SCPI Programmers Manual HMC8012

English

Subject to change without notice

Inhalt

1. Introduction / Basics

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

1.1 Remote Control Interfaces

For remote control, LAN / USB (standard interface) or GPIB (optional interface) can be used. The optional GPIB interface has its own interface module slot on the rear panel of the HMC8012.

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

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

1.1.1 USB Interface

In addition to a LAN interface, the HMC8012 includes a USB device port. For this interface, the user can select if the instrument is accessed via virtual COM port (VCP) or via USB TMC class. The traditional version of the VCP allows the user to communicate with the HMC using any terminal program via SCPI commands once the corresponding Windows drivers have been installed. For the multimeter HMC8012, these commands are mostly compatible with the Agilent multimeters 34401A and 34410A. Naturally, the free HAMEG software "HMExplorer" is also available for the HMC series. This Windows application offers HMC instruments a terminal function, the option to create screenshots and to read out the measured data from the HMC memory.

The modern alternative to the virtual COM port is to remote control the HMC8012 via USB TMC class. TMC stands for "Test & Measurement Class" which indicates that the connected measurement instrument can be recognized without special Windows drivers if VISA drivers are installed and that it can be used directly in corresponding environments. The GPIB interface serves as model to the structure of the TMC design. A major benefit of the USB TMC class is that by sampling specific registers the controlling software can determine if commands have been terminated and if they have been processed correctly. In contrast, the communication via VCP requires analysis and polling mechanisms within the controlling software which may significantly strain the interface of the measurement instruments. The TMC status registers solve this problem with the USB TMC in the same manner as is the case with the GPIB interface for the hardware, namely via corresponding control lines.

If you are using USB you need to install an USB driver, which can be downloaded free of charge from the HAMEG homepage.

☞ The available USB driver is fully tested, functional and released for Windows XP™ 32 Bit, Windows Vista™ or Windows 7™ both as 32Bit or 64Bit versions.

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

☞ If the virtual COM port will be used, you have to install the virtual COM port part of the HMC8012 USB driver. The virtual COM port (VCP) will be activated in the PC device explorer.

1.1.2 LAN Interface

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

IP address

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

TCPIP::‹IP_address›::‹IP_port›::SOCKET

The default port number for SCPI socket communication is 5025. IP address and port number are listed In the „Ethernet Settings" of the HMC8012, see also: chapter 1.2.2,"Configuring LAN Parameters", on page 4.

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

TCPIP::192.1.2.3::5025::SOCKET

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

TCPIP::‹host_name›::‹IP_port›::SOCKET

To assign a host name to the HMC8012, select `SETUP` button › `Misc` › `Device name`.
Example: If the host name is HAMEG1; the valid resource string is:

TCPIP::HAMEG1::5025::SOCKET

☞ **The end character must be set to linefeed (LF).**

1.1.3 GPIB Interface (IEC/IEEE Bus Interface)

In addition to the GPIB functions which are available via USB TMC class, the HMC8012 is optionally available with an integrated GPIB interface. This solution is particularly attractive for customers who already have an existing GPIB environment. With minimum efforts, an old instrument can be replaced by a model of the HMC8012.

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

Characteristics

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

GPIB Instrument Address

In order to operate the instrument via remote control, it must be addressed using the GPIB address. The remote control address is factory-set to 20, but it can be changed in the network environment settings or in the „Setup" menu under „Interface › Parameter". For remote control, a GPIB address from 0 to 30 are allowed. The GPIB address is maintained after a reset of the instrument settings.

1.2 Setting Up a Network (LAN) Connection

1.2.1 Connecting the Instrument to the Network

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

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

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

1.2.2 Configuring LAN Parameters

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

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

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

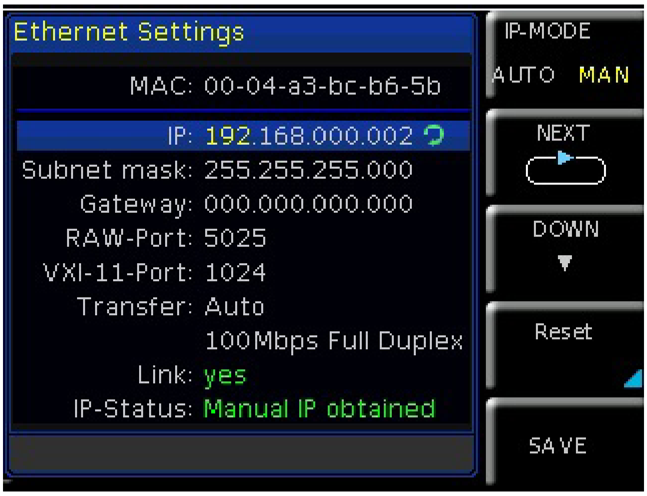
Configuring LAN parameters

1. Press the SETUP key and then the Interface softkey.

2. Press the Ethernet and then the Parameter softkey.

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

   The „Ethernet Settings" dialog box is displayed.



