HAMEG Instruments

SCPI Programmers Manual

for: HM1000x
HM1008x
HM1500x
HM1508x
HM2005-2
HM2008

for Firmware Version: 05.500-02.014
and later

English

Table of content

List of tables

List of figures

1. Introduction

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.

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 GPIB- (IEEE-488), RS232- , USB- and Ethernet-interfaces.

1.1.1. Common Commands

Common commands always start with a prefixed *(*)*.They are special system commands and are used without path declaration. They can also be present in command strings and are separated like other commands with semicolon e.g. *' *IDN? '*.

1.1.2. Program Commands

The program commands contain all instrument specific control commands and must be used with path declarations in accordance with the SCPI syntax.

1.1.3. 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 *(?)*. Path and parameters have to be specified additionally. Common commands are read without path declaration.

Example:	*Query for the current time base setting by the command*	*' :HOR:MAIN:SCAL? '*
	Query for instrument identification	*' *IDN? '*
	Query for the minimum possible time base setting	*' :HOR:MAIN:SCAL? MIN '*

1.1.4. 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. Construction of the SCPI Command Syntax

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.

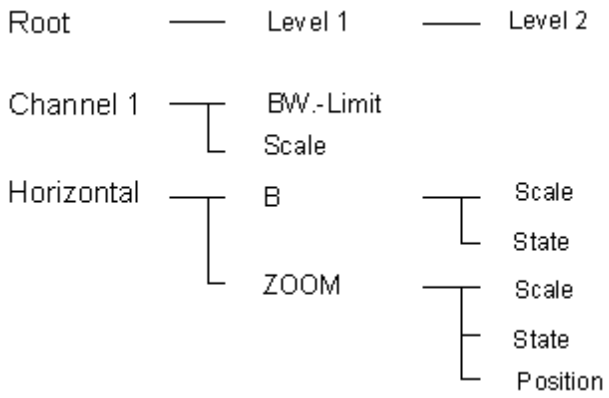

Figure 1: SCPI-rot (tree-structure)

1.2.1. Syntax

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:	*'Horizontal' menu selection*	
	short form	*:HOR*
	long form	*:HORizontal*
	acceptable spelling	*:HOR* *:HORizonTAL*
	unacceptable spelling	*:Horizont* *:Hori*

1.2.2. 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 .

1.2.3. Semicolon

A semicolon is used to separate commands from one another.

1.2.4. 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.

1.2.5. Comma

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

1.2.6. Data Formats

1.2.6.1. 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	123
- positive real number	12.34
- negative real number	-12.34
- with exponent $1.2 \cdot 10^{-3}$,	1.2E-3
- without leading zero (0.012)	.012

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

1.2.6.2. INF, NINF, NAN

To adapt an infinite range of values to the 32 bit floating point numbers (IEEE-754) the terms INF, NINF and NAN are implemented. They are delineated as follows:

INFinity	- positive infinity	-	9.9E37
NINFinity	- negative infinity	-	-9.9E37
NAN	-not defined[1], 'not a number'	-	9.91E37

1.2.6.3. 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.

1.2.6.4. Character

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

' :CHAN1:STAT ON '

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

[1] e.g. at division by zero, operation with ∞ etc.

1.2.6.5. 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><l bytes 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

#3456abcd .. ef

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

1.2.6.6. Special number formats

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

1.3 SCPI Commands - Examples

Example a) : Deactivation of channel 1 (OFF) and vertical scaling of channel 2 to 1V/DIV

:CHAN1:STAT OFF;:CHAN2:SCAL 1

:CHAN1 - selects root element 'Channel 1'
:STAT OFF - allocates 'OFF' to the parameter 'State' of 'CHAN1'
; - separates two commands
:CHAN2 - selects root element 'Channel 2'
:SCAL 1 - allocates '1' (V/DIV) as deflection coefficient of 'Channel 2'

Example b): Setting main time base to 1µs/DIV and query the setting

alternative 1: Explicit declaration of both commands

:HOR:MAIN:SCAL 1E-6;:HOR:MAIN:SCAL?

:HOR:MAIN - *‘ time base’ path selection*
:SCAL 1e-6 - *assigning 1µs to the time base*
:SCAL? - *query for the current time base (delivers 1.00E-6 in this example)*

alternative 2: Use of already existing paths

:HOR:MAIN:SCAL 1E-6;SCAL?

This command has the same effect as alternative 1, but uses the fact that the path ‘ :HOR:MAIN’ is present, so that the time base can be read with ‘ SCAL? ’.

Example c): Use of existing paths

Note: The multiple use of existing paths is only possible, if access to several end-elements of the path will be made one after another. It is not possible to change into same or higher levels and simultaneously use existing paths.

acceptable: :TRIG:A:EDGE:COUP DC;LEV MIN

The elements Coupling and Level are located in the same path.

unacceptable: :TRIG:A:EDGE:COUP DC;VID:FIEL?

The element VID uses the path ‘ :TRIGger:A’ but is not in the same level as EDGE and not in a sub level of EDGE. Thus the path must be input again.

':TRIG:A:EDGE:COUP DC;:TRIG:A:VID:FIEL?'

acceptable: :TRIG:A:EDGE:COUP DC;FILT:LPAS ON

As the element LPAS also uses the path ‘ :TRIGger:A:Edge’ it also can use the existing path.

1.4. 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

R	R	R	R	R	R	R	R
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC
bit 7							*bit 0*

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

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

bit 7 *bit 0*

The SESER determines which events are evaluated.

1 - *Event will be evaluated*
0 - *Event will be ignored*

SBR - State Byte Register

R	R	R	R	R	R	R	R
/	RQS / MSS	ESB	MAV	/	/	/	/

bit 7 *bit 0*

RQS - *Request Service* - *instrument requests to send*
MSS - *Master State 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

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
/	/	ESB	MAV	/	/	/	/

bit 7 *bit 0*

SRER determines which outputs may ask for permission to send

1 - *output permitted*
0 - *output prohibited*

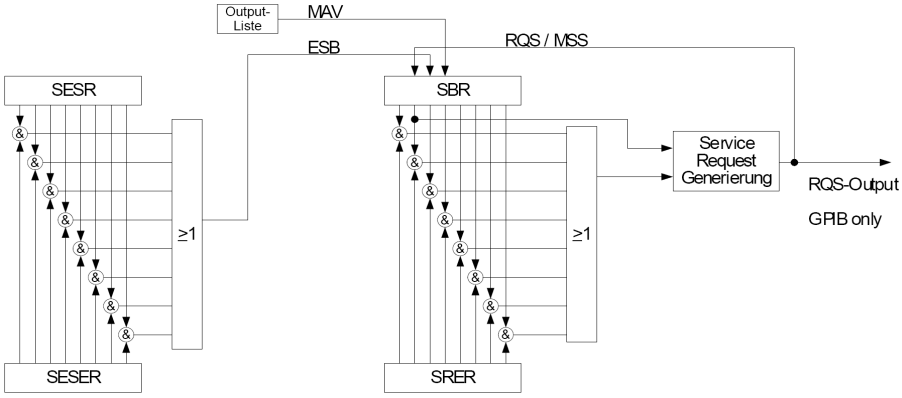

Figure 2: Event handling

The GPIB interface is the only interface with a hardware implemented line for request-to-send queries (RQS bit) directly available for the operator. The line state shows the appearance of an event. The other interfaces (RS232, USB and Ethernet) have no functions like this.

When event handling is used, the operator must check the instrument state periodically. This method is called *polling*. The polling result shows whether an event was present or not.

2. SCPI Commands

Symbols	Description
< >	*variable ,predefined element*
=	*equality, .. is identical with ..*
|	*or*
( )	*arrangement of elements, commentary*
[ ]	*optional elements*
{ }	*set of several elements*
' '	*example*

Table 1: BNF - Symbols

2.1. Common Commands

*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?* - *reads SESER content*

*ESR?

**ESR?* - *reads SESR (Standard Event State Register) content and resets it thereafter*

e.g.: *' 32 '*

*IDN?

**IDN?* - *returns the instrument identification*

e.g.: *' HAMEG,HM1508,000000000,HW10030000,SW05.100-02.005 '*

*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?* - *reads SRER*

*STB?

