "No error"
-100, "Command error"
-102, "Syntax error"
-109, "Missing parameter
-200, "Execution error"
-350, "Queue overflow"

Usage: Query only

SYSTem:ERRor:ALL? <Error>
Queries the error/event queue for all unread items and removes them from the queue.
The response is a comma separated list of error number and a short description of the error in FIFO order. Positive error numbers are instrument-dependent. Negative error numbers are reserved by the SCPI standard.

Return values:
<Error> List of: Error/event_number,"Error/event_description>[;Devicedependent info]"
If the queue is empty, the response is 0,"No error"

Usage: Query only
SCPI confirmed

SYSTem:BEEPer:ERRor {ON | OFF | 0 | 1}
Activates or deactivates the error beeper function.

Parameters:
ON | 1: Error beeper function will be activated
OFF | 0: Error beeper function will be deactivated

*RST: ON | 1

SYSTem:BEEPer:ERRor?
Queries the state of the error beeper function.

Return values:
ON: Error beeper function will be activated
OFF: Error beeper function will be deactivated

SYSTem:BEEPer:CONTrol {ON | OFF | 0 | 1}
Activates or deactivates the control beeper function.

Parameters:
ON | 1: Control beeper function will be activated
OFF | 0: Control beeper function will be deactivated

*RST: ON | 1

SYSTem:BEEPer:CONTrol?
Queries the state of the control beeper function.

Return values:
ON: Control beeper function will be activated
OFF: Control beeper function will be deactivated

SYSTem:REFerence <Reference>
Sets the system reference clock (10MHz) to internal or external source.
Parameters:
<Reference> IMMediate | EXTernal

IMMediate
Selects the internal 10MHz reference frequency.

EXTernal
Selects the external reference frequency. The external 10MHz reference frequency signal must comply with the specifications given with respect to frequency accuracy and amplitude.

*RST: IMM

SYSTem:REFerence?
Queries the type of the system reference.

Return values: IMM | EXT

2.3 Basic Display Settings

DISPlay:TRACe {OFF | ON | 0 | 1}
Activates or deactivates the trace display.

Parameters:
ON | 1: Trace display will be activated.
OFF | 0: Trace display will be deactivated

*RST: ON | 1

DISPlay:TRACe?
Queries the state of the trace display.

Return values: ON | OFF

ON: Trace display is activated.
OFF: Trace display is deactivated.

DISPlay:TRACe:INTensity {<Value in percent> | MINimum | MAXimum}
Defines the trace intensity in the diagram. *RST does not change the intensity.

Parameters:
<Value in percent> Range: 0 to 100

MIN: 0
MAX: 100

DISPlay:TRACe:INTensity?
Queries the current setting of the trace intensity.

Return values: 0...100 (value in %)

DISPlay:GRID {<Value in percent> | MINimum | MAXimum}
Defines the display intensity of the grid. *RST does not change the intensity.

Parameters:
<Value in percent> Range: 0 to 100

MIN: 0
MAX: 100

DISPlay:GRID?
Queries the current setting of the grid intensity.

Return values: 0...100 (value in %)

DISPlay:TRANsparancy {<Value in percent> | MINimum | MAXimum}
Defines the display transparancy.

Parameters:
<Value in percent> Range: 0 to 100

MIN: 0
MAX: 100

DISPlay:TRANsparancy? [MINimum | MAXimum]
Queries the current setting of the display transparancy.

Return values: 0...100 (value in %)

LED:BRIGhtness {HIGH | LOW}
Defines the LED brightness of the instrument frontside buttons.

Parameters:
<Brightness> HIGH | LOW

HIGH: High LED brightness.
LOW: Low LED brightness.

LED:BRIGhtness?
Queries the current setting of the LED brightness.

Return values: HIGH | LOW

2.4 Configuration Commands

2.4.1 General Output Configurations

FUNCtion {SINusoid | SQUare | RAMP | PULSe | ARBitrary}
Selects the output function with the selected frequency, amplitude and offset setting.

Parameters:

<Function>	**SINusoid**:	Sine wave function.
	SQUare:	Square wave function.
	RAMP:	Ramp function.
	PULSe:	Pulse function.
	ARBitrary:	Arbitrary function.

*RST: SIN

FUNCtion?
Queries the selected output function.

Return values: SIN | SQU | RAMP | PULS | ARB

OUTPut {OFF | ON | 0 | 1}
Activates or deactivates the instrument output.

Parameters:

<State>	**ON	1**:	The instrument output will be activated.
	**OFF	0**:	The instrument output will be disabled.

*RST: OFF | 0

OUTPut?
Queries the state of the instrument output.

Return values:	**ON**:	The instrument output is activated.
	OFF:	The instrument output is disabled.

OUTPut:LOAD {<Impedance in Ohm> | TERMinated | INFinity}
Defines the instrument output load.

Parameters:
<Load> Numeric value in Ohm.

TERMinated: 50Ω termination.
INFinity: Open circuit (without any load).

*RST: INF

OUTPut:LOAD?
Queries the type of the instrument output load.

Return values: Numeric value in **Ω** | INF

OUTPut:POLarity {NORMal | INVerted}
Defines the instrument output polarity.

Parameter:
<Type> NORMal | INVerted

NORMal: Default instrument output polarity.
INVerted: The instrument output polarity will be inverted.

NOTICE

The OUTPut:POLarity command is only valid for the pulse and arbitrary function.

OUTPut:POLarity?
Queries the type of the instrument output polarity.

Return values: NORM | INV

OUTPut:OFFSet {OFF | ON | 0 | 1}
Activates or deactivates the output offset voltage.

Parameters:
<State> **ON | 1**: The offset voltage will be activated.
OFF | 0: The offset voltage will be disabled.

*RST: OFF | 0

OUTPut:OFFSet?
Queries the state of the output offset voltage.