Ethernet Settings

MAC: 00-04-a3-bc-b6-5b
IP: 192.168.000.002
Subnet mask: 255.255.255.000
Gateway: 000.000.000.000
RAW-Port: 5025
VXI-11-Port: 1024
Transfer: Auto
100Mbps Full Duplex
Link: yes
IP-Status: Manual IP obtained

Fig. 1.1: Ethernet Settings dialog box

Some data is displayed for information only and cannot be edited. This includes the „MAC" (physical) address of the connector and the „Link" status information.

4. Define the IP address of the instrument by entering each of the four blocks individually (manual mode) or choose the automatic IP-Mode.

   a) In manual mode (MAN) define the first block number using the knob.
   b) Press Next to move to the next block and define the number.
   c) When the IP address is complete, press Down to continue with the next setting.

5. Define the „Subnetmask" and „Gateway" in the same way.

6. Select the „RAW Port" - the port number for SCPI socket communication.

7. Select the „VXI-11- Port" used by the instrument.

8. Select the „Transfer" mode. This mode can either be determined automatically („Auto" setting), or you can select a combination of a transfer rate and half or full duplex manually.

9. Press Save to save the LAN parameters.

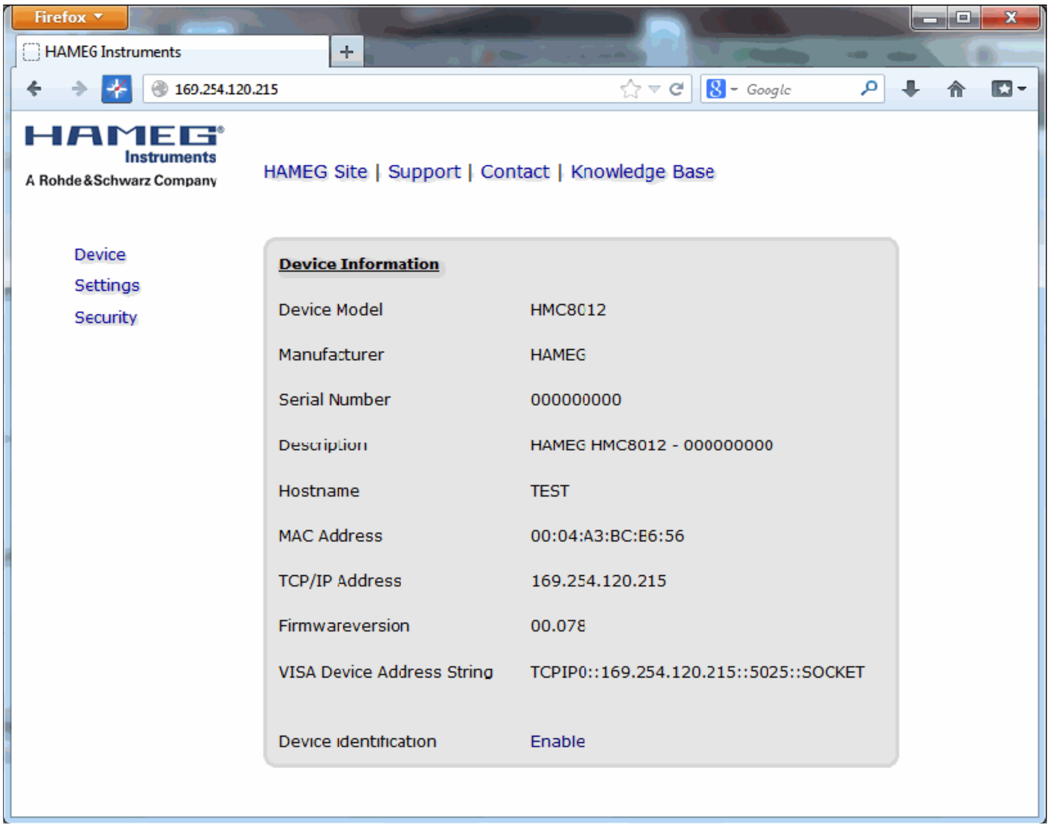
☞ **The „Link" and „IP-Status" information at the bottom of the dialog box indicates whether a LAN connection was established successfully.**

Checking LAN and SCPI connection

1. Check the LAN connection using ping: `ping xxx.yyy.zzz.xxx`.

2. If the PC can access the instrument, enter the IP address of the address line of the internet browser on your computer: http//:xxx.yyy.zzz.xxx

   The „Instrument Home" page appears. It provides information on the instrument and the LAN connection.