**STB?* - *returns SBR (State Byte Register) content*

2.2. Program Commands

2.2.1. Survey – Command Groups

Command group	Description
:Add	math. Hardware operations
:Acquire	contains signal acquisition elements
:Calibrate	instrument calibration functions
:Channel	contains all channel related settings
:Display	display control
:FFT	fast fourier transformation
:Hardcopy	Screenshot
:Horizontal	contains time base settings
:Instrument	contains instrument operation modes
:Logic	logic settings
:Math	mathematics functions, formula sets
:Measure	measuring function
:Pod	logic groups
:System	system settings
:Trace	signal curve readout functions
:Trigger	trigger options

Table 2: Command groups

2.2.2. ADD - Commands

Command	Write	Read	Description
:ADD	-	x	Survey of Add functions
:ADD:STATe <state>	x	x	<state> = { ON | OFF }

Table 3: Add - Commands

2.2.3. Acquire - Commands

Command	Write	Read	Description
:ACQuire	-	x	Summary of aquisition settings
:ACQuire:AVERage:COUNt <count>	x	x	<count> = {2 | 4 | .. | 512} , 2^n
:ACQuire:PEAKdetect <state>	x	x	<state> = {AUTO | OFF}
:ACQuire:REALtime <state>	x	x	<state> = {ON | OFF}
:ACQuire:STATe <state>	x	x	<state> = {RUN | STOP}
:ACQuire:TYPE <type>	x	x	<type> = {REFResh | ROLL | ENVelope | AVERage}

Table 4: Acquire - Commands

2.2.4. Calibrate - Commands

Command	Write	Read	Description
:CALibrate:SELF:ALL	x	x	On request a self calibration is made and the state is returned

Table 5: Calibrate - Commands

2.2.5. Channel - Commands

Command	Write	Read	Description
:CHANnel<n>	-	x	<n> = { 1 | 2 }
:CHANnel<n>:BWLimit <limit>	x	x	<limit> = {ON | OFF} (BW= Bandwidth) <n> = { 1 | 2 }
:CHANnel<n>:COUPling <coupling>	x	x	<coupling> = {AC | DC | GND} <n> = { 1 | 2 }
:CHANnel<n>:IMPedance <impedance>	x	x	<impedance> = {50 | 1000000} <n> = { 1 | 2 } 50 OHM termination only for HM2005-2, HM2008.
:CHANnel<n>:OFFSet <offset>	x	x	<offset> = {MINimum | MAXimum | floating point value in V} <n> = { 1 | 2 } HM2008 only
:CHANnel<n>:POLarity <polarity>	x	x	<polarity > = {NORMal | INVerted} <n> = { 1 | 2 }
:CHANnel<n>:PROBe <gain>	x	x	<gain > = {1 | 0.1 | 0.01 | 0.001} <n> = { 1 | 2 }
:CHANnel<n>:PROBe:AUTO <state>	-	x	<state> = {ON | OFF} <n> = { 1 | 2 }
:CHANnel<n>:POSition <position>	x	x	<position> = {MINimum | MAXimum | floating point value in DIV} <n> = { 1 | 2 }
:CHANnel<n>:SCALe <scale>	x	x	<scale > = {MINimum | MAXimum | floating point value in V/DIV} <n> = { 1 | 2 }
:CHANnel<n>:STATe <state>	x	x	<state> = {ON | OFF} <n> = { 1 | 2 }

Table 6: Channel - Commands

2.2.6. Display - Commands

Command	Write	Read	Description
:DISPlay	-	x	Summary od Display settings
:DISPlay:FORMat <format>	x	x	<format> = {YT | XY}
:DISPlay:STYLe <style>	x	x	<style> = {DOTS | VECTors | OPTimal}
:DISPlay:TEXT:INTensity <intensity>	x	x	<intensity> = {MINimum | MAXimum | floating point value in % }
DISPlay:TEXT:STATe <state>	x	x	<state> = {ON | OFF}
:DISPlay:TRACe:INTensity <intensity>	x	x	<intensity> = {MINimum | MAXimum | floating point value in % }
:DISPlay:TRACe:ZOOM:INTensity <intensity>	x	x	<intensity> = {MINimum | MAXimum | floating point value in % }

Table 7: Display - Commands

2.2.7. FFT - Commands

Command	Write	Read	Description
:FFT	-	x	Summary od FFT settings
:FFT<n>:POSition <position>	x	x	<position> = {MINimum | MAXimum | floating point value in DIV} <n> = {1 | 2}
:FFT<n>:SCALe <scale>	x	x	<scale > = {MINimum | MAXimum | floating point value in V/DIV} <n> = {1 | 2}
:FFT<n>:STATe <state>	x	x	<state> = {ON | OFF} <n> = {1 | 2}
:FFT<n>:UNIT <unit>	x	x	<unit> = {V | DB} <n> = {1 | 2}

Table 8: FFT - Commands

2.2.8. Hardcopy- Commands

Command	Write	Read	Description
:HCOPy	-	x	Survey of Hardcopy settings
:HCOPy:DATA	-	x	query delivers screenshot in block format
:HCOPy:FORMat <format>	x	x	<format> = { bmp }
:HCOPy:SIZE:X	-	x	query delivers picture width in pixel
:HCOPy:SIZE:Y	-	x	query delivers picture height in pixel

Table 9: Hardcopy - Commands

2.2.9. Horizontal - Commands

Command	Write	Read	Description
:HORizontal	-	x	Summary of horizontal settings
:HORizontal:MAIN:SCALe <scale>	x	x	<scale> = {MINimum | MAXimum | (floating point value in s/DIV)}
:HORizontal:B:SCALe <scale>	x	x	<scale> = {MINimum | MAXimum | (floating point value in s/DIV)}
:HORizontal:B:STATe <state>	x	x	<state> = {ONLY | SEARch | OFF}
:HORizontal:DELay:TIME <time>	x	x	<time> = {MINimum | MAXimum | (floating point value in s)}
:HORizontal:ZOOM:POSition <position>	x	x	<position> = {MINimum | MAXimum | (floating point value in %)}
:HORizontal:ZOOM:SCALe <scale>	x	x	<scale> = {MINimum | MAXimum | (floating point value in s/DIV)}
:HORizontal:ZOOM:STATe <state>	x	x	<state> = {ONLY | SEARch | OFF}
:HORizontal:FFT:CALCulate <type>	x	x	<type> = {NORMal | ENVelope | AVERage}
:HORizontal:FFT:CALCulate:AVERage:COUNt <no>	x	x	<no> = {2 | 4 | .. | 512} 2^n
:HORizontal:FFT:CENTer	x	x	= {MINimum | MAXimum | floating point value in Hz)}
:HORizontal:FFT:SPAN	x	x	= {MINimum | MAXimum | (floating point value in Hz)}
:HORizontal:FFT:WINDow <window>	x	x	<window> = {RECTangular | HAMMing | HANNing | BLACkman}
:HORizontal:FFT:ZOOM <zoom>	x	x	<zoom> = {NONE | TWICe | FIVefold}

Table 10: Horizontal - Commands

2.2.10. Instrument - Commands

Command	Write	Read	Description
:INSTrument:SELect <select>	x	x	<select> = {DSO | AO | CT | FFT}

Table 11: Instrument - Commands

2.2.11. Logic - Commands

Command	Write	Read	Description
:LOGic<n>	-	x	<n> = { 1 | 2 } - für HM1508, HM1508-2 (CH3, CH4) <n> = 1 | 2 | 3 | 4 - für HM2008 (LC0 | LC1 | LC2 | LC3)
:LOGic<n>:PROBe <gain>	x	x	<gain > = {1 | 0.1 | 0.01 | 0.001} <n> see :LOGic <n> Note: HM2008 query only
:LOGic<n>:PROBe:AUTO <state>	-	x	<state> = {ON | OFF} <n> see :LOGic <n> Note: HM2008 query only
:LOGic<n>:POSition <position>	x	x	<position> = {MINimum | MAXimum | floating point value in DIV} <n> see :LOGic <n> Note: HM2008 query only
:LOGic<n>:SIZE <size>	x	x	<size> = {SMALl | MEDium | LARGe} <n> see :LOGic <n> Note: HM2008 query only
:LOGic<n>:THReshold <threshold>	x	x	<threshold > = {TTL | CMOS | ECL | USER1 | USER2 | USER3} <n> = see :LOGic <n> USER3 only for HM1508/HM1508-2
:LOGic<n>:THReshold:UDLevel<u> <threshold>	x	x	<threshold > = {MINimum | MAXimum | (floating point value in V)} <n> = see :LOGic <n> <u> = { 1 | 2 | 3 } <u> = {3} only for HM1508/HM1508-2
:LOGic<n>:STATe <state>	x	x	<state> = {ON | OFF} <n> = see :LOGic <n>

Table 12: Logic - Commands

2.2.12. Mathematic - Commands

Command	Write	Read	Description
:MATH	-	x	Summary of math settings
:MATH:EXPRession<n> <expr>	x	x	<expr> = “MA<f> = <function> (<operand> [, <operand>]) [<expr.unit>]” with: <f>, <n> = {1 | 2 | 3 | 4 | 5} (<f> and <n> must be identical) <function> = {ADD | SUB | MUL | DIV | SQ | INV | 1/ | ABS | POS | NEG} <operand> = {CH<c> | MA<m> | <constant> } <c> = {1 | 2} <m> = {1 | 2 | 3 | 4} <constant> = positive floating point value [<unit>] <unit> = { dB | Hz | VA | Pa | etc. } <expr.unit> = [<unit>]