Return values: **ON**: The offset voltage is activated.
OFF: The offset voltage is disabled.

2.4.2 Sine Wave Configuration

FREQuency {<Frequency> | MINimum | MAXimum}
Defines the output frequency.

Parameters:
<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+07

FREQuency? [MINimum | MAXimum]
Queries the output frequency.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+07

PERiod {<Period in sec> | MINimum | MAXimum}
Defines the output perid.

Parameters:
<Period> Numeric value in sec.

MIN: 2.00E-08
MAX: 1.000000E+05

PERiod? [MINimum | MAXimum]
Queries the output period.

Return values: Numeric value in sec.

MIN: 2.00E-08
MAX: 1.000000E+05

VOLTage:UNIT {VOLT | DBM}
Sets the amplitude unit.

Parameters:
<Unit> VOLT | DBM

VOLT: Amplitude unit setting in V (Volt).
DBM: Amplitude unit setting in dBm.

VOLTage:UNIT?
Queries the amplitude unit type.

Return values: VOLT | DBM

VOLTage {<Amplitude> | MINimum | MAXimum}
Selects the amplitude voltage value.

Parameters:
<Amplitude> Numeric value in V (Volt).

MIN: 1.00E-02
MAX: 2.00000E+01

*RST: 1.0000E+00

VOLTage? [MINimum | MAXimum]
Queries the amplitude voltage value.

Return values: Numeric value in V (Volt).

MIN: 1.00E-02
MAX: 2.00000E+01

VOLTage:HIGH {<Voltage> | MINimum | MAXimum}
Defines the high level voltage depending on the amplitude setting.

Parameters:
<Voltage> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: 2.5000E+00

VOLTage:HIGH? [MINimum | MAXimum]
Queries the high level voltage.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

VOLTage:LOW {<Voltage> | MINimum | MAXimum}
Defines the low level voltage depending on the amplitude setting.

Parameters:
<Voltage> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: - 2.5000E+00

VOLTage:LOW? [MINimum | MAXimum]
Queries the low level voltage.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

VOLTage:OFFSet {<Offset> | MINimum | MAXimum}
Defines the output offset value depending on the amplitude setting.

Parameters:
<Offset> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: 0.0E+00

VOLTage:OFFSet? [MINimum | MAXimum]
Queries the output offset value.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

2.4.3 Square Wave Configuration

FREQuency {<Frequency> | MINimum | MAXimum}
Defines the output frequency.

Parameters:
<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+07

FREQuency? [MINimum | MAXimum]
Queries the output frequency.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+07

PERiod {<Period in sec> | MINimum | MAXimum}
Defines the output perid.

Parameters:
<Period> Numeric value in sec.

MIN: 2.00E-08
MAX: 1.000000E+05

PERiod? [MINimum | MAXimum]
Queries the output period.

Return values: Numeric value in sec.

MIN: 2.00E-08
MAX: 1.000000E+05

VOLTage:UNIT {VOLT | DBM}
Sets the amplitude unit.

Parameters:
<Unit> VOLT | DBM

VOLT: Amplitude unit setting in V (Volt).
DBM: Amplitude unit setting in dBm.

VOLTage:UNIT?
Queries the amplitude unit type.

Return values: VOLT | DBM

VOLTage {<Amplitude> | MINimum | MAXimum}
Selects the amplitude voltage value.

Parameters:
<Amplitude> Numeric value in V (Volt).

MIN: 1.00E-02
MAX: 2.00000E+01

*RST: 1.0000E+00

VOLTage? [MINimum | MAXimum]
Queries the amplitude voltage value.

Return values: Numeric value in V (Volt).

MIN: 1.00E-02
MAX: 2.00000E+01

VOLTage:HIGH {<Voltage> | MINimum | MAXimum}
Defines the high level voltage depending on the amplitude setting.

Parameters:
<Voltage> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: 2.5000E+00

VOLTage:HIGH? [MINimum | MAXimum]
Queries the high level voltage.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

VOLTage:LOW {<Voltage> | MINimum | MAXimum}
Defines the low level voltage depending on the amplitude setting.

Parameters:
<Voltage> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: - 2.5000E+00

VOLTage:LOW? [MINimum | MAXimum]
Queries the low level voltage.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

VOLTage:OFFSet {<Offset> | MINimum | MAXimum}
Defines the output offset value depending on the amplitude setting.

Parameters:
<Offset> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: 0.0E+00

VOLTage:OFFSet? [MINimum | MAXimum]
Queries the output offset value.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

FUNCtion:SQUare:DCYCle {<Percent> | MINimum | MAXimum}
Sets the duty cycle of the square wave function.

Parameters:
<Percent> Numeric value depending on the frequency settings.

MIN: 2.000E+01
MAX: 8.000E+01

FUNCtion:SQUare:DCYCle? [MINimum | MAXimum]
Queries the duty cycle of the square wave function.

Return values: Numeric value depending on the frequency settings.

MIN: 2.000E+01
MAX: 8.000E+01

FUNCtion:SQUare:WIDTh:HIGH {<Time> | MINimum | MAXimum}
Defines the high width of the square wave function depending on the frequency setting.

Parameters:
<Time> Numeric value in sec depending on the frequency setting.

MIN: 1.00E-08
MAX: 8.000000E+04

*RST: 1.00000E-05

FUNCtion:SQUare:WIDTh:HIGH? [MINimum | MAXimum]
Queries the high width of the square wave function depending on the frequency setting.

Return values: Numeric value in sec depending on the frequency setting.

MIN: 1.00E-08
MAX: 8.000000E+04

FUNCtion:SQUare:WIDTh:LOW {<Time> | MINimum | MAXimum}
Defines the low width of the square wave function depending on the frequency setting.