Device Information

Device Model	HMC8012
Manufacturer	HAMEG
Serial Number	000000000
Description	HAMEG HMC8012 - 000000000
Hostname	TEST
MAC Address	00:04:A3:BC:E6:56
TCP/IP Address	169.254.120.215
Firmwareversion	00.078
VISA Device Address String	TCPIP0::169.254.120.215::5025::SOCKET
Device identification	Enable

Fig. 1.2: Web server

1.3 Switching to Remote Control

When you switch on the instrument, it is always in manual operation state („local" state) and can be operated via the front panel.

When you send a command from the control computer, it will be received and executed by the instrument. The display remains on, manual operation via the front panel is always possible.

1.4 Messages and Command Structure

1.4.1 Messages

Instrument messages are employed in the same way for all interfaces, if not indicated otherwise in the description.

See also:
– Structure and syntax of the instrument messages: chapter 1.4.2, „SCPI Command Structure", on page 7
– Detailed description of all messages: chapter 2, „Command Reference", on page 20

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

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

1. **According to the effect they have on the instrument:**

   - **Setting commands** cause instrument settings such as a reset of the instrument or setting the frequency.

   - **Queries** cause data to be provided for remote control, e.g. for identification of the instrument or polling a parameter value. Queries are formed by directly appending a question mark to the command header.

2. **According to their definition in standards:**

   - **The function and syntax of the Common commands** are precisely defined in standard IEEE 488.2. They are employed identically on all instruments (if implemented). They refer to functions such as management of the standardized status registers, reset and self test.

   - **Instrument control commands** refer to functions depending on the features of the instrument such as frequency settings. Many of these commands have also been standardized by the SCPI committee. These commands are marked as „SCPI compliant" in the command reference chapters. Commands without this SCPI label are device-specific, however, their syntax follows SCPI rules as permitted by the standard.

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

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

- **Universal commands:** act on all instruments connected to the GPIB bus without previous addressing

- **Addressed commands:** only act on instruments previously addressed as listeners

Universal Commands
Universal commands are encoded in the range 10 through 1F hex. They affect all instruments connected to the bus and do not require addressing.

Addressed commands are encoded in the range 00 through 0F hex. They only affect instruments addressed as listeners.

1.4.2 SCPI Command Structure

SCPI commands consist of a so-called header and, in most cases, one or more parameters. The header and the parameters are separated by a „white space“ (ASCII code 0 to 9, 11 to 32 decimal, e.g. blank). The headers may consist of several mnemonics (keywords). Queries are formed by appending a question mark directly to the header.

The commands can be either device-specific or device-independent (common commands). Common and device-specific commands differ in their syntax.

Syntax for Common Commands

Common (= device-independent) commands consist of a header preceded by an asterisk (*) and possibly one or more parameters.

Examples:

*RST	RESET	Resets the instrument.
*ESE	EVENT STATUS ENABLE	Sets the bits of the event status enable registers.
*ESR?	EVENT STATUS QUERY	Queries the contents of the event status register.
*IDN?	IDENTIFICATION QUERY	Queries the instrument identification string.

Table 1.4: Examples of Common Commands

Syntax for Device-Specific Commands

Not all commands used in the following examples are necessarily implemented in the instrument.

For demonstration purposes only, assume the existence of the following commands for this section:

```
- CALCulate:FUNCtion {NULL | DB | DBM | AVERage | LIMit}
- CALCulate:FUNCtion?
- CALCulate[:STATe] {OFF | ON}
- CALCulate[:STATe]?
```

Long and short form

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

Example:

`CALCulate:FUNCtion NULL` is equivalent to `CALC:FUNC NULL`

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

Optional mnemonics

Some command systems permit certain mnemonics to be inserted into the header or omitted. These mnemonics are marked by square brackets in the description. The instrument must recognize the long command to comply with the SCPI standard. Some commands are considerably shortened by these optional mnemonics.

Example:

```
INITiate[:IMMediate]
```

Definition: INITiate[:IMMediate]
Command: `INIT:IMM` is equivalent to `INIT`

Special characters

|	Parameters A vertical stroke in parameter definitions indicates alternative possibilities in the sense of „or“. The effect of the command differs, depending on which parameter is used. Example: Definition: `UNIT:TEMPerature {C | K | F}` Command: `UNIT:TEMP C` selects °C for temperature measurements Command: `UNIT:TEMP K` selects Kelvins for temperature measurements
[ ]	Mnemonics in square brackets are optional and may be inserted into the header or omitted. Example: `INITiate[:IMMediate]` `INIT:IMM` is equivalent to `INIT`

{ }	Parameters in curly brackets are optional and can be inserted once or several times, or omitted.
	Example: `CALCulate:LIMit:LOWer {<Value> | MINimum | MAXimum}`
	The following are valid commands:
	`CALC:LIM:LOW 10`
	`CALC:LIM:LOW MIN`
	`CALC:LIM:LOW MAX`

Table 1.5: Special characters

SCPI Parameters

Many commands are supplemented by a parameter or a list of parameters. The parameters must be separated from the header by a „white space“ (ASCII code 0 to 9, 11 to 32 decimal, e.g. blank). Allowed parameters are:

- Numeric values
- Special numeric values
- Boolean parameters
- Text
- Character strings
- Block data

The parameters required for each command and the allowed range of values are specified in the command description.

