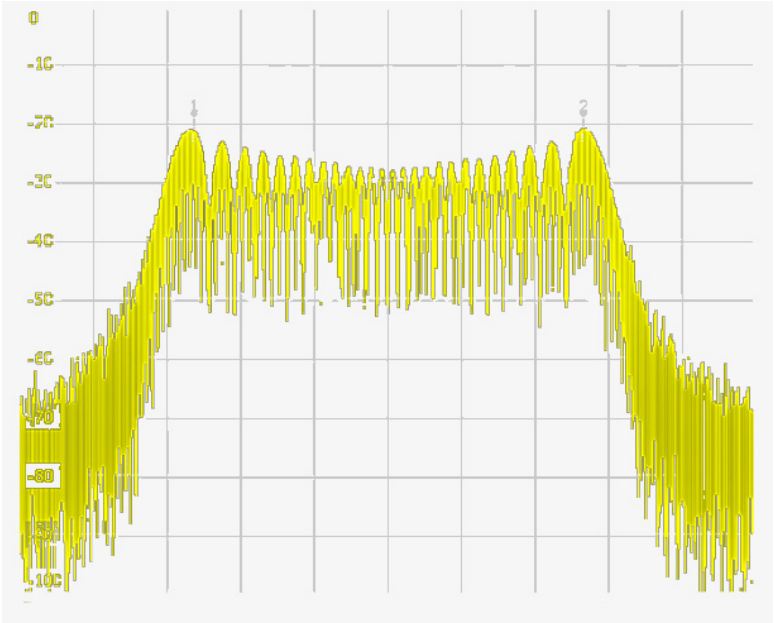
HAMEG Instruments
A Rohde&Schwarz Company

SCPI Programmers Manual
HMS Series / HMS-X

Firmware Version: 2.0 and later

English



HAMEG Instruments GmbH • Industriestraße 6 • D-63533 Mainhausen
Tel.: +49 (0) 6182 800 0 • Fax: +49 (0) 6182 800 100 • E-Mail: info@hameg.com
Geschäftsführer: Dipl.-Ing. Holger Asmussen • AG Offenbach am Main HRB 41200

Content

1 Introduction

If your HAMEG instrument is equipped with an interface, it can be remote controlled. You get access to nearly all functions that can be called manually via the front panel. HAMEG Instruments offers instrument control via:

HO720: USB and RS-232
HO730: USB and Ethernet
HO740: GPIB

1.1 What is SCPI?

SCPI (standard commands for programmable instruments) is an international standard (IEEE-488.2) for external control of measuring instruments etc. It allows you to network complex test setup with host (PC) control. This allows enormous time and manpower reduction.

The SCPI standard contains the programming syntax as well as the main functions of measuring instrument etc. This enables the exchange of comparable instruments from different producers without the need to reprogram the test procedure, as long as these functions and commands have been implemented identically.

1.2 SCPI Syntax

1.2.1 Spelling

Always pay attention that correct spelling is used when composing SCPI commands. All forms of spelling are forbidden except the exact short and long form of a command. Upper and lower case writing can be used. In this document upper case is used for short form, followed by lower case for the long form.

Example: The selection of the menu item 'Measure'

Short form	**:MEAS**
Long form	**:MEASure**
Acceptable spelling	**:MEASURE**
Unacceptable spelling	**:MEASUR**

1.2.2 Path structure

SCPI commands are based on a hierarchical order like a root (tree structure). Each command consists of the declaration of paths, different functions, keywords etc. and the optional allocation of parameters.

1.2.3 Colon

A colon serves as a separator for several key words in the path declaration. Based on the current path a ':' selects a lower hierarchy level. A ':' at the beginning of a command indicates that the following declaration is an element at the ' ROOT' level. The ':' is not applicable, if access is to be made to several elements of the same path. The multiple use of a path is forbidden, if the following command skips to a lower hierarchical level.

1.2.4 Semicolon

A semicolon is used to separate commands from one another.

1.2.5 Comma

If several values can be allocated to a function, they must be separated by commas.

1.2.6 Query

A command forcing the instrument to a direct reply is called a query. They can be used to query system states, parameters and possible border functions. Parameter read out is carried out by question mark (?). Common commands are read without path declaration.

Path and parameters of program command queries have to be specified additionally.

Query without parameter	**:FREQ:SPAN?**	**1.0E+06, 1MHz**
Query with parameter	**:FREQ:SPAN? MIN**	**1.0E+02, 100Hz**

1.2.7 Parameter

The transmission of parameters to the instrument is made with the declaration of path and the respective value. The latter is separated from the path by a space character. Please note the different data formats in which values can be assigned.