Table 13: Mathematic - Commands

2.2.13. Measure - Commands

Command	Write	Read	Description
:MEASure	-	x	Survey of auto measurement settings
:MEASure:TRACe:TYPE <type>	x	x	<type > = {FREQuency | PERiod | VPP | VMAX | VMIN | VAVerage | VRMS} *
:MEASure:TRACe:SOURce <source>	x	x	<source> = {CH<c> | ADD<a> | LOG<l> | RE<r> | MA<m>} *
:MEASure:TRACe:VALue	-	x	query delivers <result>,<error> *
:MEASure:TRIGger:TYPE <type>			<type> = {FREQuency | PERiod | VDC | VPP | VMAX | VMIN}
:MEASure:TRIGger:VALue	-	x	query delivers <result>,<error>

Table 14: Measure - Commands

* only for HM1008, HM1508x, HM2008 (digital mode)

2.2.14. Pod - Commands

Command	Write	Read	Description
:POD	-	x	Survey for logic pod
:POD:THReshold <threshold>	x	x	<threshold > = {TTL | CMOS | ECL | USER<u>} <u> = {1 | 2 }
:POD:THReshold:UDLevel<u> <threshold>	x	x	<threshold > = {MINimum | MAXimum | (floating point value in V)} <u> = { 1 | 2 }
:POD:STATe <state>	x	x	<state> = {ON | OFF}

Table 15: Pod - Commands

Pod commands only for HM2008

2.2.15. System - Commands

Command	Write	Read	Description
:SYSTem	-	x	Summary of system settings
:SYSTem:ERRor	-	x	query delivers the last instrument error and deletes the stack entry
:SYSTem:ELISt	-	x	query delivers all filled errors
:SYSTem:LANGuage <language>	x	x	<language> = {ENGLish | GERMan | FRENch | SPANish}
:SYSTem:LOCK <state>	x	x	<state> = {ON | OFF}, keylock
:SYSTem:NAME <name>	x	x	<name> = instrument name in string format

Command	Write	Read	Description
:SYSTem:SNUMber	-	x	query delivers the instrument serial number
:SYSTem:SOFTware	-	x	<SW-Scope>-<SW-MesOS>, Software
:SYSTem:HARDware	-	x	Hardware-ID 32-Bit '#Hxxxxxxxx'
:SYSTem:DEVice	-	x	e.g. 'HM1508'
:SYSTem:SET <set>	x	x	<set> system settings in block format
:SYSTem:SET:AUTO	x	-	activates the oscilloscope AUTOSET function
:SYSTem:SET:CHANged <change>	-	x	<change> = { TRUE | FALSe }
:SYSTem:FPANel:RECall <setNO>	x	-	<setNO> = {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}
:SYSTem:FPANel:SAVE <setNO>	x	-	<setNO> = {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}

Table 16: System - Commands

2.2.16. Trace - Commands

Command	Write	Read	Description
:TRACe	-	x	Trace data with informations to selected trace
:TRACe:BORDer <byte_order>	x	x	<byte_order> = {MSBFirst | LSBFirst}
:TRACe:DATA	_	x	trace data
:TRACe:FORMat <format>	x	x	<format> = {BYTE | WORD | ASCii | CSV}
:TRACe:POINts <points>	x	x	<points> = {MAXimum | DEFault}
:TRACe:REFerence<r>:SAVE <source>	x		<r> = { 1 | 2 | .. | 9 } <source> = { CH1 | CH2 | ADD1 }
:TRACe:SOURce <source>	x	x	<source> = {CH<n> | ADD1 | FFT<n> | POD1 | ZCH<n> | ZADD1 | ZPOD1 | MA<m> | RE<r>} <n> = 1, 2 <m> = 1 – 5 <r> = 1 – 9
:TRACe:SOURce:CATalog	-	x	list of currently available functions/curves
:TRACe:TYPE	-	x	{NORMAL | MINMAX}
:TRACe:XINCrement	-	x	time step / frequency step between the data (samples/points)
:TRACe:XORigin	-	x	time/frequency of the first displayed sample/point with respect to the trigger point (time) / center frequency
:TRACe:XREFerence	-	x	{0} - 1. data point of the display (left border)
:TRACe:XUNit	-	x	unit of horizontal axis
:TRACe:YINCrement	-	x	Y resolution in Volt/Bit
:TRACe:YORigin	-	x	voltage value at the screen centre
:TRACe:YREFerence	-	x	binary value at the screen centre
:TRACe:YRESolution	-	x	vertical graticule resolution in Bit/DIV
:TRACe:YDISplacement	-	x	additionally offset of the graph in DIV
:TRACe:YUNit	-	x	unit of vertical axis

Table 17: Trace - Commands

2.2.17. Trigger - Commands

Command	Write	Read	Description
:TRIGger	-	x	Summary of trigger settings
:TRIGger:A:EDGE:FILTer:LPASs <state>	x	x	<state> = {ON | OFF}
:TRIGger:A:EDGE:FILTer:NREJect <state>	x	x	<state> = {ON | OFF}
:TRIGger:A:EDGE:SOURce <source>	x	x	<source> = {CH1 | CH2 | CH3 | CH4 | EXT | LINE | ALTernating} HM 1508 <source> = {CH1 | CH2 | LC0 | LC1 | LC2 | LC3 | EXT | LINE | ALTernating } HM 2008
:TRIGger:A:EDGE:SLOPe <slope>	x	x	<slope> = {POSitive | NEGative | EITHer}
:TRIGger:A:EDGE:COUPling <coupling>	x	x	<coupling> = {AC | DC | HF}
:TRIGger:A:EDGE:LEVel <level>	x	x	<level> ={MINimum | MAXimum | floating point value in V}
:TRIGger:A:LOGic:FUNCtion <function>	x	x	<function> = {AND | OR}
:TRIGger:A:LOGic:WHEN <condition>	x	x	<condition> = {TRUE | FALSE}
:TRIGger:A:LOGic:SOURce <string>	x	x	<string> = "<CH1><CH2><CH3><CH4>" HM 1508x <string> = "<LC0><LC1><LC2><LC3>" HM 2008 <CH1>,<CH2>,<CH3>,<CH4>,<LC0>,<LC1>,<LC2>,<LC3> = { X | H | L }
:TRIGger:A:MODE <mode>	x	x	<mode> = {AUTO | NORMal | SINGle}
:TRIGger:A:TYPE <type>	x	x	<type> = {EDGE | VIDeo | LOGic}
:TRIGger:A:VIDeo:FIELd <field>	x	x	<field> = {ODD | EVEN | ALL | LINE | ALLLine}
:TRIGger:A:VIDeo:LINE <line>	x	x	<line> = {MINimum | MAXimum | line number}
:TRIGger:A:VIDeo:STANdard <standard>	x	x	<standard> = {PAL | NTSC}
:TRIGger:A:VIDeo:POLarity <polarity>	x	x	<polarity> = {POSitive | NEGative}
:TRIGger:A:VIDeo:SOURce <source>	x	x	<source> = {CH1 | CH2 | EXT}

Table 18: Trigger - Commands

3. SCPI Commands (detailed)

:ADD?

:ADD?
A query delivers an ADD function survey

e.g.: *' :ADD:STAT OFF; '*

:ADD:STATe <state>

<state> = {ON | OFF}

<state> - enables, disables the hardware signal addition of CH1 and CH2

Note: If the hardware addition is activated CH1 and CH2 cannot be displayed separately.

:ACQuire?

:ACQuire?

A query delivers all signal acquisition relevant settings (only available in digital mode)

e.g.: *' :ACQ:STAT RUN;:ACQ:TYPE REFR;*
:ACQ:AVER:COUN 2.00;:ACQ:REAL OFF;:ACQ:PEAK AUTO; '

:ACQuire:AVERage:COUNt <count>

<count> = {2 | 4 | .. | 512} ($2^n, n \in \mathbb{N}$)

<count> *- Number of acquisitions to be averaged.*

:ACQuire:AVERage:COUNt ? *returns the following Data { 2.00 | 4.00 | 8.00 | 16.0 | 32.0 | 64.0 | 128 | 256 | 512}*

Note: If user sends floating point values to the device, therer will be an automatically rounding.