Numeric values

Numeric values can be entered in any form, i.e. with sign, decimal point and exponent. Values exceeding the resolution of the instrument are rounded up or down. The mantissa may comprise up to 255 characters, the exponent must lie inside the value range -32000 to 32000. The exponent is introduced by an „E“ or „e“. Entry of the exponent alone is not allowed. In the case of physical quantities, the unit can be entered. Allowed unit prefixes are MA (mega) / MOHM, K (kilo), M (milli) and U (micro). If the unit is missing, the basic unit is used.

Example: `CALC:NULL:OFF 10mV = CALC:NULL:OFF 10E-3`

Special numeric values

The texts listed below are interpreted as special numeric values. In the case of a query, the numeric value is provided.

- MIN/MAX
  MINimum and MAXimum denote the minimum and maximum value.

Example:

Setting command: `CALCulate:LIMit:LOWer MAXimum`
Query: `CALC:LIM:LOW MAX?, Response: 1E3`

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

Example: `CALC:LIM:LOW? MAXimum`

Returns the maximum numeric value as a result.

Boolean parameters

Boolean parameters represent two states. The „ON“ state (logically true) is represented by „ON“ or a numeric value 1. The „OFF“ state (logically untrue) is represented by „OFF“ or the numeric value 0. The numeric values are provided as the response for a query.

Example:

Setting command: `CALCulate[:STATe] ON`
Query: `CALC:STAT?`
Response: `1`

Text parameters

Text parameters observe the syntactic rules for mnemonics, i.e. they can be entered using a short or long form. Like any parameter, they have to be separated from the header by a white space. In the case of a query, the short form of the text is provided.

Example:

Setting command: `HCOPy:FORMat BMP`
Query: `HCOPy:FORMat?`
Response: `BMP`

Overview of Syntax Elements

The following table provides an overview of the syntax elements:

:	The colon separates the mnemonics of a command. In a command line the separating semicolon-marks the uppermost command level.
;	The semicolon separates two commands of a command line. It does not alter the path.
,	The comma separates several parameters of a command.
?	The question mark forms a query.
*	The asterisk marks a common command.
“	Quotation marks introduce a string and terminate it.
	A „white space“ (ASCII-Code 0 to 9, 11 to 32 decimal, e.g. blank) separates the header from the parameters.

Table 1.6: Syntax Elements

Structure of a command line

A command line may consist of one or several commands. It is terminated by one of the following:
- a ‹New Line›
- a ‹New Line› with EOI
- an EOI together with the last data byte

Several commands in a command line must be separated by a semicolon „;“. If the next command belongs to a different command system, the semicolon is followed by a colon.

If the successive commands belong to the same system, having one or several levels in common, the command line can be abbreviated. To this end, the second command after the semicolon starts with the level that lies below the common levels. The colon following the semicolon must be omitted in this case.

Example:
```
CALC:FUNC NULL; CALC ON
```

This command line is represented in its full length and contains two commands separated from each other by the semicolon. Both commands are part of the CALC command system, i.e. they have one level in common. When abbreviating the command line, the second command begins with the level below CALC The colon after the semicolon is omitted. The abbreviated form of the command line reads as follows:

```
CALC:FUNC NULL;ON
```

A new command line always begins with the complete path.

Example:
```
CALC:FUNC NULL
CALC ON
```

Responses to Queries

A query is defined for each setting command unless explicitly specified otherwise. It is formed by adding a question mark to the associated setting command. According to SCPI, the responses to queries are partly subject to stricter rules than in standard IEEE 488.2.

- The requested parameter is transmitted without a header.

  Example: `HCOPy:FORMat?,`
  Response: `BMP`

- Maximum values, minimum values and all other quantities that are requested via a special text parameter are returned as numeric values.

  Example: `CALCulate:LIMit:LOWer? MAXimum,`
  Response: `1E3`

- Numeric values are output without a unit. Physical quantities are referred to the basic units or to the units set using the Unit command. The response 3.5E9 in the previous example stands for 3.5 GHz.

- Truth values (Boolean values) are returned as 0 (for OFF) and 1 (for ON).

  Example:
  Setting command: `CALCulate:STATe ON`
  Query: `CALCulate:STATe?`
  Response: `1`

1.5 Command Sequence and Synchronization

IEEE 488.2 defines a distinction between overlapped and sequential commands:

- A sequential command finishes executing before the next command starts executing. Commands that are processed quickly are usually implemented as sequential commands.