1.2.8 Instruction termination

The SCPI standard contains so called PMTs (program message terminators) used during instrument control to enable the identification of the end of a command or query by decoding the message bytes. Differentiation is made between 'new line' (NL) and 'end' (END). NL (defined as ' h0a') will e.g. be transmitted as termination of a command string. Any combination of NL and END is possible. However an instrument has to treat NL, NL+END and END semantically equivalently.

1.2.9 Data Formats

Float

At the input of floating point numbers a '.' is used as a decimal separator. Floating point numbers can be delineated in the following ways:

integer	**102**
positive real number	**+10.2**
negative real number	**-10.2**
with exponent	**1.2E-3**
without leading zero	**.123**

The input of the positive leading sign '+' is optional.

String

When designating strings as parameters, the string to be transferred is set in quotation marks (""). The string is defined as a whole value and therefore is separated from the path by a space character.

Character

Character data are text characters which are not set in "". For example, the activation of marker 1:

Example: **:MARK1:STAT ON**

In this case ON is the value the function can take over.

Block

This format is especially used for outputting great amounts of data , e.g. when a signal trace or the current system settings are read out. The structure of a data block is as follows:

<#> <ln> <n> <1bytes data>

#	marking a special data format
ln	length of the number that contains the number of data bytes

n	number of data bytes
data	data bytes (1 .. n)

Example for the data stream caused by a query:

#3600abc … xyz

#	start of block data
3	the number containing number of databytes consists of 3 characters
600	number of subsequent data (456 bytes)
a	value of 1st data byte
b	value of 2nd data byte
z	value of 600th data byte

Special number formats

#H	description in hexadecimal form	#Hxxxxxxxx
#B	description in binary form	#Bxxxxxxxx
#Q	description in octal form	#Qxxxxxxxx

1.3 State and Event

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.

SESR - Standard Event State Register

The Standard Event Status Register includes status indicators as well as error messages of the instrument.

Bit 7							Bit 0
R	R	R	R	R	R	R	R
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Table 1: SESR - Standard Event Status Register

PON	Power On	The instrument was switched on
URQ	User Request	unused (0)
CME	Command Error	Error during the analysis of a command
EXE	Execution Error	Error during command execution
DDE	Device Dependend Error	An instrument error has appeared
QYE	Query Error	Data got lost or are not available during a query
RQC	Request Control	unused (0)
OPC	Operation Complete	all current operations have ended

SESER Standard Event State Enable Register

Bit 7							Bit 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Table 2: SESER - Standard Event Status Enable Register

The SESER determines which events are evaluated.

1	Event will be evaluated
0	Event will be ignored

SBR - State Byte Register

Bit 7							Bit 0
R	R	R	R	R	R	R	R
/	RQS/MSS	ESB	MAV	/	/	/	/

Table 3: SBR - State Byte Register

RQS	Request Service	instrument requests to send
MSS	Master Status Summary	logical sum of ESB and MAV bits
ESB	Event State Bit	an event message is available
MAV	Message Available	an output is available

SRER - Service Request Enable Register

Bit 7							Bit 0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
/	/	ESB	MAV	/	/	/	/

Table 4: SRER - Service Request Enable Register

1	output permitted
0	output prohibited

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 wishes 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.

2 Overview of SCPI Commands

2.1 System- and register commands

*CLS

*ESE

*ESR

*IDN

*LRN

*OPC

*RST

*SRE

*STB

*WAI

*PSC

*TRG

SYSTem:DATE
SYSTem:DATE?

SYSTem:TIME
SYSTem:TIME?

SYSTem:NAME?

SYSTem:ERRor?

SYSTem:ERRor:NEXT?

SYSTem:ERRor:ALL?

SYSTem:ELISt?

SYSTem:SNUM?

SYSTem:SOFTware?

SYSTem:HARDware?

SYSTem:LANGuage
SYSTem:LANGuage?

2.2 Trigger commands

TRIGger:SOURce { IMMediate | EXTernal | VIDeo }
TRIGger:SOURce?

TRIGger:SLOPe { POSitive | NEGative }
TRIGger:SLOPe?

TRIGger:SOFTware — Only valid in single sweep mode

2.3 Configuration of the parameters

SYSTem:AUTotune

SYSTem:PRESet

SYSTem:REFerence { INTernal | EXTernal }
SYSTem:REFerence?

MEASure:TGENerator { ON | OFF | 0 | 1 } — Only valid for HMS1010 and HMS3010
MEASure:TGENerator?

MEAS:PAMP { ON | OFF | 0 | 1 } — Only valid with option HO3011
MEAS:PAMP?

MEASure {CFRX | M1RX}

MEASure:UNCAl?

BANDwidth:RBW { 100 | 300 | 100 | 3000 | 10000 | 30000 | 100000 | 200000 | 300000 | 1000000 | C200 | C9k | C120k | C1M }
BANDwidth:RBW?