Parameters:
<Time> Numeric value in sec depending on the frequency setting.

MIN: 1.00E-08
MAX: 8.000000E+04

*RST: 1.00000E-05

FUNCtion:SQUare:WIDTh:LOW? [MINimum | MAXimum]
Queries the low width of the square wave function depending on the frequency setting.

Return values: Numeric value in sec depending on the frequency setting.

MIN: 1.00E-08
MAX: 8.000000E+04

2.4.3 Ramp Configuration

FREQuency {<Frequency> | MINimum | MAXimum}
Defines the output frequency.

Parameters:
<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 1.000000E+07

FREQuency? [MINimum | MAXimum]
Queries the output frequency.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 1.000000E+07

PERiod {<Period in sec> | MINimum | MAXimum}
Defines the output perid.

Parameters:
<Period> Numeric value in sec.

MIN: 1.000E-07
MAX: 1.000000E+05

PERiod? [MINimum | MAXimum]
Queries the output period.

Return values: Numeric value in sec.

MIN: 1.000E-07
MAX: 1.000000E+05

VOLTage:UNIT {VOLT | DBM}
Sets the amplitude unit.

Parameters:
<Unit> VOLT | DBM

VOLT: Amplitude unit setting in V (Volt).
DBM: Amplitude unit setting in dBm.

VOLTage:UNIT?
Queries the amplitude unit type.

Return values: VOLT | DBM

VOLTage {<Amplitude> | MINimum | MAXimum}
Selects the amplitude voltage value.

Parameters:
<Amplitude> Numeric value in V (Volt).

MIN: 1.00E-02
MAX: 2.00000E+01

*RST: 1.0000E+00

VOLTage? [MINimum | MAXimum]
Queries the amplitude voltage value.

Return values: Numeric value in V (Volt).

MIN: 1.00E-02
MAX: 2.00000E+01

VOLTage:HIGH {<Voltage> | MINimum | MAXimum}
Defines the high level voltage depending on the amplitude setting.

Parameters:
<Voltage> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: 2.5000E+00

VOLTage:HIGH? [MINimum | MAXimum]
Queries the high level voltage.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

VOLTage:LOW {<Voltage> | MINimum | MAXimum}
Defines the low level voltage depending on the amplitude setting.

Parameters:
<Voltage> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: - 2.5000E+00

VOLTage:LOW? [MINimum | MAXimum]
Queries the low level voltage.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

VOLTage:OFFSet {<Offset> | MINimum | MAXimum}
Defines the output offset value depending on the amplitude setting.

Parameters:
<Offset> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: 0.0E+00

VOLTage:OFFSet? [MINimum | MAXimum]
Queries the output offset value.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

FUNCtion:RAMP:SYMMetry {<Percent> | MINimum | MAXimum}
Defines the symmetrie of the ramp function.

Parameters:
<Percent> Numeric value in percent.

MIN: 1.00E+00
MAX: 1.0000E+02

*RST: 5.000E+01

FUNCtion:RAMP:SYMMetry? [MINimum | MAXimum]
Queries the symmetrie of the ramp function.

Return values: Numeric value in percent.

MIN: 1.00E+00
MAX: 1.0000E+02

FUNCtion:RAMP:TIME:RISE {<Time> | MINimum | MAXimum}
Defines the rise time of the ramp function depending on the frequency setting.

Parameters:
<Time> Numeric value in percent.

MIN: 8.0E-09
MAX: 1.000000E+05

*RST: 1.00000E-05

FUNCtion:RAMP:TIME:RISE? [MINimum | MAXimum]
Queries the rise time of the ramp function depending on the frequency setting.

Parameters:
<Time> Numeric value in percent.

MIN: 8.0E-09
MAX: 1.000000E+05

*RST: 1.00000E-05

FUNCtion:RAMP:TIME:FALL {<Time> | MINimum | MAXimum}
Defines the fall time of the ramp function depending on the frequency setting.

Parameters:
<Time> Numeric value in percent.

MIN: 8.0E-09
MAX: 1.000000E+05

*RST: 1.00000E-05

FUNCtion:RAMP:TIME:FALL? [MINimum | MAXimum]
Queries the fall time of the ramp function depending on the frequency setting.

Parameters:
<Time> Numeric value in percent.

MIN: 8.0E-09
MAX: 1.000000E+05

*RST: 1.00000E-05

2.4.4 Pulse Configuration

FREQuency {<Frequency> | MINimum | MAXimum}
Defines the output frequency.

Parameters:

<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 2.500000E+07

FREQuency? [MINimum | MAXimum]
Queries the output frequency.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 2.500000E+07

PERiod {<Period in sec> | MINimum | MAXimum}
Defines the output perid.

Parameters:
<Period> Numeric value in sec.

MIN: 4.00E-08
MAX: 1.000000E+05

PERiod? [MINimum | MAXimum]
Queries the output period.

Return values: Numeric value in sec.

MIN: 4.00E-08
MAX: 1.000000E+05

VOLTage:UNIT {VOLT | DBM}
Sets the amplitude unit.

Parameters:
<Unit> VOLT | DBM

VOLT: Amplitude unit setting in V (Volt).
DBM: Amplitude unit setting in dBm.

VOLTage:UNIT?
Queries the amplitude unit type.

Return values: VOLT | DBM

VOLTage {<Amplitude> | MINimum | MAXimum}
Selects the amplitude voltage value.

Parameters:
<Amplitude> Numeric value in V (Volt).

MIN: 1.00E-02
MAX: 2.00000E+01

*RST: 1.0000E+00

VOLTage? [MINimum | MAXimum]
Queries the amplitude voltage value.

Return values: Numeric value in V (Volt).

MIN: 1.00E-02
MAX: 2.00000E+01

VOLTage:HIGH {<Voltage> | MINimum | MAXimum}
Defines the high level voltage depending on the amplitude setting.