- An overlapping command does not automatically finish executing before the next command starts executing. Usually, overlapping commands take longer to process and allow the program to do other tasks while being executed. If overlapping commands do have to be executed in a defined order, e.g. in order to avoid wrong measurement results, they must be serviced sequentially. This is called synchronization between the controller and the instrument.

Setting commands within one command line, even though they may be implemented as sequential commands, are not necessarily serviced in the order in which they have been received. In order to make sure that commands are actually carried out in a certain order, each command must be sent in a separate command line.

Example: Commands and queries in one message

The response to a query combined in a program message with commands that affect the queried value is not predictable. The following commands always return the specified result:

```
:CALC:FUNC DB;CALC:STAT ON;CALC:DB:REF MAX :CALC:DB:REF?
```

Result: 7.500E+02

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

1.5.1 Preventing Overlapping Execution

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

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

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

Command synchronization using `*WAI` or `*OPC?` appended to an overlapped command is a good choice if the overlapped command takes time to process. The two synchronization techniques simply block overlapped execution of the command.

For time consuming overlapped commands it is usually desirable to allow the controller or the instrument to do other useful work while waiting for command execution. Use one of the following methods:

***OPC with a service request**

1. Set the OPC mask bit (bit no. 0) in the ESE: *ESE 1

2. Set bit no. 5 in the SRE: *SRE 32 to enable ESB service request.

3. Send the overlapped command with *OPC

4. Wait for a service request
   The service request indicates that the overlapped command has finished.

***OPC? with a service request**

1. Set bit no. 4 in the SRE: *SRE 16 to enable MAV service request.
2. Send the overlapped command with *OPC?
3. Wait for a service request
   The service request indicates that the overlapped command has finished.

Event Status Register (ESE)

1. Set the OPC mask bit (bit no. 0) in the ESE: *ESE 1
2. Send the overlapped command without *OPC, *OPC? or *WAI
3. Poll the operation complete state periodically (by means of a timer) using the sequence: *OPC; *ESR?
   A return value (LSB) of 1 indicates that the overlapped command has finished.

***OPC? with short timeout**

1. Send the overlapped command without *OPC, *OPC? or *WAI
2. Poll the operation complete state periodically (by means of a timer) using the sequence: ‹short timeout›; *OPC?
3. A return value (LSB) of 1 indicates that the overlapped command has finished. In case of a timeout, the operation is ongoing.
4. Reset timeout to former value
5. Clear the error queue with `SYStem:ERRor?` to remove the „-410, Query interrupted" entries.

Using several threads in the controller application

As an alternative, provided the programming environment of the controller application supports threads, separate threads can be used for the application GUI and for controlling the instrument(s) via SCPI.

A thread waiting for a *OPC? thus will not block the GUI or the communication with other instruments.

1.6 Status Reporting System

The status reporting system stores all information on the current operating state of the instrument, and on errors which have occurred. This information is stored in the status registers and in the error queue. Both can be queried via GPIB bus or LAN interface (`STATus`... commands).

1.6.1 Structure of a SCPI Status Register

Each standard SCPI register consists of 5 parts. Each part has a width of 16 bits and has different functions. The individual bits are independent of each other, i.e. each hardware status is assigned a bit number which is valid for all five parts. Bit 15 (the most significant bit) is set to zero for all parts. Thus the contents of the register parts can be processed by the controller as positive integers.

Description of the five status register parts (please refer to page 20)

The five parts of a SCPI register have different properties and functions:

- **CONDition**
  The `CONDition` part is written into directly by the hardware or the sum bit of the next lower register. Its contents reflect the current instrument status. This register part can only be read, but not written into or cleared. Its contents are not affected by reading.

- **EVENt**
  The `EVENt` part indicates whether an event has occurred since the last reading, it is the „memory" of the condition part. It only indicates events passed on by the transition filters. It is permanently updated by the instrument. This part can only be read by the user. Reading the register clears it. This part is often equated with the entire register.

- **ENABle**
  The `ENABle` part determines whether the associated `EVENt` bit contributes to the sum bit (see below). Each bit of the EVENt part is „ANDed" with the associated ENABle bit (symbol ‚&'). The results of all logical operations of this part are passed on to the sum bit via an „OR" function (symbol ‚+').
  `ENABle` bit = 0: the associated `EVENt` bit does not contribute to the sum bit `ENABle` bit = 1: if the associated `EVENt` bit is „1", the sum bit is set to „1" as well. This part can be written into and read by the user as required. Its contents are not affected by reading.

Sum bit

The sum bit is obtained from the `EVENt` and `ENABle` part for each register. The result is then entered into a bit of the `CONDition` part of the higher-order register.