:ACQuire:PEAKdetect <state>

<state> = {AUTO | OFF}

<state> *Activates/deactivates peak detection. In this mode the maximum sampling rate is always used so that samples are taken at times when normally no samples are taken. Then the minimum and maximum signal values are detected.*

:ACQuire:REALtime <state>

<state> = {ON | OFF}

<state> *- switches the signal acquisition between real time sampling and random sampling.*

:ACQuire:STATe <state>

<state> = {RUN | STOP | COMPlete* }

<state> *describes the signal acquisition state*

'RUN' *- starting of signal acquisition / .. is active*
'STOP' *- stopping of signal acquisition / .. will be finished*
'COMPlete' *- signal acquisition has ended*

** only available as a query reply.*

Note: *The answer 'Complete' on a query indicates that the last signal recording has ended. After the 'Stop' command the current signal acquisition will continue until the end is reached. Before the end is reached the query ' :ACQuire:STATe? ' causes the reply 'STOP'. After the acquisition is completed the query will reply 'COMPlete'.*

:ACQuire:TYPE <type>

<type> = {REFResh | ROLL | ENVelope | AVERage}

<type> *Determine a mode for digital signal acquisition.*

'REFResh' *- Normal acquisition mode – A new trigger event causes a new signal acquisition and display. The previous acquisition values get lost.*
'ROLL' *- Roll mode – The signal acquisition is made continuously and independent from a trigger event. New recorded data are first displayed at the right display border and shifted to the left.*
'ENVelope' *- Envelope mode – Several signal acquisitions are analysed for minimum and maximum values for each sampling point in time. Thereafter the signal display consists only of these minima and maxima.*

Information: This mode eases the recognition of amplitude, phase and frequency changes.

'AVERage' *- Average display - In this mode an average is made from the samples of several acquisitions taken at the same point in time.*

:CALibrate

:CALibrate:SELF:ALL

:CALibrate:SELF:ALL

... triggers the instrument self calibration function. The query (:CALibrate:SELF:ALL?) causes additionally the reply of errors which possibly happened during the self calibration process (<number>,<message>) .

<number> - error number
<message> - error message

e.g.: successful calibration: ' 0,"No error" '

calibration not successful: ' 1109,"upper limit DC Offset adjustment CH:" '

:CHANnel<n>?

<n> = { 1 | 2 }

The query of channel <n> causes the reply of all parameters belonging to the channel.

A possible reply on ' :CHANnel1 ? ' can be:

' :CHAN:COUP DC;:CHAN:POL NORM;:CHAN:POS -119.658E-3;:CHAN:PROB 1.00;
:CHAN:PROB:AUTO OFF;:CHAN:SCAL 500E-3;:CHAN:STAT ON;
:CHAN:BWL OFF;:CHAN:IMP 1.00E6; '

:CHANnel<n>:BWLimit <limit>

<n> = { 1 | 2 }
<limit> = { ON | OFF }

' ON ' activates the measuring amplifier 20 MHz bandwidth limit

:CHANnel<n>:COUPling <coupling>

<n> = { 1 | 2 }
<coupling> = { AC | DC | GND }

' AC ' - *signal is displayed with suppressed DC content*
' DC ' - *signal is displayed with DC and AC content*
' GND ' - *The measuring signal will not be displayed as the measuring amplifier is "grounded" (trace without Y deflection).*

Note : *On condition GND the trace position is indicated by a GND symbol.*

:CHANnel<n>:IMPedance <impedance>

<n> = { 1 | 2 }
<impedance> = { 50 | 1000000 }

<impedance> - *determines the (instrument internal) input termination resistor.*

Note: The internal 50 Ohm termination is only available for HM2005-2 and HM2008.

:CHANnel<n>:OFFSet <offset>

<n> = { 1 | 2 }
<offset> = { MINimum | MAXimum | floating point value in V}

<offset> - *determines a measuring amplifier DC offset.*

Note: only for HM2008

:CHANnel<n>:POLarity <polarity>

<n> = { 1 | 2 }
<polarity > = {NORMal | INVerted }

' NORMal ' - *Normal signal display.*
' INVerted ' - *Inverted signal display (inversion is made by hardware).*

:CHANnel<n>:PROBe <gain>

<n> = { 1 | 2 }
<gain > = { 1 | 0.1 | 0.01 | 0.001 }

<gain> *Description of a probe's "gain" factor (with internal parameter correction). A divider ratio of 1/10 (x10) corresponds with '0.1' for the 'PROBe' parameter.*

Note: *These settings are ineffective if ' :CHANnel<n>:PROBe:AUTO ' is activated (ON).*

:CHANnel<n>:PROBe:AUTO?

<n> = { 1 | 2 }
<state> = { ON | OFF }

If a probe with an automatic identification is attached, the instrument delivers <state> as a reply.

:CHANnel<n>:POSition <position>

<n> = { 1 | 2 }
<position> = {MINimum | MAXimum | floating point value in DIV}

This command enables the vertical positioning of the signal display; referring to the GND trace position. '0 DIV' sets the trace to the screen centre (horizontal graticule centre line is the reference position).

Note: *The channel Y position reply - output in DIV - refers to the trace Y reference position.*

:CHANnel<n>:SCALe <scale>

<n> = { 1 | 2 }
<scale > = {MINimum | MAXimum | floating point value in V/DIV}

<scale> *determines the vertical scaling in V/DIV.*

Note: *The reply is carried out in V/DIV.*

:CHANnel<n>:STATe <state>

<n> = { 1 | 2 }
<state> = {ON | OFF }

<state> *activates/deactivates the selected channel.*

:DISPlay?

:DISPlay?

:DISPlay? delivers the current display parameters.

e.g.: *' :DISP:TEXT:STAT ON;:DISP:TEXT:INT 35.0;:DISP:FORM YT;*
:DISP:TRAC:INT 28.2431;:DISP:TRAC:ZOOM:INT 71.6216;:DISP:STYL OPT; '

:DISPlay:FORMat <format>

<format> = {YT | XY}

<format> *determines the channels display mode.*

' YT ' *- CH1 and CH2 displayed as a time dependent function f(t).*

Y deflection by CH1 / CH2
X deflection by internal time base and possible trigger options.

' XY ' *- CH1 and CH2 displaying in XY mode.*

CH1 *- deflection in X direction*
CH2 *- deflection in Y direction*

Note: XY mode can be used to get information about amplitude, phase or frequency differences of two signals (LISSAJOUS figures).

:DISPlay:STYLe <style>

<style> = {DOTS | VECTors | OPTimal}

<style> *contains signal display modes.*

'DOTS' *- Signal values are displayed as dots.*
'VECTors' *- Signal values are joined (interpolate).*
'OPTimal' *- up to 1 MByte samples are taken with the maximum sampling rate. Thereafter only minimum and maximum values are taken into consideration and displayed in vectors mode.*

:DISPlay:TEXT?

:DISPlay:TEXT?

.. contains parameter of the additional texts.

e.g.: *' :DISP:TEXT:STAT ON;:DISP:TEXT:INT 30.0; '*

:DISPlay:TEXT:INTensity <intensity>

<intensity> = {MINimum | MAXimum | floating point value in %}

<intensity> *sets brightness of additional displayed text.*

:DISPlay:TEXT:STATe <state>

<state> = {ON | OFF}

<state> *switches additional text On or OFF.*

:DISPlay:TRACe ?

:DISPlay:TRACe ? returns data for trace intensities

:DISPlay:TRACe:INTensity <intensity>

<intensity> = {MINimum | MAXimum | floating point value in %}

<intensity> sets brightness of signal display.

:DISPlay:TRACe:ZOOM:INTensity <intensity>

<intensity> = {MINimum | MAXimum | floating point value in %}

<intensity> *sets brightness of the sector to be zoomed in the original signal display and in the zoomed signal display.*

:FFT<n>?

:FFT<n>?

<n> = { 1 | 2 }

<n> - number of analogue channel

A query delivers all Fast-Fourier-Transformation parameters
e.g.: ' :FFT:STAT ON;:FFT:UNIT V;:FFT:SCAL 50.00E-3;:FFT:POS -2.50; '

:FFT<n>:POSition <position>

<n> = { 1 | 2 }
<position> = { MINimum | MAXimum | floating point value in DIV}

<position> - determines the vertical FFT display position.

:FFT<n>:SCALe <scale>

<n> = { 1 | 2 }
<scale> = { MINimum | MAXimum | floating point value */DIV }

<scale> - describes Y scaling of the screen

** see :FFT<n>:UNIT <unit>*

:FFT<n>:STATe <state>

<n> = { 1 | 2 }
<state> = { ON | OFF}

<state> - switches the FFT display of the current channel ON or OFF.
However the device will stay FFT-mode.

:FFT<n>:UNIT <unit>

<n> = { 1 | 2 }
<unit> = { V | DB }

<unit> - determines the FFT mode unit and consequently the display.

' V ' - signal display scaling in Volt (V_{rms})
' DB ' - signal display scaling in dBV (level with reference value 1V)

:HCOPy?

:HCOPy?

A query delivers the basic data for a screen shot.

Query: *' :HCOPy? '*
Reply: *' :HCOP:FORM BMP;:HCOP:SIZE:X 550;:HCOP:SIZE:Y 550; '*

:HCOPy:DATA?

:HCOPy:DATA?

A query delivers a screen shot in block format.

:HCOPy:FORMat <format>

<format> = { BMP }

Hinweis: *At the moment a hardcopy is only possible in bitmap format (*.bmp).*

:HCOPy:SIZE:X?

:HCOPy:SIZE:X?

A query delivers the number of pixels in X direction as unsigned integer number.