Parameters:
<Voltage> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: 2.5000E+00

VOLTage:HIGH? [MINimum | MAXimum]
Queries the high level voltage.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

VOLTage:LOW {<Voltage> | MINimum | MAXimum}
Defines the low level voltage depending on the amplitude setting.

Parameters:
<Voltage> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: - 2.5000E+00

VOLTage:LOW? [MINimum | MAXimum]
Queries the low level voltage.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

VOLTage:OFFSet {<Offset> | MINimum | MAXimum}
Defines the output offset value depending on the amplitude setting.

Parameters:
<Offset> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: 0.0E+00

VOLTage:OFFSet? [MINimum | MAXimum]
Queries the output offset value.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

FUNCtion:PULSe:WIDTh:HIGH {<Time> | MINimum | MAXimum}
Defines the high width of the pulse function depending on the frequency setting.

Parameters:
<Time> Numeric value in sec depending on the frequency setting.

MIN: 2.00E-08
MAX: 1.000000E+05

*RST: 1.00000E-05

FUNCtion:PULSe:WIDTh:HIGH? [MINimum | MAXimum]
Queries the high width of the pulse function depending on the frequency setting.

Return values: Numeric value in sec depending on the frequency setting.

MIN: 2.00E-08
MAX: 1.000000E+05

FUNCtion:PULSe:WIDTh:LOW {<Time> | MINimum | MAXimum}
Defines the low width of the pulse function depending on the frequency setting.

Parameters:
<Time> Numeric value in sec depending on the frequency setting.

MIN: 2.00E-08
MAX: 1.000000E+05

*RST: 1.00000E-05

FUNCtion:PULSe:WIDTh:LOW? [MINimum | MAXimum]
Queries the low width of the pulse function depending on the frequency setting.

Return values: Numeric value in sec depending on the frequency setting.

MIN: 2.00E-08
MAX: 1.000000E+05

FUNCtion:PULSe:DCYCle {<Percent> | MINimum | MAXimum}
Defines the duty cycle of the pulse function.

Parameters:
<Percent> Numeric value in percent.

MIN: 1.000000E-01
MAX: 9.990000E+01

*RST: 5.000000E+01

FUNCtion:PULSe:DCYCle? [MINimum | MAXimum]
Queries the duty cycle of the pulse function.

Return values: Numeric value in percent.

MIN: 1.000000E-01
MAX: 9.990000E+01

FUNCtion:PULSe:ETIMe {<Time> | MINimum | MAXimum}
Defines the edge time of the pulse function depending on the frequency settings.

Parameters:
<Time> Numeric value in sec.

MIN: 8.0E-09
MAX: 5.000E-07

*RST: 8.0E-09

FUNCtion:PULSe:ETIMe? [MINimum | MAXimum]
Defines the edge time of the pulse function depending on the frequency settings.

Return values: Numeric value in sec.

MIN: 8.0E-09
MAX: 5.000E-07

2.4.4 Arbitrary Configuration

FREQuency {<Frequency> | MINimum | MAXimum}
Defines the output frequency.

Parameters:
<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 2.500000E+07

FREQuency? [MINimum | MAXimum]
Queries the output frequency.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 2.500000E+07

PERiod {<Period in sec> | MINimum | MAXimum}
Defines the output perid.

Parameters:
<Period> Numeric value in sec.

MIN: 4.00E-08
MAX: 1.000000E+05

PERiod? [MINimum | MAXimum]
Queries the output period.

Return values: Numeric value in sec.

MIN: 4.00E-08
MAX: 1.000000E+05

VOLTage:UNIT {VOLT | DBM}
Sets the amplitude unit.

Parameters:
<Unit> VOLT | DBM

VOLT: Amplitude unit setting in V (Volt).
DBM: Amplitude unit setting in dBm.

VOLTage:UNIT?
Queries the amplitude unit type.

Return values: VOLT | DBM

VOLTage {<Amplitude> | MINimum | MAXimum}
Selects the amplitude voltage value.

Parameters:
<Amplitude> Numeric value in V (Volt).

MIN: 1.00E-02
MAX: 2.00000E+01

*RST: 1.0000E+00

VOLTage? [MINimum | MAXimum]
Queries the amplitude voltage value.

Return values: Numeric value in V (Volt).

MIN: 1.00E-02
MAX: 2.00000E+01

VOLTage:HIGH {<Voltage> | MINimum | MAXimum}
Defines the high level voltage depending on the amplitude setting.

Parameters:
<Voltage> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: 2.5000E+00

VOLTage:HIGH? [MINimum | MAXimum]
Queries the high level voltage.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

VOLTage:LOW {<Voltage> | MINimum | MAXimum}
Defines the low level voltage depending on the amplitude setting.

Parameters:
<Voltage> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

*RST: - 2.5000E+00

VOLTage:LOW? [MINimum | MAXimum]
Queries the low level voltage.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

VOLTage:OFFSet {<Offset> | MINimum | MAXimum}
Defines the output offset value depending on the amplitude setting.

Parameters:
<Offset> Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

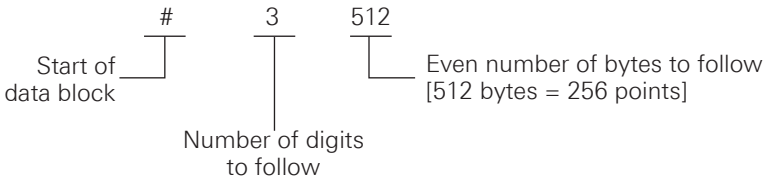
*RST: 0.0E+00

VOLTage:OFFSet? [MINimum | MAXimum]
Queries the output offset value.

Return values: Numeric value in V (Volt) depending on the amplitude setting.