The instrument automatically generates the sum bit for each register. Thus an event can lead to a service request throughout all levels of the hierarchy.

1.6.2 Hierarchy of status registers

The status information has an hierarchical structure.

- **STB, SRE**
  The STatus Byte (STB) register and its associated mask register Service Request Enable (SRE) form the highest level of the status reporting system. The STB provides a rough overview of the instrument status, collecting the information of the lower-level registers.

- **ESR, SCPI registers**
  The STB receives its information from the following registers:
  – The Event Status Register (ESR) with the associated mask register standard Event Status Enable (ESE).
  – The STATus:OPERation and STATus:QUEStionable registers which are defined by SCPI and contain detailed instrument information.

- **Output buffer**
  The output buffer contains the messages the instrument returns to the controller. It is not part of the status reporting system, but determines the value of the MAV bit in the STB and thus is represented in the overview.

All status registers have the same internal structure.

☞ SRE, ESE
The service request enable register SRE can be used as ENABle part of the STB if the STB is structured according to SCPI. By analogy, the ESE can be used as the ENABle part of the ESR.

1.6.3 Contents of the Status Registers

In the following sections, the contents of the status registers are described more detailed (please refer to page 20).
Status Byte (STB) and Service Request Enable Register (SRE)

The STatus Byte (STB) is already defined in IEEE 488.2. It provides a rough overview of the instrument status by collecting the pieces of information of the lower registers. A special feature is that bit 6 acts as the sum bit of the remaining bits of the status byte.

The STB can thus be compared with the CONDition part of an SCPI register and assumes the highest level within the SCPI hierarchy.

The STB is using the command *STB or a serial poll.

The STatus Byte (STB) is linked to the Service Request Enable (SRE) register. Each bit of the STB is assigned a bit in the SRE. Bit 6 of the SRE is ignored. If a bit is set in the SRE and the associated bit in the STB changes from 0 to 1, a service request (SRQ) is generated. The SRE can be set by using the command *SRE and can be read by using the command *SRE?.

Bit No.	Meaning
0...1	Not used
2	Error Queue The bit is set when an error is occured. If this bit is enabled by the SRE, each entry of the error queue generates a service request. Thus an error can be recognized and specified in greater detail by polling the error queue. The poll provides an informative error message. This procedure is to be recommended since it considerably reduces the problems involved with remote control.
3	QUEStionable status sum bit The bit is set, if an `EVENt` bit is set in the `QUEStionable` status register and the associated `ENABle` bit is set to 1. A set bit indicates a questionable instrument status, which can be specified in detail by polling the `QUEStionable` status register.
4	MAV bit (message available) The bit is set, if a readable message in the output buffer is available. This bit can be used to enable data to be automatical read from the instrument.
5	ESB bit Sum bit of the event status register. It is set, if one of the bits in the event status register is set and enabled in the event status enable register. Setting of this bit indicates a serious error, which can be specified in greater detail by polling the event status register.
6	MSS bit (master status summary bit) The bit is set, if the instrument triggers a service request. This is the case, if one of the other bits of this register is set together with its mask bit in the service request enable register SRE.
7	`OPERation` status register sum bit The bit is set, if an `EVENt` bit is set in the `OPERation` status register and the associated `ENABle` bit is set to 1. A set bit indicates that the instrument is just performing an action. The type of action can be determined by polling the `OPERation` status register.

Table 1.8: Bits of the status byte **(please refer to page 20)**

Event Status Register (ESR) and Event Status Enable Register (ESE)

The ESR is defined in IEEE 488.2. It can be compared with the `EVENt` part of a SCPI register. The event status register can be read out using command `*ESR?`.

The ESE corresponds to the `ENABle` part of a SCPI register. If a bit is set in the ESE and the associated bit in the ESR changes from 0 to 1, the ESB bit in the STB is set. The ESE register can be set using the command `*ESE` and read using the command `*ESE?`.

Bit No.	Meaning
0	Operation Complete This bit is set on receipt of the command *OPC exactly, when all previous commands have been executed.
1	Not used
2	Query Error This bit is set, if either the controller wants to read data from the instrument without having sent a query, or if it does not fetch requested data and sends new instructions to the instrument instead. The cause is often a query which is faulty and hence cannot be executed.
3	Device-dependent Error This bit is set, if a device-dependent error occurs. An error message with a number between -300 and -399 or a positive error number is entered into the error queue.
4	Execution Error This bit is set if a received command is syntactically correct, but cannot be performed for other reasons. An error message with a number between -200 and -300 is entered into the error queue.
5	Command Error This bit is set, if a command is received, which is undefined or syntactically incorrect. An error message with a number between -100 and -200 is entered into the error queue.
6	Not used
7	Power On (supply voltage on) This bit is set, when switching on the instrument.