:HCOPy:SIZE:Y?

:HCOPy:SIZE:Y?

A query delivers the number of pixels in Y direction as unsigned integer number.

:HORizontal?

:HORizontal?

A query delivers all horizontal menu related parameters.

e.g.: *' :HOR:DEL:TIME 0;:HOR:MAIN:SCAL 10.0E-6;*
:HOR:B:SCAL 5.00E-6;:HOR:B:STAT OFF;
:HOR:ZOOM:SCAL 5.00E-6;:HOR:ZOOM:POS 50.0;
:HOR:ZOOM:STAT OFF; '

:HORizontal:MAIN?

:HORizontal:MAIN:SCALe <scale>

<scale> = {MINimum | MAXimum | floating point value in s/DIV}

<scale> *Time base setting.*

:HORizontal:B?

A query delivers all analogue mode search function parameters.

:HORizontal:B:SCALe <scale>

<scale> = {MINimum | MAXimum | floating point value in s/DIV }

<scale> *sets the zoom time base and the width of the sector (on the original signal display) to be zoomed.*

:HORizontal:B:STATe <state>

<state> = {ONLY | SEARch | OFF}

<state> *determines the search mode in analogue mode.*

'Only' *zoom display of the previously selected signal part (sector).*
'Search' *display of original signal curve and zoomed signal part (sector).*
'Off' *search function deactivated (only the original signal curve will be displayed).*

:HORizontal:DELay?

:HORizontal:DELay:TIME <time>

<time> = { MINimum | MAXimum | floating point value in s }

Note: *In digital mode the trigger point is indicated by a cross symbol. The delay time is the distance between trigger point and vertical graticule line in the screen centre serving as reference (0 s) point. Please note that the delay time depends on the time base setting. The delay time enables the display of signal curves before and after the trigger event (point) which would not be visible without this function. If the trigger point is left of the screen centre a delay time query will cause a reply with a negative sign.*

:HORizontal:FFT?

:HORizontal:FFT:CALCulate <type>

<type> = { NORMal | ENVelope | AVERage}

<type> *- determines the signal display.*

'Normal' *- each acquisition causes a new calculation and signal display (refresh mode)*
'Envelope' *- causes the display of all acquisitions minima and maxima.*
'Average' *- A predefined number of acquisitions are averaged. The weighting depends on the selected parameter (number). Only values of the same frequency are averaged.*

:HORizontal:FFT:CALCulate:AVERage:COUNt <no>

<no> = { 2 | .. | 512} $2^n, n \in \mathbb{N}$

<no> *- determines the weighting parameter (number) during averaging*

:HORizontal:FFT:CENTer

 = { MINimum | MAXimum | floating point value in Hz}

* * *- centre frequency (frequency displayed in the screen centre)*

:HORizontal:FFT:SPAN

 = { MINimum | MAXimum | floating point value in Hz}

** - *determines the frequency range to be displayed between the left and the right graticule border on condition ZOOM inactive.*

:HORizontal:FFT:WINDow <window>

<window> = { RECTangular | HAMMing | HANNing | BLACkman }

<window> - *contains the window functions for weighting the time domain samples in the FFT calculation process.*

:HORizontal:FFT:ZOOM <zoom>

<zoom> = { NONE | TWICe | FIVefold }

<zoom> - *contains the ZOOM rate of the FFT display*

' None ' - *original display (without ZOOM)*
' Twice ' - *twofold ZOOM*
' FIVefold ' - *fivefold ZOOM*

Note: *By the use of the ZOOM function the displayed frequency range changes.*

Example: *Span frequency 20MHz : :HOR:FFT:SPAN 20e6*
5fold ZOOM : :HOR:FFT:ZOOM FIV

The displayed span is $\frac{20\text{MHz}}{5} = 4\text{MHz}$.

The HORIZONTAL knob can be used to change the centre frequency so that it is possible to freely display selectable parts of the complete 20 MHz spectrum with a span of 4 MHz.

:HORizontal:ZOOM?

A query delivers all ZOOM function regarding parameter in digital mode.

:HORizontal:ZOOM:POSition <position>

<position> = {MINimum | MAXimum | floating point value in %}

<position> *defines the sector of the original signal to be zoomed.*

Note: *The range between the left and the right vertical graticule border lines is defined as 0% for the left and 100% for the right border line.*

:HORizontal:ZOOM:SCALe <scale>

<scale> = {MINimum | MAXimum | floating point value in s/DIV }

<scale> *defines the time base respectively the size of the sector to be zoomed.*

:HORizontal:ZOOM:STATe <state>

<state> = {ONLY | SEARch | OFF}

<state> *determines the search modus in digital mode.*

'Only' *only the zoomed signal is displayed.*
'Search' *original- and ZOOM curve are displayed.*
'Off' *the search function is deactivated. Only the original signal is displayed.*

:INSTrument?

:INSTrument? replies with the oscilloscopes operating mode.

:INSTrument:SELect <select>

<select> = { DSO | AO | CT | FFT }

<select> *defines the current instrument mode.*

'DSO' *- oscilloscope in digital mode*
'AO' *- oscilloscope in analogue mode*
'CT' *- oscilloscope in component test mode*
'FFT' *- oscilloscope in FFT mode*

:LOGic<n>?

<n> = { 1 | 2 } HM 1508, HM1508-2 (CH3, CH4)

<n> = { 1 | 2 | 3 | 4 } HM2008 (LC0, LC1, LC2, LC3)

returns all logic channel pertaining parameters.

Example : *' :LOGic!? ' (HM 1508-2)*

Reply: *' :LOG:PROB 1.00;:LOG:PROB:AUTO OFF;:LOG:THR TTL;:LOG:THR:UDL 0;*
:LOG:POS -1.998;:LOG:SIZE SMAL;:LOG:STAT OFF;'

:LOGic<n>:PROBe <gain>

<n> = { 1 | 2 } HM 1508, HM1508-2 (CH3, CH4)

<n> = { 1 | 2 | 3 | 4 } HM2008 (LC0, LC1, LC2, LC3)

<gain > = {1 | 0.1 | 0.01 | 0.001 }

<gain> *Description of a probes "gain" factor (with internal parameter correction).*
A divider ratio of 1/10 (x10) corresponds with '0.1' for the 'PROBe' parameter.

Note: *These settings are ineffective if ' :LOGic<n>:PROBe:AUTO ' is activated (ON).*

HM2008: only query

:LOGic<n>:PROBe:AUTO <state>

<n> = { 1 | 2 } HM 1508, HM1508-2 (CH3, CH4)

<n> = { 1 | 2 | 3 | 4 } HM2008 (LC0, LC1, LC2, LC3)

<state> = {ON | OFF}

<state> *determines the state of each logic channel.*

Note: *The instrument sets the parameter automatically if a probe with identification contact is connected.*

HM2008: only query

:LOGic<n>:POSition <position>

<n> = { 1 | 2 } HM 1508, HM1508-2 (CH3, CH4)

<n> = { 1 | 2 | 3 | 4 } HM2008 (LC0, LC1, LC2, LC3)

<position> = { MINimum | MAXimum | floating point value in DIV }

determines the logic signals Y position

HM2008: only query

:LOGic<n>:SIZE <size>

<n> = { 1 | 2 } HM 1508, HM1508-2 (CH3, CH4)

<n> = { 1 | 2 | 3 | 4 } HM2008 (LC0, LC1, LC2, LC3)

<size> = { SMALl | MEDium | LARGe }

determines the logic signal display height.

HM2008: only query

:LOGic<n>:THReshold <threshold>

<n> = { 1 | 2 } HM 1508, HM1508-2 (CH3, CH4)

<n> = { 1 | 2 | 3 | 4 } HM2008 (LC0, LC1, LC2, LC3)

<threshold > = { TTL | CMOS | ECL | USER1 | USER2 | USER3 }

<threshold> *determines the trigger level (signal height necessary for triggering). Use standard logic levels or user definable logic levels.*

User defined logic levels can be defined for each logic input <n> (HM1508, HM1508-2).

USER3 only available with HM 1508, HM1508-2

:LOGic<n>:THReshold:UDLevel<u> <threshold>

<n> = { 1 | 2 } HM 1508, HM1508-2 (CH3, CH4)

<n> = { 1 | 2 | 3 | 4 } HM2008 (LC0, LC1, LC2, LC3)

<threshold > = { MINimum | MAXimum | floating point value in V }

<u> = { 1 | 2 | 3 }

Note: *This command allows assignment of a trigger level (<threshold>) for different users <u> (USER1, USER2 and USER3). The currently selected trigger level (LOGic<n>:THReshold <threshold>) is not affected as long as none of the 3 user settings is activated.*

<u> = 3 only available with HM1508, HM1508-2

:LOGic<n>:STATe <state>

<n> = { 1 | 2 } HM 1508, HM1508-2 (CH3, CH4)

<n> = { 1 | 2 | 3 | 4 } HM2008 (LC0, LC1, LC2, LC3)

<state> = { ON | OFF }

<state> *switches the logic channels ON or OFF*

:MATH?