MIN: - 1.00000E+01
MAX: 1.00000E+01

DATA {<binary block>}
Defines the arbitrary waveform. The instrument can handle data as binary values. In the binary block format, a block header precedes the waveform data. The block header has the following format:



The instrument represents binary data as 16-bit integers, which are sent as two bytes. Therefore, the total number of bytes is always twice the number of data points in the waveform and must always be an even number. For example, 512 bytes are required to download a waveform with 256 points.

DATA:MODulation {<Binary Block>}
Defines and transfers an arbitrary modulation waveform. The instrument can handle data as binary values. Please refer to the DATA {<binary block>} command.

DATA:SAVE „Arbitrary Name“
Transfers the previously generated arbitrary waveform from RAM (random access memory) into the non-volatile memory (ROM). The generated file is accessable via the internal file system afterwards.

Example: DATA:SAVE „ABC.HRT“

NOTICE

You are only able to save an USB stick file in HRT format into the internal memory of the HMF intrument.

FUNCtion:ARBitrary {<Waveform>}
Selects a built-in arbitrary waveform or the user-defined waveform which was transferred into the RAM (random access memory).

Parameters:

<Waveform> SINusoid | SQUare | PRAMp | NRAMp | TRIangle | WNOise | PNOise | CARDinal | EXPRise | EXPFall | RAM

SINusoid:	Sine wave function
SQUare:	Square wave function
PRAMp:	Positive ramp function
NRAMp:	Negative ramp function
TRIangle:	Triangle function
WNOise:	White noise function
PNOise:	Pink noise function
CARDinal:	Cardinal sine wave function
EXPRise:	Exponential rise function
EXPFall:	Exponential fall function
RAM:	User-defined waveform

FUNCtion:ARBitrary?
Queries the selected arbitrary waveform function.

Return values: SIN | SQU | PRAM | NRAM | TRI | WNO | PNO | CARD | EXPR | EXPF | RAM

2.5 Modulation commands

2.5.1 AM Modulation

AM:INTernal:FUNCtion <Shape>
Defines the shape of the AM modulation waveform.

Parameters:

<Shape> SINusoid | SQUare | PRAMp | NRAMp | TRIangle | WNOise | PNOise| CARDinal | EXPRise | EXPFall | RAM

SINusoid: Sine wave function
SQUare: Square wave function
PRAMp: Positive ramp function
NRAMp: Negative ramp function
TRIangle: Triangle function
WNOise: White noise function
PNOise: Pink noise function
CARDinal: Cardinal sine wave function
EXPRise: Exponential rise function
EXPFall: Exponential fall function
RAM: User-defined waveform

AM:INTernal:FUNCtion:FUNCtion?
Queries the shape of the AM modulation waveform.

Return values: SIN | SQU | PRAM | NRAM | TRI | WNO | PNO | CARD | EXPR | EXPF | RAM

AM:INTernal:FREQuency {<Frequency> | MINimum | MAXimum}
Defines the frequency of the AM modulation waveform.

Parameters:
<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+04

*RST: 5.000000E+03

AM:INTernal:FREQuency? [MINimum | MAXimum]
Queries the frequency of the AM modulation waveform.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+04

AM:DEPTh {<Modulation> | MINimum | MAXimum}
Defines the AM modulation depth.

Parameters:
<Modulation> Numeric value in percent.

MIN: 1.00E+00
MAX: 1.0000E+02

*RST: 1.0000E+02

AM:DEPTh? [MINimum | MAXimum]
Queries the AM modulation depth.

Return values: Numeric value in percent.

MIN: 1.00E+00
MAX: 1.0000E+02

AM:SOURce {INTernal | EXTernal}
Defines the AM modulation source.

Parameters:
<Source> INTernal | EXTernal

INTernal: Internal modulation source.
EXTernal: External modulation source.

*RST: INT

AM:SOURce?
Queries the AM modulation source.

Return values: INT | EXT

AM:STATe {OFF | ON | 0 | 1}
Activates or deactivates the AM modulation.

Parameters:

<State>	**ON | 1**:	The AM modulation will be activated.
	OFF | 0:	The AM modulation will be disabled.
	*RST: OFF | 0	

AM:STATe?
Queries the state of the AM modulation.

Return values:	**ON**:	The AM modulation is activated.
	OFF:	The AM modulation is disabled.

2.5.2 FM Modulation

FM:INTernal:FUNCtion <Shape>
Defines the shape of the FM modulation waveform.

Parameters:

<Shape> SINusoid | SQUare | PRAMp | NRAMp | TRIangle | WNOise | PNOise| CARDinal | EXPRise | EXPFall | RAM

SINusoid: Sine wave function
SQUare: Square wave function
PRAMp: Positive ramp function
NRAMp: Negative ramp function
TRIangle: Triangle function
WNOise: White noise function
PNOise: Pink noise function
CARDinal: Cardinal sine wave function
EXPRise: Exponential rise function
EXPFall: Exponential fall function
RAM: User-defined waveform

FM:INTernal:FUNCtion:FUNCtion?
Queries the shape of the FM modulation waveform.

Return values: SIN | SQU | PRAM | NRAM | TRI | WNO | PNO | CARD | EXPR | EXPF | RAM

FM:INTernal:FREQuency {<Frequency> | MINimum | MAXimum}
Defines the frequency of the FM modulation waveform.

Parameters:

<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+04

*RST: 5.000000E+03

FM:INTernal:FREQuency? [MINimum | MAXimum]
Queries the frequency of the FM modulation waveform.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+04

FM:DEViation {<Peak frequency> | MINimum | MAXimum}
Defines the deviation of the FM modulation.

Parameters:
<Peak frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+04

*RST: 1.000000E+03

FM:DEViation? [MINimum | MAXimum]
Queries deviation of the FM modulation.