BANDwidth:RBW:AUTO { ON | OFF | 0 | 1 }
BANDwidth:RBW:AUTO?

BANDwidth:VBW { 10 | 30 | 100 | 300 | 1000 | 3000 | 10000 | 30000 | 100000 | 300000 | 1000000 }
BANDwidth:VBW?

BANDwidth:VBW:AUTO { ON | OFF | 0 | 1 }
BANDwidth:VBW:AUTO?

AMPLitude:ATTenuation {LNOIse| LDIStortion}
AMPLitude: ATTenuation?

AMPLitude:ATTenuation:LEVel?

AMPLitude:RLEVel { <value in dBm/dBuV> | MINimum | MAXimum }
AMPLitude:RLEVel? [MINimum | MAXimum]

AMPLitude:UNIT {dBm | dBuV | V | W}
AMPLitude:UNIT?

AMPLitude:RANGe { LINear, 0.5, 1, 2, 5, 10 }
AMPLitude:RANGe?

AMPLitude:TGATtenuation ←value→
AMPLitude:TGATtenuation?

FREQuency:CENTer { <value in Hz> | MARKer[n] | MINimum | MAXimum }
FREQuency:CENTer? [MINimum | MAXimum]

FREQuency:CENTer:STEPsize { <value> | SPAN01 | SPAN05 | STC | MINimum | MAXimum }
FREQuency:CENTer:STEPsize?

FREQuency:SPAN { <value in Hz> | LAST | FULL | ZERO | MINimum | MAXimum }
FREQuency:SPAN? [MINimum | MAXimum]

FREQuency:STARt { <value in Hz> | MINimum | MAXimum }
FREQuency:STARt? [MINimum | MAXimum]

FREQuency:STOP { <value in Hz> | MINimum | MAXimum }
FREQuency:STOP? [MINimum | MAXimum]

FREQuency:STEP { <value in Hz> | MINimum | MAXimum } Only valid in Receiver Mode
FREQuency:STEP? [MINimum | MAXimum]

SWEep:TIME { <value in s> }
SWEep:TIME?

SWEep:TIME:AUTO { ON | OFF | 0 | 1 }
SWEep:TIME:AUTO?

SWEep:MODE { CONTinous | SINGle }
SWEep:MODE?

SWEep:STATe?

2.4 Trace capture commands

HCOPy?

HCOPy:DATA

HCOPy:FORM { BMP, GIF }
HCOPy:FORM?

HCOPy:SIZE:X?

HCOPy:SIZE:Y?

2.5 Receiver mode - Commands

SYSTem:MODE { SWEep | RMODe }
SYSTem:MODE?

RMODe:FREQuency { <value in Hz> | MINimum | MAXimum }
RMODe:FREQuency? [MINimum | MAXimum]

RMODe:FREQuency:STEP { <value in Hz> | MINimum | MAXimum }
RMODe:FREQuency:STEP? [MINimum | MAXimum]

RMODe:MTIMe { <value in s> | MINimum | MAXimum }
RMODe:MTIMe? [MINimum | MAXimum]

RMODe:DETector { PEAK | AVG | QPEak | RMS }
RMODe:DETector?

RMODe:AUDio:DEModulation { ON | OFF | 0 | 1 }
RMODe:AUDio:DEModulation?

RMODe:AUDio:MODulation { AM | FM }
RMODe:AUDio:MODulation?

RMODe:AUDio:VOLume { <value in %> | MINimum | MAXimum }
RMODe:AUDio:VOLume? [MINimum | MAXimum]

RMODe:LEVel?

2.6 Marker Commands

MARKer[n]:STATe { OFF | ON | 0 | 1 }
MARKer[n]:STATe?

MARKer[n]:MODE { POSition | DELTa }
MARKer[n]:MODE?

MARKer:FCOunter:STATe { OFF | ON | 0 | 1 }
MARKer:FCOunter:STATe?

MARKer:FCOunter:VALue?

MARKer:AOFF

MARKer[n]:[SET]:FREQuency { <value in Hz> | MINimum | MAXimum }
MARKer[n]:[SET]:FREQuency?

MARKer[n][:SET]:CENTer

MARKer[n]:NOISe { OFF | ON | 0 | 1 }
MARKer[n]:NOISe?

MARKer[n][:SET]:LEVel?

MARKer[n][:SET]:REFerence

2.7 Peak Commands

MARKer[n]:MAXimum:PEAK

MARKer[n]:MAXimum:NEXTpeak

MARKer[n]:MAXimum:LEFT

MARKer[n]:MAXimum:RIGHt

MARKer[n]:MINimum

MARKer[n]:MAXimum:ALL

2.8 Display Commands

DISPlay:TRACe { OFF | ON | 0 | 1 }
DISPlay:TRACe?