:MATH?

A query delivers all mathematical functions in the actual formula set.

e.g.

```
'  :MATH:EXPR "MA1 = SUB(CH1, CH2) [V]";
   :MATH:EXPR2 "MA2 = DIV(MA1, 0.001) [A]";
   :MATH:EXPR3 "MA3 = MUL(MA2, CH2) [W]";
   :MATH:EXPR4 "MA4 = DIV(MA1, MA2) [O]";
   :MATH:EXPR5 "MA5 = SQ(0)";   '
```

:MATH:EXPRession<n>

<n> = { 1 | 2 | 3 | 4 | 5 }

<n> *A query delivers the mathematical expression of the mathematic function <n> .*

Example: *' :MATH:EXPR1? '*
Reply: *' "MA1 = SQ(CH2)" '*

:MATH:EXPRession<n> <expr>

<n> = { 1 | 2 | 3 | 4 | 5 }

<n> *represents the mathematic function number*
<expr> *represents the mathematical expression (formula)*

Note: *The expression <expr> is a complete parameter. It will be transferred to the mathematic function <n> in string format (in ""signs).*

<expr> = "MA<f> = <function> (<operand_1> [, <operand_2>]) [<expr.unit>]"

<f> = <n> = { 1 | 2 | 3 | 4 | 5 }

<function> = {ADD | SUB | MUL | DIV | SQ | INV | 1/ | ABS | POS | NEG}

<operand_1>, <operand_2> = {CH<c> | MA<m> | <constant> }

<c> = {1 | 2}
<m> = {1 | 2 | 3 | 4}
<constant> = { positive floating point value [<unit>] }
<unit> = { dB | Hz | VA | Pa | etc. }
<expr.unit> = { [<unit>] }

<f>, <n> *- number of the mathematical expression*
<function> *- mathematical operation*
<operand> *- operands*
<c> *- channel number*
<m> *- number of a mathematic function*

Note: *By use of a present mathematic function as an operand the number <m> must always be at least 1 less than <f> respectively <n>*
(1 <= <m> < <f>,<n>). Thus the use of a mathematic function as an operand for the function MA1 is forbidden.

<constant> *- constant*
<unit> *- physical unit; the declaration of a unit is optional. The unit must be set in square brackets ([]).*

Note: *All characters of the alphabet can be used.*

<expr.unit> *- physical unit*

Note: The unit of a mathematic function (<expr.unit>) and the unit of a constant (<unit>) don't have to be identical.

	Operation		Operation
ADD	Addition	INV	Inversion
SUB	Subtraction	1/	Reciprocal (value)
MUL	Multiplication	ABS	Absolute (value)
DIV	Division	POS	Positive (fraction).
SQ	Square	NEG	Negative (fraction)

Table 19: Mathematical Operation

Note: *Subtraction is made in the following way* *MA<f> = <operand_1> - <operand_2>.*
Division is made in the following way *MA<f> = <operand_1> / <operand_2>*

Example: Measuring the power consumption of a load

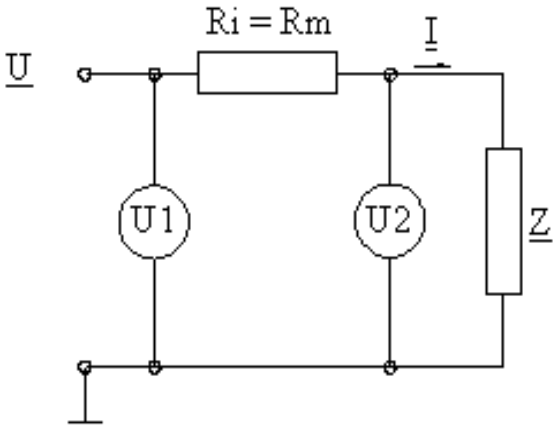

Figure 3: Power consumption measurement

Task: The power consumption of an unknown load with the impedance "Z" has to be measured. The measurement of the momentary power consumption can be performed with instruments such as multimeter etc. . The internal resistance (Ri) must be taken into account; the measuring lead resistance can be neglected.

As oscilloscopes cannot measure current, the power consumption must be determined using voltages. Thus the current will be determined by the voltage drop across a measuring resistor R_m, inserted in the current path.

Given: $R_m = R_i = 1\,\mathrm{m}\Omega$

$\underline{U}_1$ - channel 1

$\underline{U}_2$ - channel 2

All mathematic functions will be written in the formula set currently (manually) selected.

Demanded: $S_Z(t)$, display of the effective and the momentary apparent power.

ansatz:

1.)Preparations for measurement

As mathematic signals can only be activated/deactivated manually, press the MATH pushbutton of the oscilloscope and thereafter select mathematic curve MA3 in submenu “Display”.

Please activate the automatic measurement function of the oscilloscope by pressing the AUTOMEASURE pushbutton.

2.) Current consumption determination, $I_{R_m} = \frac{U_{R_m}}{R_m}$, $U_{R_m} = \Delta U = U_{CH1} - U_{CH2}$

SCPI ansatz: ΔU - mathematic function 'MA1'

:MATH:EXPR1 "MA1 = SUB (CH1,CH2) [V]" - unit: 1V

I_{R_m} - mathematic function 'MA2'

:MATH:EXPR2 "MA2 = DIV(MA1, 0.001) [A]" - unit: 1A

3.) Power determination $S_Z = U \cdot I$

SCPI ansatz: S_Z - mathematic function 'MA3'

:MATH:EXPR3 "MA3 = MUL(MA2, CH2) [W]" - unit: 1W

4.) Display of the effective power consumption using the oscilloscope.

SCPI ansatz:

:MEAS:TRAC:SOUR MA3 - selection of the power curve to be used for auto measure

:MEAS:TRAC:TYPE VRMS - selection of the 'rms value' function.

Now the power consumption and its rms value are displayed in the time domain.

:MEASure?

:MEASure?

A query delivers the current auto measurement settings

e.g.:	*' :MEAS:TRAC:TYPE FREQ;*	*- frequency measurement*
	:MEAS:TRAC:SOUR CH1;	*- source: Channel 1*
	:MEAS:TRAC:VAL 20242.917376,2000; '	*- result ' ~ 20,24 kHz ' , valid*

:MEASure:TRACe?

All functions belonging to the group :MEASure:TRACe can be used only in digital mode

:MEASure:TRACe:TYPE <type>

<type > = {FREQuency | PERiod | VPP | VMAX | VMIN | VAVerage | VRMS}

<type> *determines the measuring function*

'frequency'	*- frequency measurement of the source signal*
'period'	*- measurement of the signal period*
'VPP'	*- peak to peak voltage measurement*
'Vmax'	*- maximum measurement*
'Vmin'	*- minimum measurement*
'Vaverage'	*- calculation of the arithmetic average value*
'Vrms'	*- calculation of the rms value*

:MEASure:TRACe:SOURce <source>

<source> = {CH<c> | ADD<a> | LOG<l> | RE<r> | MA<m>}

<source> *source selection for signal measurement*

<c> = {1 | 2}	*- measuring channel <c>*
<a> = {1}	*- CH1 and CH2 addition mode*
<l> = {1 | 2}	*- logic channel <l>, HM 1508x*
<l> = {1 | 2 | 3 | 4 }	*- logic channel <l> HM 2008*
<r> = {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}	*- reference curve <r>*
<m> = {1 | 2 | 3 | 4 | 5}	*- mathematic function <m>*

Note: *Measurement can only take place on displayed curves.*

:MEASure:TRACe:VALue?

:MEASure:TRACe:VALue?

A query delivers a reply in the following way *<result>,<error>*

<result> - *measurement result in floating point format (decimal point as delimiter)*
<error> - *measurement result status (error number '2000' indicates a valid result). This error code number will not be filed on the stack.*

:MEASure:TRIGger?

:MEASure:TRIGger?

This query only replies, if a measurement of the trigger is done.
e.g.: *' :MEAS:TRIG:TYPE FREQ;:MEAS:TRIG:VAL 0.000000,2001; '*

:MEASure:TRIGger:TYPE <type>

<type> = {FREQuency | PERiod | VDC | VPP | VMAX | VMIN}

<type> *determines the trigger signal measurement*

'frequency' - *triggers frequency*
'period' - *triggers period*
'VDC' - *dc content*
'VMAX' - *maximum*
'VMIN' - *minimum*

:MEASure:TRIGger:VALue?

:MEASure:TRIGger:VALue?

A query delivers the reply in the following way *<result>,<error>*

<result> - *measurement result in floating point format (decimal point as delimiter)*
<error> - *measurement result status (error number '2000' indicates a valid result). This error code number will not be filed on the stack.*

:POD?

:POD?

A query delivers all Logic-Pod related parameters.

Note: At the time only available for HM2008.