Return values: Numeric value in percent.

MIN: 1.0E-05
MAX: 5.000000E+04

FM:SOURce {INTernal | EXTernal}
Defines the FM modulation source.

Parameters:
<Source> INTernal | EXTernal

INTernal: Internal modulation source.
EXTernal: External modulation source.

*RST: INT

FM:SOURce?
Queries the FM modulation source.

Return values: INT | EXT

FM:STATe {OFF | ON | 0 | 1}
Activates or deactivates the FM modulation.

Parameters:

<State>	**ON | 1**:	The FM modulation will be activated.
	OFF | 0:	The FM modulation will be disabled.
	*RST: OFF | 0	

FM:STATe?
Queries the state of the FM modulation.

Return values:	**ON**:	The FM modulation is activated.
	OFF:	The FM modulation is disabled.

2.5.3 PM Modulation

PM:INTernal:FUNCtion <Shape>
Defines the shape of the PM modulation waveform.

Parameters:

<Shape>	SINusoid | SQUare | PRAMp | NRAMp | TRIangle | WNOise | PNOise| CARDinal | EXPRise | EXPFall | RAM	
	SINusoid:	Sine wave function
	SQUare:	Square wave function
	PRAMp:	Positive ramp function
	NRAMp:	Negative ramp function
	TRIangle:	Triangle function
	WNOise:	White noise function
	PNOise:	Pink noise function
	CARDinal:	Cardinal sine wave function
	EXPRise:	Exponential rise function
	EXPFall:	Exponential fall function
	RAM:	User-defined waveform

PM:INTernal:FUNCtion:FUNCtion?
Queries the shape of the PM modulation waveform.

Return values: SIN | SQU | PRAM | NRAM | TRI | WNO | PNO | CARD | EXPR | EXPF | RAM

PM:INTernal:FREQuency {<Frequency> | MINimum | MAXimum}
Defines the frequency of the PM modulation waveform.

Parameters:
<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+04

*RST: 5.000000E+03

PM:INTernal:FREQuency? [MINimum | MAXimum]
Queries the frequency of the PM modulation waveform.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+04

PM:DEViation {<Phase> | MINimum | MAXimum}
Defines the deviation of the PM modulation.

Parameters:
<Phase> Numeric value in degree.

MIN: - 1.8000E+02
MAX: 1.8000E+02

*RST: 0.0E+00

PM:DEViation? [MINimum | MAXimum]
Queries deviation of the PM modulation.

Return values: Numeric value in percent.

MIN: - 1.8000E+02
MAX: 1.8000E+02

PM:SOURce {INTernal | EXTernal}
Defines the PM modulation source.

Parameters:

<Source> INTernal | EXTernal

INTernal: Internal modulation source.
EXTernal: External modulation source.

*RST: INT

PM:SOURce?
Queries the PM modulation source.

Return values: INT | EXT

PM:STATe {OFF | ON | 0 | 1}
Activates or deactivates the PM modulation.

Parameters:

<State>
ON | 1: The PM modulation will be activated.
OFF | 0: The PM modulation will be disabled.

*RST: OFF | 0

PM:STATe?
Queries the state of the PM modulation.

Return values:
ON: The PM modulation is activated.
OFF: The PM modulation is disabled.

2.5.4 FSK Modulation

FSKey:FREQuency {<Frequency> | MINimum | MAXimum}
Defines the frequency of the FSK modulation waveform.

Parameters:
<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+07

*RST: 1.000000E+03

FSKey:FREQuency? [MINimum | MAXimum]
Queries the frequency of the FSK modulation waveform.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+07

FSKey:INTernal:RATE {<Rate> | MINimum | MAXimum}
Defines the frequency rate of the FSK modulation (time frame between frequency shifts).

Parameters:
<Rate> Numeric value in Hz.

MIN: 2.000E-03
MAX: 2.500000E+05

FSKey:INTernal:RATE? [MINimum | MAXimum]
Queries the frequency rate of the FSK modulation.

Return values: Numeric value in Hz.

MIN: 2.000E-03
MAX: 2.500000E+05

FSKey:DCYCle {<Duty cycle> | MINimum | MAXimum}
Defines the duty cycle of the FSK modulation.

Parameters:
<Rate> Numeric value in percent.

MIN: 1.00E+00
MAX: 1.0000E+02

FSK:DCYCle? [MINimum | MAXimum]
Queries the duty cycle of the FSK modulation.

Return values: Numeric value in percent.

MIN: 1.00E+00
MAX: 1.0000E+02

FSKey:SOURce {INTernal | EXTernal}
Defines the FSK modulation source.

Parameters:
<Source> INTernal | EXTernal

INTernal: Internal modulation source.
EXTernal: External modulation source.

*RST: INT

FSKey:SOURce?
Queries the FSK modulation source.

Return values: INT | EXT

FSKey:STATe {OFF | ON | 0 | 1}
Activates or deactivates the FSK modulation.

Parameters:

<State>	**ON | 1**:	The FSK modulation will be activated.
	OFF | 0:	The FSK modulation will be disabled.
	*RST: OFF | 0	

FSKey:STATe?
Queries the state of the FSK modulation.

Return values:	**ON**:	The FSK modulation is activated.
	OFF:	The FSK modulation is disabled.

2.5.5 PWM Modulation

PWM:INTernal:FUNCtion <Shape>
Defines the shape of the PWM modulation waveform.

Parameters:

<Shape> SINusoid | SQUare | PRAMp | NRAMp | TRIangle | WNOise | PNOise| CARDinal | EXPRise | EXPFall | RAM

SINusoid: Sine wave function
SQUare: Square wave function
PRAMp: Positive ramp function
NRAMp: Negative ramp function
TRIangle: Triangle function
WNOise: White noise function
PNOise: Pink noise function
CARDinal: Cardinal sine wave function
EXPRise: Exponential rise function
EXPFall: Exponential fall function
RAM: User-defined waveform

NOTICE

The PWM modulation is only available in combination with the pulse wave function.