DISPlay:TRACE:INTensity { <value in percent> | MINimum | MAXimum }
DISPlay:TRACE:INTensity?

DISPlay:BACKlight { <value in percent> | MINimum | MAXimum }
DISPlay:BACKlight?

DISPlay:GRID { <value in percent> | MINimum | MAXimum }
DISPlay:GRID?

DISPlay:GRID:SETup { RETicle | LINE | OFF }
DISPlay:GRID:SETup?

DISPlay:GRID:SCALe { OFF | ON | 0 | 1 }
DISPlay:GRID:SCALe?

DISPlay:TRANsparancy { <value in percent> | MINimum | MAXimum }
DISPlay:TRANsparancy? [MINimum | MAXimum]

LED:BRIGhtness { HIGH | LOW }
LED:BRIGhtness?

2.9 Trace Commands

TRACe:MODE { CLR | MAXimum | MINimum | AVErage | HOLD }
TRACe:MODE?

TRACe:MEMory { SAVE | SHOW }

TRACE:MEMory:SHOW { ON | OFF | 0 | 1 }
TRACE:MEMory:SHOW?

TRACe:DETector { AUTO | SAMPle | MAXimum | MINimum }
TRACe:DETector?

TRACe:MATH { OFF | TMEM | MTRACe }
TRACe:MATH?

TRACe:DATA?

TRACe:DATA:FORMat {BIN | CSV}
TRACe:DATA:FORMat?

TRACe:DATA:BORDer {LSBFirst | MSBFirst}
TRACe:DATA:BORDer?

3 Detailed description of SCPI commands

3.1 System- and register commands

Common commands always start with a prefixed (*).They are special system commands and are used without path declaration.

***CLS** *CLS Resets state and error list and deletes the OPC state

***ESE** *ESE Sets SESER (Standard Event State Enable Register) content as a decimal number
*ESE? Returns SESER content
Example: Using of the Power On bit as state report
*ESE 128; *ESE? provides 128

***ESR** *ESR? Returns SESR (Standard Event State Register) content and resets it thereafter
Example: *ESR? provides ‚128', binary: 1000 0000
Power On bit is set, the instrument was switching on

***IDN** *IDN? Returns the instrument identification
Example: *IDN? provides HAMEG,HMO3522,055970006,HW10100000,SW01.021

***LRN** *LRN? Returns all read- and writeable parameters; these are separated by semicolon and quoted with a header path (system and instruments parameter).

***OPC** *OPC Sets the Operation Complete Bit in the Standard Event State Register active, if dependent operations are finished
*OPC? If all dependent operations are finished, the OPC bit will not be set, but the output will be directly output as „1".

***RST** *RST Causes a new start with the factory default settings

***SRE** *SRE Writes into SRER register (Service Request Enable Register)
*SRE? Returns SRER
Example: Disabling of all remote response, sets the register to zero
*SRE 0 ; *SRE? provides , 0 ,

***STB** *STB? Returns SBR (State Byte Register) content
Example: *STB? provides , 32 ,
Event State Bit ESB and Message Available MAV are set, a event entry and output query are available.

***WAI** Waits for all pending operations to complete before executing any additional commands over the interface.

***PSC** <Action> Power on Status Clear
<Action>: 0 | 1
0: The contents of the status registers are preserved.
1: Resets the status registers.
Determines whether the contents of the ENABle registers are preserved or reset when the instrument is switched on.
Thus a service request can be triggered when the instrument is switched on, if the status registers ESE and SRE Are suitably configured. The query reads out the contents of the „power-on-status-clear" flag.

***TRG** Triggers all actions waiting for a trigger event. In particular, *TRG generates a manual trigger signal (manual trigger).

SYSTem:DATE
Sets the date.

SYSTem:DATE <date>
<date> = The date given in the format: year – month – day.
Example: :SYSTem:DATE 2009,5,1 for the date May 1, 2009.

SYSTem:DATE? will response <date>
Example: :SYSTem:DATE? answers 2009,5,1 , for May 1, 2009.

SYSTem:TIME
Sets the time.

SYSTem:TIME <time>
<time> = Time in the format: hour – minute – second.
Example: :SYSTem:TIME 12,5,0 , for setting the time to 12h 5min 0s.

SYSTem:TIME? will response <time>
Example: :SYSTem:TIME? answers 12,5,0.

SYSTem:NAME?
Queries the instrument name.

SYSTem:ERRor?
Reads one error from the error queue.
After reading of an error it will be erased from the register. The response consists of an error number and a short description of the error. Positive error numbers are instrument-dependent. Negative error numbers are reserved by the SCPI standard.
If the queue is empty, the response is 0,"No error".