:POD:THReshold <threshold>

<threshold> = { TTL | CMOS | ECL | USER<u> }

<u> = { 1 | 2 }

<threshold> - sets the trigger level for all Logic-Pod inputs.

:POD:THReshold:UDLevel<u> <treshold>

<threshold> = { MINimum | MAXimum | floating point value in V}

<u> = { 1 | 2 }

<threshold> - determines the user specific threshold <u>.

:POD:STATe <state>

<state> = { ON | OFF }

<state> - activates/deactivates the logic group (Pod)

:SYSTem?

:SYSTem?

A query causes the reply of the system settings.

e.g.: ' :SYST:LANG GERM;:SYST:NAME "HAMEG COMBISCOPE";:SYST:LOCK OFF; '

:SYSTem:ERRor ?

:SYSTem:ERRor?

A query causes the last error to be displayed and deletes it on the stack.

A query delivers the reply in the following way <number>,<message>.

<number> - *error code number*
<message> - *error code number in string format*

e.g.: ' 0,"No error" '

:SYSTem:ELISt?

:SYSTem:ELIst?

A query causes the complete error stack to be read out and thereafter being deleted from the stack.

A query delivers the reply in the following way <number>,<message>

:SYSTem:LANGuage <language>

<language> = { ENGLish | GERMan | FRENch | SPANish }

<language> - *Selects the oscilloscope language regarding the menu and setting display.*

:SYSTem:LOCK <state>

<state> = {ON | OFF}

<state> *contains the key lock function of the instrument (locks all keys)*

Note: *The instrument can be protected against accidental key board operation. Then the instrument can only be remote controlled.*

:SYSTem:NAME <name>

<Name> assigns your instrument a user specific name.

The parameter <name> will be delivered as a string with maximal 20 characters length.

e.g. *' :SYSTem:NAME „HAMEG“'*

:SYSTem:SNUMber?

:SYSTem:SNUMber?

A query delivers the instruments serial number.

:SYSTem:SOFTware?

:SYSTem:SOFTware?

A query delivers the current instrument software version

' <SW-Scope>-<SW-MesOS>'

e.g.: *'05.100-02.005'*

Note: Please update the instrument software with downloads from ' http://www.hameg.com …'

:SYSTem:HARDware ?

:SYSTem:HARDware?

A query delivers the instrument 32 bit hardware identification in hexadecimal format (#Hxxxxxxxx).

e.g.: *' #H10030000 '*

The hardware ID contains information about the instruments hardware components.

:SYSTem:DEVice?

:SYSTem.:DEVice?

A query delivers the instrument type, e.g.: ' HM1508 '

:SYSTem:SET <set>

:SYSTem:SET? delivers the current settings in block format.

<set> *transfers externally stored instrument settings in block format to the oscilloscope.*

:SYSTem:SET:AUTO

:SYSTem:SET:AUTO starts the AUTOSET function of the oscilloscope.

Not available as a query.

:SYSTem:SET:CHANged <change>?

<change> = { TRUE | FALSe }

<change> *- indicates changes by the keyboard since last query command*
only available as a query

' True ' *- a key board operation has made changes since the last query.*
' False ' *- no key board operation made changes since the last query.*

.

Note: *This command can be used to recognise if an inadvertent settings change has been made at the oscilloscope (important if the instrument is integrated in automatic test systems). If the answer is' True ' the instrument parameters can be set anew.*

To prevent accidental keyboard operation please use the key lock function
(:SYST:LOCK ON).

:SYSTem:FPANel

:SYSTem:FPANel:RECall <setNO>

<setNO> = {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}

load of internally stored instrument settings <setNO>.

:SYSTem:FPANel:SAVE <setNO>

<setNO> = {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}

stores the current instrument settings - without reference and mathematics curves - internally <setNO>.

:TRACe?

:TRACe?

A Query returns trace data with appended informations to selected trace.

i.E.: *' #42000DDDD .. D;:TRAC:SOUR CH1;:TRAC:SOUR:CAT CH1;*
:TRAC:POIN 2000;:TRAC:TYPE NORMAL;:TRAC:YINC 40.0000E-3;
:TRAC:YOR 2.41499;:TRAC:YREF 128;:TRAC:YRES 25;:TRAC:YDIS 0;
:TRAC:XINC 10.0000E-3;:TRAC:XOR -10.0000;:TRAC:XREF 0;:TRAC:FORM BYTE;
:TRAC:BORD LSBF; '

(#42000<data 0><data 1> .. <data n>:TRAC:SOUR CH1; ...)

Note: *Queriing an invalid trace will return ' #10 ', data block with length 0.*

i.E.: *' #10;:TRAC:SOUR CH1;:TRAC:SOUR:CAT CH1;*
:TRAC:POIN 2000;:TRAC:TYPE NORMAL;:TRAC:YINC 40.0000E-3;
:TRAC:YOR 2.41499;:TRAC:YREF 128;:TRAC:YRES 25;:TRAC:YDIS 0;
:TRAC:XINC 10.0000E-3;:TRAC:XOR -10.0000;:TRAC:XREF 0;:TRAC:FORM BYTE;
:TRAC:BORD LSBF; '

Information: Trace data can be invalid if the readen channel is deactivated for example.

Use the commands ' :TRACe:SOURce ' to choose trace and ' :CHANnel<n>:STAT ON ' to activate channel n.

:TRACe:BORDer <byte_order>

<byte_order> = {MSBFirst | LSBFirst}

<byte_order> defines the byte order when logic signal records are output.

'MSB' - *most significant byte*
'LSB' - *least significant byte*

:TRACe:DATA?

:TRACe:DATA?

Query return trace data in block format without appended informations to the readen trace.

Note: A Query will return ' #10 ' if the selected trace is invalid.

:TRACe:FORMat <format>

<format> = { BYTE | WORD | ASCii | CSV }

<format> defines the data output format.

'Byte' - signal data output in byte format
'WORD' - signal data output in word format (double byte)
'ASCii' - signal data output in floating point values (with comma separation).
'CSV ' - comma separated value format with prefix header

Note: The decimal limiter is indicated by a decimal point and the data separation by comma. Please note the representation in your national language (e.g for table calculation software) and reformat the data before being processed.

:TRACe:POINts <points>

<points> = {MAXimum | DMAXimun | DEFault}

'MAXimum' - entire output of all signal data stored in the acquisition memory
'DMAXimun' - output of all sampled data shown on display
'DEFault' - only the displayed signal data are output

A query delivers the currently available volume of data which depends on time base setting and (acquisition mode).

Note: The complete memory can only be read out if refresh mode is present and the signal acquisition has been stopped. Otherwise only the currently displayed signal data will be output.

:TRACe:REFerence<r>:SAVE <source>

<r> = { 1 | 2 | .. | 9}

<source> = { CH1 | CH2 | ADD1}

'source' contains the source of the signal curve to be stored in the reference memory <r>.'

'CH1' channel 1 is the signal curve source
'CH2' channel 2 is the signal curve source
'ADD1' the addition result of signal curves channel 1 and 2 will be stored.

Note: signal curves can only be stored if they are displayed

.

:TRACe:SOURce <source>

<source> = {CH<n> | ADD1 | FFT<n> | POD1 | ZCH<n> | ZADD1 | ZPOD1 | MA<m> | RE<r>}

<n> = { 1 | 2 }
<m> = { 1 | 2 | 3 | 4 | 5 }
<r> = { 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 }

<source> *enables the selection of analogue and logic channels, mathematic, zoom and reference curves as source.*

'CH<n>' - *channel <n> as data source*
'ADD1' - *addition curve of CH1 and CH2 as data source*
'FFT<n>' - *contains the FFT signal curve of channel <n>.*
'POD1' - *contains the logic channels as data source*
CH3, CH4 - HM1508, HM1508-2
LC0, .. , LC3 - HM2008

'ZCH<n>' - *zoomed signal curve of channel <n>*
'ZADD1' - *CH1 and CH2 addition zoomed signal curve*
'ZPOD1' - *logic channels zoomed signal curve*

'MA<m>' - *mathematic signal curve <m>*
'RE<r>' - *reference signal curve <r>*

:TRACe:SOURce:CATalog

:TRACe:SOURce:CATalog?

a list of currently displayed signal curves will be output (with comma separation)

:TRACe:TYPE?

:TRACe:TYPE? = {NORMAL | MINMAX}

'normal' - *each acquisition generates a data set representing the signal.*
'minmax' - *each dot contains two values (one minimal and one maximal value). In standard output format (BYTE) there are twice as much data being displayed with ' :TRACe:POINts <points> '.*

The degree (height) of minima and maxima is signal dependent and therefore undefined.

:TRACe:XINCrement?

:TRACe:XINCrement?

.. returns the difference between two in X direction adjacent signal points.

:TRACe:XORigin?

:TRACe:XORigin?

In Yt (time base) mode a query causes the reply of the time difference between trigger event and the XREF point ; in FFT mode the frequency difference between the centre frequency and the most left dot.