PWM:INTernal:FUNCtion?
Queries the shape of the PWM modulation waveform.

Return values: SIN | SQU | PRAM | NRAM | TRI | WNO | PNO | CARD | EXPR | EXPF | RAM

PWM:INTernal:FREQuency {<Frequency> | MINimum | MAXimum}
Defines the frequency of the PWM modulation waveform.

Parameters:
<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+04

*RST: 1.00000E-01

PWM:INTernal:FREQuency? [MINimum | MAXimum]
Queries the frequency of the PWM modulation waveform.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+04

PWM:DCYCle {<Deviation> | MINimum | MAXimum}
Defines the duty cycle deviation of the PWM modulation.

Parameters:
<Deviation> Numeric value in percent.

MIN: 0.0E+00
MAX: 5.000000E+01

*RST: 2.500000E+01

PWM:DCYCle? [MINimum | MAXimum]
Queries the duty cycle deviation of the PWM modulation.

Return values: Numeric value in percent.

MIN: 0.0E+00
MAX: 5.000000E+01

PWM:SOURce {INTernal | EXTernal}
Defines the PWM modulation source.

Parameters:
<Source> INTernal | EXTernal

INTernal: Internal modulation source.
EXTernal: External modulation source.

*RST: INT

PWM:SOURce?
Queries the PWM modulation source.

Return values: INT | EXT

PWM:STATe {OFF | ON | 0 | 1}
Activates or deactivates the PWM modulation.

Parameters:

<State>	**ON | 1**:	The PWM modulation will be activated.
	OFF | 0:	The PWM modulation will be disabled.
	*RST: OFF | 0	

PWM:STATe?
Queries the state of the PWM modulation.

Return values:	**ON**:	The PWM modulation is activated.
	OFF:	The PWM modulation is disabled.

2.6 Sweep commands

FREQuency:STARt {<Frequency> | MINimum | MAXimum}
Defines the start frequency of the sweep.

Parameters:

<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+07

*RST: 1.000000E+03

FREQuency:STARt? [MINimum | MAXimum]
Queries the start frequency of the sweep.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+07

FREQuency:STOP {<Frequency> | MINimum | MAXimum}
Defines the stop frequency of the sweep.

Parameters:
<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+07

*RST: 1.000000E+05

FREQuency:STOP? [MINimum | MAXimum]
Queries the stop frequency of the sweep.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+07

NOTICE

Setting or changing of parameters are also possible during a sweep, any changes will be immediately apparent. The sweep actually running will be terminated and a new one started; the display will show the parameters activated.

FREQuency:CENTer {<Frequency> | MINimum | MAXimum}
Defines the center frequency of the sweep.

Parameters:
<Frequency> Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+07

*RST: 4.950000E+04

FREQuency:CENTer? [MINimum | MAXimum]
Queries the center frequency of the sweep.

Return values: Numeric value in Hz.

MIN: 1.0E-05
MAX: 5.000000E+07

FREQuency:SPAN {<Frequency> | MINimum | MAXimum}
Defines the span of the sweep.

Parameters:
<Frequency> Numeric value in Hz.

MIN: -5.000000E+07
MAX: 5.000000E+07

*RST: 9.900000E+04

FREQuency:SPAN? [MINimum | MAXimum]
Queries the span of the sweep.

Return values: Numeric value in Hz.

MIN: -5.000000E+07
MAX: 5.000000E+07

SWEep:SPACing {LINear | LOGarithmic}
Defines the sweep spacing.

Parameters:
<Spacing> LINear | LOGarithmic

LINear: Linear spacing.
LOGarithmic: Logarithmic spacing.

*RST: LIN

SWEep:SPACing?
Queries the sweep spacing.

Return values: LIN | LOG

SWEep:TIME {<Time> | MINimum | MAXimum}
Defines the sweep time from start frequency to the stop frequency.

Parameters:
<Time> Numeric value in sec.

MIN: 1.000000E-03
MAX: 5.000000E+02

*RST: 1.000000E+01

SWEep:TIME? [MINimum | MAXimum]
Queries the sweep time.

Return values: Numeric value in sec.

MIN: 1.000000E-03
MAX: 5.000000E+02

SWEep:STATe {OFF | ON | 0 | 1}
Activates or deactivates the sweep function.

Parameters:
<State>
ON | 1: The sweep function will be activated.
OFF | 0: The sweep function will be disabled.

*RST: OFF | 0

SWEep:STATe?
Queries the state of the sweep function.

Return values:
ON: The sweep function is activated.
OFF: The sweep function is disabled.

MARKer:FREQuency {<Frequency> | MINimum | MAXimum}
Defines the sweep marker frequency. The marker frequency have be set between the start and stop frequency. If the signal frequency reaches the marker frequency a signal will be generated available at the TRIG OUTPUT connector.

Parameters:
<Time> Numeric value in Hz depending on the start and stop frquency.

MARKER:FREQuency? [MINimum | MAXimum]
Queries the sweep marker frequency.

Return values: Numeric value in Hz depending on the start and stop frquency.

MARKer {OFF | ON | 0 | 1}
Activates or deactivates the sweep marker function.

Parameters:
<State>
ON | 1: The sweep marker function will be activated.
OFF | 0: The sweep marker function will be disabled.

*RST: OFF | 0

MARKer?
Queries the state of the sweep marker function.

Return values: **ON**: The sweep marker function is activated.
OFF: The sweep marker function is disabled.

MARKer:STYLe {NORMal | CURSor | PULSe}
Defines the sweep marker style.