SYSTem:ERRor:NEXT?
Queries the next error and removes it from the queue.
If the queue is empty, the response is 0,"No error".

SYSTem:ERRor:ALL?
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.
If the queue is empty, the response is 0,"No error"

SYSTem:ELISt?
Queries the system internal error list.
After reply the internal error list will be erased.

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

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

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

SYSTem
:LANGuage { GERMan | ENGLish | SPANish | FRENch }
:LANGuage?
Set the system laguage to german, english, french or spanish.
The LANG? query returns the current language set.

3.2 Trigger commands

TRIGger
:SOURce { IMMediate | EXTernal | VIDeo }
:SOURce?
Set the trigger source. IMM selects the internal source for the trigger, EXT selects an external source as reference for the trigger and VID selects the video trigger. The external source needs to be connected to the rear "trigger input" connector (TTL). The video trigger can only be activated in zero span (span = 0 Hz).
The SOUR? query returns the current setting of the trigger source (immediate, external, video).

TRIGger
:SLOPe { POSitive | NEGative }
:SLOPe?
Select whether the function generator uses the rising edge or falling edge of the trigger signal.
The SLOP? query returns the type of the edge.

TRIGger **Only valid in single sweep mode**
:SOFTware
Executes a trigger via remote command.

3.3 Configuration of the parameters

SYSTem
:AUTOtune
Triggers a full span search and automatically set up the instruments common settings (like RBW, VBW, span etc.) to display the highest peak found during the full span search.

SYSTem
:PREset
Reset the common settings (like RBW, VBW, span etc.) of the instrument to default values.

SYSTem
:REFerence { INTernal | EXTernal }
:REFerence?
Set the system reference clock (10MHz) to internal or external source.
The REF? query returns the current state of the system reference (internal, external).

MEASure **Only valid for HMS1010 and HMS3010**
:TGENerator { ON | OFF | 0 | 1 }
:TGENerator?
Enable or disable the internal tracking generator.
The TGEN? query returns the current setting of the tracking generator (ON, OFF).

MEASure **Only valid with option HO3011**

:PAMP { ON | OFF | 0 | 1 }
:PAMP?
Enable or disable the internal preamplifier.
The PAMP? query returns the current setting of the preamplifier (ON, OFF).

MEASure
{ CFRX | M1RX }
Switch the instrument into receiver mode and inherit the current center frequency of the sweep mode (CFRX) or switch the instrument into receiver mode and inherit the current marker frequency of marker one (M1RX).

MEASure
:UNCAl?
Queries whether the current instrument setting triggers the "UNCAL" message or not.

BANDwidth
:RBW { 100 | 300 | 1000 | 3000 | 10000 | 30000 | 100000 | 200000 | 300000 | 1000000 | C200 | C9k | C120k | C1M }
:RBW?
Set the resolution bandwidth filter.
The values are: 100, 300, 1000, 3000, 10000, 30000, 100000, 200000, 300000, 1000000 Hertz.
For remote control decimal as well as scientific number format is allowed.

For example:
For the 1MHz RBW filter 1000000 as well as 1e6 is valid.
The C200, C9k, C120k and C1M filters are the -6dB CISPR filters and need to be send to the HMS exactly like described.

For example:
To setup the 120kHz CISPR filter, the string „C120k" needs to be send to the instrument.
Scientific number format is not supported for the CISPR filters.
Please notice that all CISPR filters are only available in Receiver Mode.
The RBW? query returns the current resolution bandwidth filter setting in Hertz.

BANDwidth
:RBW:AUTO { ON | OFF | 0 | 1 }
:RBW:AUTO?
Enable or disable the automatic resolution bandwidth selection.
The AUTO? query returns the current setting of the automatic resolution bandwidth selection feature (ON, OFF).

BANDwidth
:VBW { 10 | 30 | 100 | 300 | 1000 | 3000 | 10000 | 30000 | 100000 | 300000 | 1000000 }
:VBW? [MINimum | MAXimum]
Set the video bandwidth filter.
The values are: 10, 30, 100, 300, 1000, 3000, 10000, 30000, 100000, 200000, 300000, 1000000, 3000000 Hertz.
The RBW? query returns the current video bandwidth fi lter setting in Hertz.
For remote control decimal as well as scientific number format is allowed. For example: For the 1MHz VBW filter 1000000 as well as 1e6 is valid.

BANDwidth
:VBW:AUTO { ON | OFF | 0 | 1 }
:VBW:AUTO?
Enable or disable the automatic video bandwidth selection.
The AUTO? query returns the current setting of the automatic video bandwidth selection feature (ON, OFF).