:TRACe:XREFerence?

:TRACe:XREFerence? = {0}

:TRACe:XREFerence? describes the most left displayed dot (Index 0).

:TRACe:XUNit ?

:TRACe:XUNit ?

return string with unit of x-axis, i.e. ' "s" ' (seconds)

:TRACe:YINCrement?

:TRACe:YINCrement?

.. returns the Y resolution in the unit Y (unit)/Bit.

:TRACe:YORigin?

:TRACe:YORigin?

A query delivers the signal value of the reference voltage YREF

YORigin is the equivalent of the voltage value of YReference (binary form).

:TRACe:YREFerence?

:TRACe:YREFerence?

.. describes the binary reference value of the display centre.

The voltage of a sample point can be determined as follows:

$$U_{SP} = (Data - YREFerence) * YINCrement + YORigin$$

Data – binary value of a signal curve sample (point)

:TRACe:YRESolution?

:TRACe:YRESolution?

.. describes the graticule vertical resolution Bit/DIV

:TRACe:YDISplacement?

:TRACe:YDISplacement?

A query delivers an offset value in DIV. If the captured graph has an additional offset this will be displayed. Other characteristics remain unchanged.

:TRACe:YUNit ?

:TRACe:YUNit ?

return string with unit of y-axis, i.e. ' "V" ' (Volts)

:TRIGger?

:TRIGger?

A query cause the reply of all trigger menu parameters

e.g.: *' :TRIG:A:MODE AUTO;:TRIG:A:TYPE EDGE;:TRIG:A:EDGE:SOUR CH1;*
:TRIG:A:EDGE:SLOP POS;:TRIG:A:EDGE:COUP DC;:TRIG:A:EDGE:LEV 0;
:TRIG:A:EDGE:FILT:LPAS OFF;:TRIG:A:EDGE:FILT:NREJ OFF;:TRIG:A:VID:FIEL ALLL;
:TRIG:A:VID:LINE 8.00;:TRIG:A:VID:POL NEG;:TRIG:A:VID:STAN PAL;
:TRIG:A:VID:SOUR CH1;:TRIG:A:LOG:FUNC AND;
:TRIG:A:LOG:WHEN TRUE;:TRIG:A:LOG:SOUR "HXXX"; '

:TRIGger:A:EDGE?

:TRIGger:A:EDGE ?

contains all standard trigger functions (trigger level and trigger slope)

:TRIGger:A:EDGE:FILTer:LPASs <state>

<state> = {ON | OFF}

<state> *switches the trigger low pass filter ON or OFF.*

:TRIGger:A:EDGE:FILTer:NREJect <state>

<state> = {ON | OFF}

<state> *activates/deactivates the noise rejection*

:TRIGger:A:EDGE:SOURce <source>

<source> = {CH1 | CH2 | CH3 | CH4 | LC0 | LC1 | LC2 | LC3 | EXT | LINE | ALTernating}

<source> *contains the trigger source*

'CH1', 'CH2'	*- channel 1 and 2*
'CH3', 'CH4'	*- logic channels 1 and 2, HM1508*
'LC0', 'LC1', 'LC2', 'LC3'	*- logic channels 1 to 4, HM2008*
'EXT'	*- external trigger*
'LINE'	*- mains trigger*
'ALTernating'	*- channel 1 and 2 alternate as trigger source*

:TRIGger:A:EDGE:SLOPe <slope>

<slope> = {POSitive | NEGative | EITHer}

<slope> *determines the slope used for triggering.*

'positive'	*- positive slope*
'negative'	*- negative slope*
'either'	*- both slope directions*

:TRIGger:A:EDGE:COUPling <coupling>

<coupling> = {AC | DC | HF}

<coupling> *- trigger signal coupling*

'AC'	*- dc content suppressed*
'DC'	*- dc content not suppressed*
'HF'	*- dc and low frequency content suppressed*

:TRIGger:A:EDGE:LEVel <level>

<level> ={MINimum | MAXimum | (floating point value in V)}

<level> *determines the trigger level (signal height required to trigger the time base).*
This level information is only regarding CH1 and CH2.

:TRIGger:A:LOGic?

:TRIGger:A:LOGic?

A query delivers all logic triggering regarding information.

e.g.: *' :TRIG:A:LOG:FUNC AND;:TRIG:A:LOG:WHEN TRUE;:TRIG:A:LOG:SOUR "HXXX"; '*

:TRIGger:A:LOGic:FUNCtion <function>

<function> = {AND | OR}

:TRIGger:A:LOGic:WHEN <condition>

<condition> = {TRUE | FALSE}

:TRIGger:A:LOGic:SOURce <string>

<string> = { „<sCH1><sCH2><sLOG1><sLOG2>“ } HM1508x

<sCH1>,<sCH2>,<sLOG1>,<sLOG2> = { X | H | L }

<string> = { "<sLC0><sLC1><sLC2><sLC3>" } HM2008

<sLC0>, <sLC1>, <sLC2>, <sLC3> = { X | H | L }

<s .. > *sets the logic status to be considered as TRUE*

'X' - *both states are TRUE*
'H' - *high state is TRUE*
'L' - *low state is TRUE*

Example: *:TRIgger:A:LOGic:SOURce "HXXL"*

:TRIGger:A:MODE <mode>

<mode> = {AUTO | NORMal | SINGle}

<mode> *determines the trigger mode.*

'auto' - *the time base is triggered continuously independent of a trigger signal.*

'normal' - *the time base can only be triggered by a trigger signal.*

'single' - *one time triggering*

:TRIGger:A:TYPE <type>

<type> = {EDGE | VIDeo | LOGic}

<type> *enables the selection of signal type specific triggering.*

'Edge' - *standard triggering type by slope and trigger level*
'Video' - *special triggering on video signals*
'Logic' - *special triggering on logic signals**

**only HM1508x, HM2008*

:TRIGger:A:VIDeo?

:TRIGger:A:VIDeo?

A query causes the reply of all video trigger regarding parameters

e.g.: *' :TRIG:A:VID:FIEL ALLL;:TRIG:A:VID:LINE 8.00;:TRIG:A:VID:POL NEG;*
:TRIG:A:VID:STAN PAL;:TRIG:A:VID:SOUR CH1; '

:TRIGger:A:VIDeo:FIELd <field>

<field> = {ODD | EVEN | ALL | LINE | ALLLine}

<field> *selects the part of the video signal to be used for triggering.*

'ODD' - *triggering on odd frames*
'EVEN' - *triggering on even frames*
'ALL' - *triggering on both frames*
'LINE' - *prepares to trigger on a selected line**
'ALLLine' - *triggering on all lines*

** The line number must be input with the command: TRIGger:A:VIDeo:LINE <line>.*

:TRIGger:A:VIDeo:LINE <line>

<line> = {MINimum | MAXimum | line number}

<line> *is the number of the line to be used for triggering*

'MIN' - *selects the first line*
'MAX' - *selects the last line*
'line number' - *selects the line to be used for triggering*

Note: *This function can be used only if the parameter field in :TRIGger:A:VIDeo:FIELd <field> has been set to 'LINE'.*

e.g.: *' :TRIGger:A:VIDeo:FIELd LINE ; LINE 111'*

:TRIGger:A:VIDeo:STANdard <standard>

<standard> = {PAL | NTSC}

<standart> *selects the analogue video signal standard*

'PAL' - *this abbreviation is used as a synonym for 625 lines, 50 frames/s*

'NTSC' -..*this abbreviation is used as a synonym for 525 lines, 60 frames/s*

:TRIGger:A:VIDeo:POLarity <polarity>

<polarity> = {POSitive | NEGative}

<polarity> describes the synchronisation pulse`s polarity

'positive' - *positive synchronisation pulse*
'negative' - *negative synchronisation pulse*

:TRIGger:A:VIDeo:SOURce <source>

<source> = {CH1 | CH2 | EXT}

<source> *defines the source from which the video signal originates*

'CH1' - *channel 1*
'CH2' - *channel 2*
'EXT' - *video signal triggering via the external trigger input*

4. Measure – Error Messages

Error number	Description
2000	The read out value is valid.
2001	At the time of measurement no valid data are available.
2002	The current oscilloscope settings do not allow the desired measuring function.
2003	The measurement has been made with not calibrated oscilloscope settings.
2004	The measurement has been made with not calibrated oscilloscope settings. The real measuring result is higher then the returned result.
2005	The measurement has been made with not calibrated oscilloscope settings. The real measuring result is lower then the returned result.
2006	The measuring result is based on the addition of two signals, although the Y deflection coefficient settings were not equal for both channels.
2007	The input signal has exceeded the negative Analogue/Digital Converter dynamic range.
2008	The input signal has exceeded the positive Analogue /Digital Converter dynamic range.
2009	The input signal has exceeded the positive and negative Analogue /Digital Converter dynamic range.

Table 20: Measure – Error Messages