Parameters:
<Style> NORMal | CURSor | PULSe

MARKer:STYLe?
Queries the sweep marker style.

Return values: NORM | CURS | PULS

2.7 Burst commands

BURSt:MODE <Mode>
Defines the burst mode.

Parameters:
<Mode> TRIGgered | GATed

TRIGgered: A trigger will generate a burst with a predefined number of cycles.
GATed: The signal will be either on or off, depending on the level of the external signal at the „Trigger input/ouput" connector. If the gate signal is „true" (high +5 V), the function generator will deliver a continuous signal until the „gate" closes (0V TTL low). If no power is applied to the TRIG INPUT connector, the output signal will stop as the function generator will stop generation.

BURSt:MODE?
Queries the burst mode.

Return values: TRIG | GAT

BURSt:NCYCles {<Number of cycles> | MINimum | MAXimum}
Defines the number of burst cycles.

Parameters:
<Number of cycles> Numeric value

MIN: 1.0E+00
MAX: 5.00000E+04

*RST: 1.0E+00

BURSt:NCYCles? [MINimum | MAXimum]
Queries the number of burst cycles.

Return values: Numeric value

MIN: 1.0E+00
MAX: 5.00000E+04

BURSt:INTernal:PERiod {<Period> | MINimum | MAXimum}
Defines the burst period.

Parameters:
<Period> Numeric value in sec.

MIN: 2.10000E-05
MAX: 5.000000E+02

*RST: 1.000000E-04

BURSt:INTernal:PERiod? [MINimum | MAXimum]
Queries the burst period.

Return values: Numeric value in sec.

MIN: 2.10000E-05
MAX: 5.000000E+02

BURSt:PHASe {<Phase> |MINimum | MAXimum}
Defines the burst phase.

Parameters:
<Period> Numeric value in degrees.

MIN: 0.0E+00
MAX: 3.6000E+02

*RST: 0.0E+00

BURSt:PHASe? [MINimum | MAXimum]
Queries the burst phase.

Return values: Numeric value in degrees.

MIN: 0.0E+00
MAX: 3.6000E+02

BURSt:STATe {OFF | ON | 0 | 1}
Activates or deactivates the burst function.

Parameters:

<State>	**ON | 1**:	The burst function will be activated.
	OFF | 0:	The burst function will be disabled.
	*RST: OFF | 0	

BURSt:STATe?
Queries the state of the burst function.

Return values:	**ON**:	The burst function is activated.
	OFF:	The burst function is disabled.

2.8 Trigger commands

TRIGger
Generates a manual trigger signal in external trigger mode.

Usage: Event

TRIGger:SOURce {IMMediate | EXTernal}
Defines the trigger source.

Parameters:

<Source>	IMMediate | EXTernal	
	IMMediate:	Internal trigger source.
	EXTernal:	External trigger source; external source needs to be connected to the front “trigger input” connector (TTL).
	*RST: IMM	

TRIGger:SOURce?
Queries the trigger source.

Return values: IMM | EXT

TRIgger:SLOPe {POSitive | NEGative}
Defines the slope of the trigger input.

Parameters:
<Slope> POSitive | NEGative

POSitive: Rising edge
NEGative: Falling edge

*RST: POS

TRIGger:SLOPe?
Queries the slope of the trigger input.

Return values: POS | NEG

OUTPut:TRIGger {OFF | ON | 0 | 1}
Activates or deactivates the trigger output (TTL - front BNC connector).

Parameters:
<State> **ON | 1**: The trigger output will be activated.
OFF | 0: The trigger output will be disabled.

*RST: OFF | 0

OUTPut:TRIGger?
Queries the state of the trigger output.

Return values: **ON**: The trigger output is activated.
OFF: The trigger output is disabled.

OUTPut:TRIGger:SLOPe {POSitive | NEGative}
Defines the slope of the trigger output.

Parameters:
<Slope> POSitive | NEGative

POSitive: Rising edge
NEGative: Falling edge

*RST: POS

OUTPut:TRIGger:SLOPe?
Queries the slope of the trigger output.

Return values: POS | NEG

2.9 Screenshot commands

HCOPy:DATA?
Returns the actual display content (screenshot) in block format.

Example: - Reading the screenshot data bytes
- Removing the specific header and saving the data stream

#6333878BM6
#6 = 6 letters following
333878 = number of bytes to be transmitted
BM = BMP file format

In order to get a valid BMP which can be opened by any standard image viewer, the header information must be stripped down and removed until the file starts with „**BM6**".

Usage: Query only

HCOPy:FORMat
Defines the data format of the screenshot.

Parameters:
<Format> BMP | GIF | PNG

BMP: Windows Bitmap Format
GIF: Graphics Interchange Format
PNG: Portable Network Graphics Format

HCOPy:FORMat?
Queries the data format of the screenshot.

Return values: BMP | GIF | PNG

HCOPy:SIZE:X?
Queries the horizontal expansion of the screenshots.

Usage: Query only

HCOPy:SIZE:Y?
Queries the vertical expansion of the screenshots.

Usage: Query only

3 SCPI Commands

in alphabetic order

© 2015 Rohde & Schwarz GmbH & Co. KG
Mühldorfstr. 15, 81671 München, Germany
Phone: +49 89 41 29 - 0
Fax: +49 89 41 29 12 164
E-mail: info@rohde-schwarz.com
Internet: www.rohde-schwarz.com
Customer Support: www.customersupport.rohde-schwarz.com
Service: www.service.rohde-schwarz.com
Subject to change – Data without tolerance limits is not binding.
R&S® is a registered trademark of Rohde & Schwarz GmbH & Co. KG.
Trade names are trademarks of the owners.

5800.5718.02 │Version 01 │R&S®HMF25xx

The following abbreviations are used throughout this manual: R&S®HMF25xx is abbreviated as R&S HMF25xx.