AMPLitude
:ATTenuation { LNOIse| LDIStortion }
:ATTenuation?
Change the attenuation setup.
Choose low noise (LNOIse) to minimize the noise or choose low distortion (LDIStortion) to minimize the distortion.

AMPLitude
:ATTenuation:LEVel?
The query returns the current setting of the attenuation level (is set automatically).

AMPLitude
:RLEVel { <value in dBm/dBuV> | MINimum | MAXimum }
:RLEVel? [MINimum | MAXimum]
Set the reference level of the Y-Axis in dBm, resp. dBµV (see also AMPL:UNIT).
Valid values are all whole numbers between +30dB...-500dB, or +30dBµV...-500dBµV.
MIN selects the lowest reference level and MAX selects the highest reference level allowed.
The RLEV? Query returns the current reference level in dBm, resp. dBµV.

AMPLitude
:UNIT { dBm | dBuV }
:UNIT?
Set the unit of the amplitude (Y-Axis) to dBm or dBµV.
The UNIT? query returns the current setting of the amplitude (dBm, dBµV).

AMPLitude
:RANGe { <LINear, 0.5, 1, 2, 5, 10> }
:RANGe?
Set the range per division of the amplitude (Y-Axis).
The RANG? query returns the current range per division setting in dB (0.5, 1, 2, 5,10) or as linear value.

AMPLitude
:TGATtenuation <value>
:TGATtenuation?
Set the output of the tracking generator attenuation value in 1dB steps up to 20dBm.
The TGAT? query returns the current value of the tracking generator attenuation.

FREQuency
:CENTer { <value in Hz> | MARKer[n] | MINimum | MAXimum }
:CENTer? [MINimum | MAXimum]
Set the center frequency.
Also allows to set the center frequency to the current marker frequency
MIN selects the lowest center frequency and MAX selects the highest center frequency.
The CENT? query returns the current center frequency in Hertz.

FREQuency
:CENTer:STEPsize { <value> | SPAN01 | SPAN05 | STC | MINimum | MAXimum }
:CENTer:STEPsize?
Set the step size of the center frequency.
All integer values between 1Hz and 3GHz are valid.
Also allows to set the center frequency step size to 0.1 x span, 0.5 x span or set the step size to the center frequency value.
MIN selects the lowest step size and MAX selects the highest step size possible.
The STEP? query returns the current step size in Hertz.

FREQuency

:SPAN { <value in Hz> | LAST | FULL | ZERO | MINimum | MAXimum }
:SPAN? [MINimum | MAXimum]
Set the frequency span.
All integer values between 1Hz and 3GHz are valid.
Also allows to set the frequency span to LAST (span), FULL (span) or ZERO (span).
MIN selects the lowest frequency span and MAX selects the highest frequency span.
The SPAN? query returns the current frequency span in Hertz.

FREQuency
:STARt { <value in Hz> | MINimum | MAXimum }
:STARt? [MINimum | MAXimum]
Set the start frequency.
MIN selects the lowest start frequency and MAX selects the highest start frequency.
The STAR? query returns the current start frequency in Hertz.

FREQuency
:STOP {<value in Hz> | MINimum | MAXimum }
:STOP? [MINimum | MAXimum]
Set the stop frequency.
MIN selects the lowest stop frequency and MAX selects the highest stop frequency.
The STOP? query returns the current stop frequency in Hertz.

FREQuency **Only valid in Receiver Mode**
:STEP { <value in Hz> | MINimum | MAXimum }
:STEP? [MINimum | MAXimum]
Set the center frequency step size.
MIN selects the lowest center frequency step size and MAX selects the highest center frequency step size.
The STEP? query returns the current center frequency step size in Hertz.

SWEep
:TIME {<value in s> }
:TIME?
Set the time required to sweep from the start frequency to the stop frequency.
The TIME? query returns the sweep time in seconds.

SWEep
:TIME:AUTO { ON | OFF | 0 | 1 }
:TIME:AUTO?
Activate or deactivate the automatic sweep time mode.
The AUTO? query returns the current setting of the sweep time mode (ON,OFF).

SWEep
:MODE { CONTinous | SINGle }
:MODE?
Set the sweep mode to continuous or single sweep.
The MODE? query returns the current setting of the sweep mode (continuous, single).

SWEep
:STATe?
Queries the state of the sweep (RUN, READY). Only applicable in single sweep mode.

3.4 Trace capture commands

HCOPy?

Returns the parameters of the screenshot.
Screenshots have a 640x520 format and will be responsed as Bitmap.

Example:
:HCOP:FORM BMP; :HCOP:SIZE:X 640; :HCOP:SIZE:Y 520

HCOPy

:DATA?
Returns the actual display content (screenshot).
The DATA? query responses the screenshot in block format.