Table 1.9: Bits of the event status register **(please refer to page 20)**

STATus:OPERation Register

In the `CONDition` part, the register contains information which operations the instrument is being executing. In the EVENt part, it contains information which operations the instrument has executed since the last reading. It can be read using the commands `STATus:OPERation:CONDition?` or `STATus:OPERation[:EVENt]?.`

The remote commands for the STATus:OPERation register are described on page 65.

Bit No.	Meaning
0	`Calibrating` (for service department only)
1 to 3	`Not used`

4	`Measuring`
	`The bit is set, while the instrument is measuring.`

5	`Waiting for Trig` This bit is set while the instrument is waiting for the trigger.
6 to 9	Not used
10	Instrument Locked (RWLock)
11 to 15	Not used

Table 1.10: Bits of the STATus:OPERation register **(please refer to page 20)**

STATus:QUEStionable Register

This register contains information about indefinite states which may occur, if the unit is operated without meeting the specifications. It can be read using the commands `STATus:QUEStionable:CONDition` on page 66 and `STATus:QUEStionable[:EVENt]` on page 67.

Bit No.	Meaning
0	Voltage overrange This bit is set, if a voltage range overload occurs.
1	Current overrange This bit is set if a current range overload occurs.
2 to 3	`Not used`
4	`Temperature overrange` This bit is set if a temperature range overload occurs.
5	Frequency overload / underflow This bit is set if a frequency range overload / underflow occurs.
6 to 8	`Not used`
9	Resistance overrange This bit is set if a resistance range overload occurs.
10	Capacitance overload / underflow This bit is set if a capacitance range overload / underflow occurs.
11	`Lower limit failed` This bit is set if a lower limit value is violated.
12	Upper limit failed This bit is set if an upper limit value is violated.
13 to 15	Not used

Table 1.11: Bits of the STATus:QUEStionable register **(please refer to page 20)**

1.6.4 Application of the Status Reporting System

The purpose of the status reporting system is to monitor the status of one or several devices in a measuring system. The controller must receive and evaluate the information of all devices. The following standard methods are used:

- **Service request** (SRQ) initiated by the instrument
- **Serial poll** of all devices in the bus system, initiated by the controller in order to find out who sent a SRQ and why
- **Parallel poll** of all devices
- Query of a **specific instrument status** by means of commands
- Query of the **error queue**

Serial Poll

In a serial poll, with command `*STB` the status byte of an instrument is queried. The query is realized via interface messages and thus clearly faster. The serial poll method is defined in IEEE 488.1 and used to be the standard possibility for different instruments to poll the status byte. The method also works for instruments, which do not adhere to SCPI or IEEE 488.2.

The serial poll is mainly used to obtain a fast overview of the state of several instruments connected to the controller.

Query of an instrument status

Each part of any status register can be read using queries. There are two types of commands:

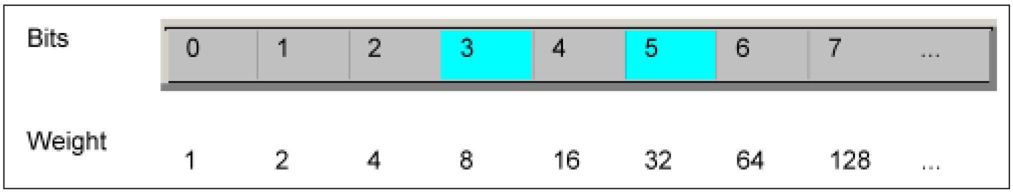
- The common commands `*ESR?, *IDN?, *STB?` query the higher-level registers.
- The commands of the STATus system query the SCPI registers (`STATus:QUEStionable...`)

The returned value is always a decimal number that represents the bit pattern of the queried register. This number is evaluated by the controller program.

Queries are usually used after an SRQ in order to obtain more detailed information on the SRQ cause.

Decimal representation of a bit pattern (binary weights)

The STB and ESR registers contain 8 bits, the SCPI registers 16 bits. The contents of a status register are specified and transferred as a single decimal number. To make this possible, each bit is assigned a weighted value. The decimal number is calculated as the sum of the weighted values of all bits in the register that are set to 1.



Bits	0	1	2	3	4	5	6	7	...
Weight	1	2	4	8	16	32	64	128	...