HCOPy

:FORM { BMP | GIF }
:FORM?
This parameter selects the data format of the screenshot.
The FORM? query returns the current setting of the screenshot format (BMP, GIF).

BMP	Windows-Bitmap in 8-Bit size
GIF	Screenshot in Graphics Interchange Format

HCOPy

:SIZE:X?
Returns the horizontal expansion of the screenshots.

HCOPy

:SIZE:Y?
Returns the vertical expansion of the screenshots.

3.5 Receiver mode

SYSTem

:MODE { SWEep | RMODe }
:MODE?
Switch the measurement mode of the instrument to sweep mode or receiver mode.
The MODE? query returns the current setting of the measurement mode (sweep mode or receiver mode).

RMODe

:FREQuency { <value in Hz> | MINimum | MAXimum }
:FREQuency? [MINimum | MAXimum]
Set the receiver mode frequency.
MIN selects the lowest receiver mode frequency and MAX selects the highest receiver mode frequency.
The FREQ? query returns the current receiver mode frequency in Hertz.

RMODe

:FREQuency:STEP { <value in Hz> | MINimum | MAXimum }
:FREQuency:STEP? [MINimum | MAXimum]
Set the receiver mode frequency step size.
MIN selects the lowest receiver mode frequency step size and MAX selects the highest receiver mode frequency step size. The STEP? query returns the current receiver mode frequency step size in Hertz.

RMODe
:MTIMe { <value in s> | MINimum | MAXimum }
:MTIMe? [MINimum | MAXimum]
Set the receiver mode measurement time.
MIN selects the lowest receiver mode measurement time and MAX selects the highest receiver mode measurement time. The MTIM? query returns the current receiver mode measurement time in Seconds.

RMODe
:DETector { PEAK | AVG | QPEak | RMS }
:DETector?
Set the receiver mode detector to peak, average, quasi-peak or RMS mode.
The DET? query returns the current setting of the receiver mode detector (peak, avg, qpeak,rms).

RMODe
:AUDio:DEModulation { ON | OFF | 0 | 1 }
:AUDio:DEModulation?
Activate or deactivate the audio demodulation in receiver mode.
The DEMO? query returns the current setting of the audio demodulation (ON,OFF).

RMODe
:AUDio:MODulation { AM | FM }
:AUDio:MODulation?
Switch to AM/FM audio modulation in receiver mode.
The MOD? query returns the current setting of the audio modulation (AM,FM).

RMODe
:AUDio:VOLume { <value in %> | MINimum | MAXimum }
:AUDio:VOLume? [MINimum | MAXimum]
Set the sound volume in receiver mode.
MIN selects the lowest volume and MAX selects the highest volume.
The VOL? query returns the current volume setting in percent.

RMODe
:LEVel?
Queries the current measurement result in receiver mode.

3.6 Marker Commands

MARKer[n]
:STATe { OFF | ON | 0 | 1 }
:STATe?
Activate or deactivate the selected marker.
The HMS series provides up to 8 markers labeled with marker1 to marker8.
The STAT? query returns the state of the labeled marker (ON, OFF).

MARKer[n]
:MODE { POSition | DELTa }
:MODE?
Switch the marker mode to absolute position or delta marker mode.
The MODE? query returns the current marker mode (POSITION, DELTA).

MARKer
:FCOunter:STATe { OFF | ON | 0 | 1 }
:FCOunter:STATe?
Activate or deactivate the frequency counter.
The STAT? query returns the current state of the frequency counter (ON, OFF).

MARKer
: FCOunter:VALue?
Queries the current value of the frequency counter.

MARKer
:AOFF
Deactivate all activated markers.

MARKer[n]
:[SET]:FREQuency { <Value in Hz> | MINimum | MAXimum }
:[SET]:FREQuency?
Set the marker to an selected frequency.
MIN selects the lowest marker frequency and MAX selects the highest marker frequency.
The FREQ? query returns the current marker frequency in Hertz.

MARKer[n]
[:SET]:CENTer
Set the marker to the currently selected center frequency.

MARKer
:NOISe { OFF | ON | 0 | 1 }
:NOISe?
Activate or deactivate the noise marker function.
The NOIS? query returns the current status of the marker.

MARKer[n]
[:SET]:LEVel?
Returns the actual marker value in dBm/dBµV.

MARKer[n]
[:SET]:REFerence
Set the reference level of the HMS to the current value of the resp. marker.

3.7 Peak Commands

MARKer[n]
:MAXimum:PEAK
Perform a maximum peak search with the selected marker. The selected marker will be set to the highest peak of the actual measurement result.
There are 8 individual markers available.

MARKer[n]
:MAXimum:NEXTpeak
Perform a maximum peak search with the selected marker. The selected marker will be set to the next highest peak of the actual measurement result.
There are 8 individual markers available.

MARKer[n]
:MAXimum:LEFT
:MAXimum:RIGHt
Set the selected marker to the next right/left peak found from the current marker position.

MARKer[n]
:MINimum
Set the marker to the minimum value measured at the actual measurement result.

MARKer[n]
:MAXimum:ALL
Set all markers to the highest peak.

3.8 Display Commands

DISPlay
:TRACe { OFF | ON | 0 | 1 }
:TRACe?
Activate or deactivate the trace display.
The TRAC? query returns the current setting.

DISPlay
:TRACE:INTensity { <Value in %> | MINimum | MAXimum }
:TRACE:INTensity?
Set the intensity of the trace display.
MIN selects the lowest trace intensity and MAX selects the highest trace intensity.
The INTEN? query returns the current setting of the intensity in percent.

DISPlay
:BACKlight { <Value in %> | MINimum | MAXimum }
:BACKlight?
Set the intensity of the backlight.
MIN selects the lowest backlight intensity and MAX selects the highest backlight intensity.
The BACK? query returns the current setting of the intensity in percent.

DISPlay
:GRID { <Value in %> | MINimum | MAXimum }
:GRID?
Set the intensity of the display grid.
MIN selects the lowest grid intensity and MAX selects the highest grid intensity.
The GRID? query returns the current setting of the intensity in percent.

DISPlay
:GRID:SETup { RETicle | LINE | OFF }
:GRID:SETup?
Switch the grid appearance to reticle, line or off.
The SET? query returns the current setting (RETICLE, LINE, OFF).

DISPlay
:GRID:SCALe { OFF | ON | 0 | 1 }
:GRID:SCALe?
Switch the scale of the grid on or off.
The SCALE? query returns the current setting.

DISPlay
:TRANsparancy { <Value in %> | MINimum | MAXimum }
:TRANsparancy?
Set the display transparency.
MIN selects the lowest transparency and MAX selects the highest transparency.
The GRID? query returns the current setting of the intensity in percent.

LED
:BRIGhtness { HIGH | LOW }
:BRIGhtness?
Switch the LED brightness to high or low.
The GRID? qery returns the current setting of the LED brightness.

3.9 Trace Commands

TRACe
:MODE { CLR | MAXimum | MINimum | AVErage | HOLD }
:MODE?
Set the trace mode to clear, maximum, minimum, average or hold.
The MODE? query returns the current setting of the trace mode (CLR, MAX, MIN, AVERAGE, HOLD).

TRACe
:MEMory { SAVE | SHOW }
Store or recall a trace to/from the memory.

TRACe
:MEMory:SHOW { ON | OFF | 0 | 1 }
:MEMory:SHOW?
Activating/deactivating of the trace stored in memory before.
The SHOW? query returns the current state of the memory state display (ON, OFF).

TRACe

:DETector { AUTO | SAMPle | MAXimum | MINimum }
:DETector?
Set the type of trace detector to automatic, sample, maximum or minimum trace detector.
The DET? query returns the current setting (AUTO, SAMP, MAX, MIN).

TRACe

:MATH { OFF | TMEM | MTRACe }
:MATH?
Activate or deactivate the mathematic functions.
The mathematic functions are: Trace minus memory (TMEM), or memory minus trace (MTRACe).
The MATH? query returns the current setting (OFF, TMEM, MTRAC).

TRACe

:DATA?
Returns the waveform data in sweep mode.

TRACe

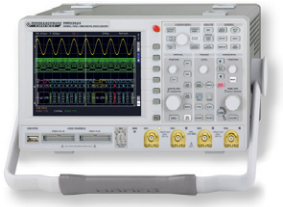
:DATA:FORMat { BIN | CSV }
:DATA:FORMat?
Set the waveform data format to binary (BIN) or comma-separated list of values (CSV).
The FORM? query returns the current format of the trace data.

TRACe

:DATA:BORDer { LSBFirst | MSBFirst }
:DATA:BORDer?
Set the byte order of the trace data.
The HMS represents binary data as 16-bit integers, which are sent as two bytes.
The BORD? query returns the byte order of the trace data (LSB, MSB).

HAMEG®
Instruments
A Rohde&Schwarz Company

Oscilloscopes



Spectrum Analyzer

Power Supplies

Modular System
Series 8000

Programmable Instruments
Series 8100

authorized dealer

www.hameg.com

Release: Februrary 2014

DQS-Certification: DIN EN ISO 9001
Reg.-Nr.: 071040 QM

HAMEG Instruments GmbH
Industriestraße 6
D-63533 Mainhausen
Tel +49 (0) 61 82 800-0
Fax +49 (0) 61 82 800-100
sales@hameg.com