Fig. 1.7: Decimal representation of a bit pattern **(please refer to page 20)**

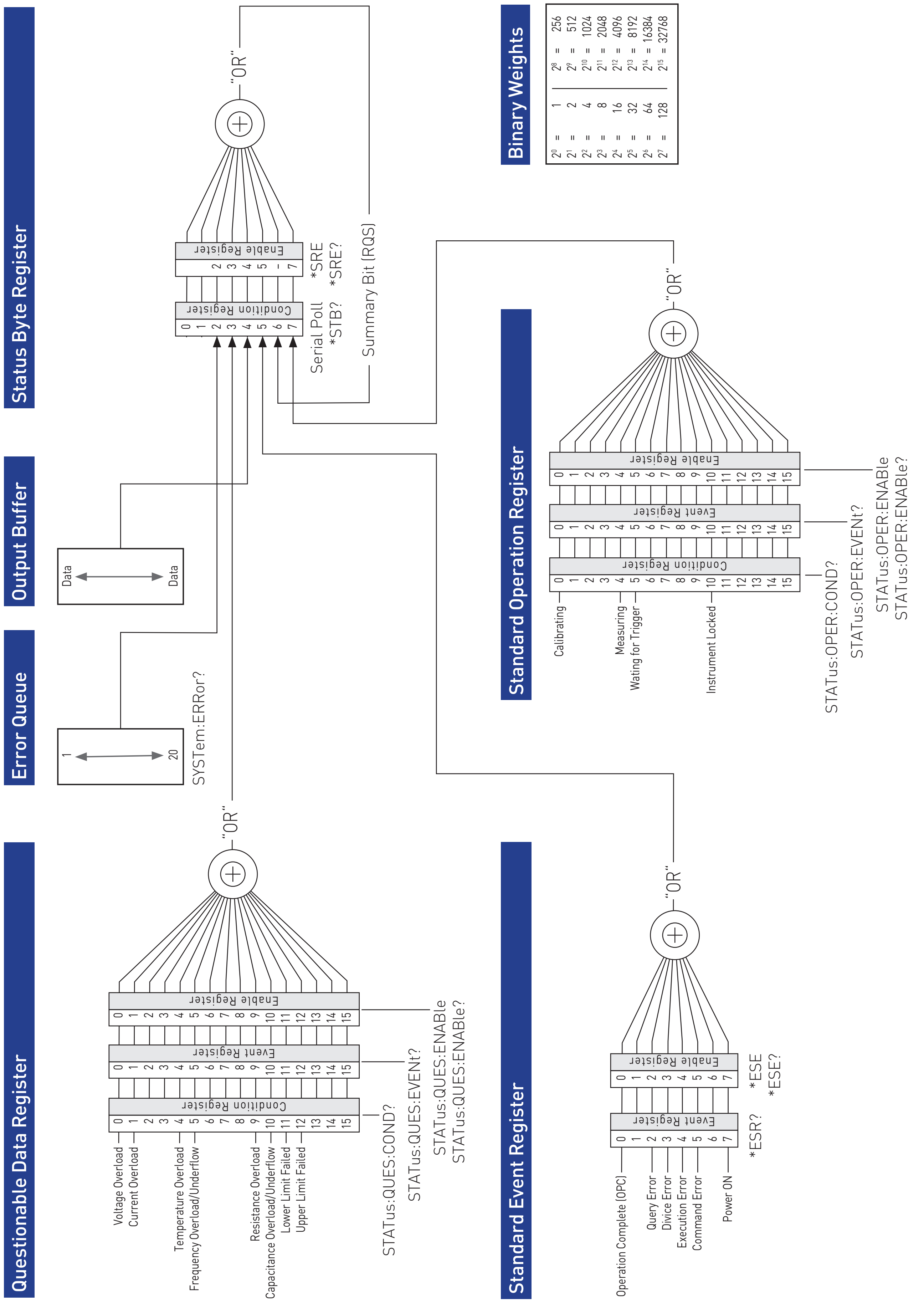
Example:

The decimal value 40 = 32 + 8 indicates that bits no. 3 and 5 in the status register (e.g. the `QUEStionable` status summary bit and the `ESB` bit in the `STatus` Byte) are set.

Error Queue

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

The error queue should be queried after every SRQ in the controller program as the entries describe the cause of an error more precisely than the status registers. Especially in the test phase of a controller program the error queue should be queried regularly since faulty commands from the controller to the instrument are recorded there as well.



2^0 = 1	2^8 = 256
2^1 = 2	2^9 = 512
2^2 = 4	2^{10} = 1024
2^3 = 8	2^{11} = 2048
2^4 = 16	2^{12} = 4096
2^5 = 32	2^{13} = 8192
2^6 = 64	2^{14} = 16384
2^7 = 128	2^{15} = 32768

2 Command Reference

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

2.1 Common Commands

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

Available common commands:

***CLS**
CLear Status

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

Usage:	Setting only

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

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

Parameters:

<Value>	Range:	0 to 255

***ESR?**
Event Status Read
Returns the contents of the event status register in decimal form and subsequently sets the register to zero.

Return values:

<Contents>	Range:	0 to 255
Usage:	Query only	

***IDN?**
IDeNtification: returns the instrument identification.

Return values:	
<ID>	HAMEG,‹device type›,‹serial number›,‹firmwareversion›
Example:	HAMEG,HMC8012,12345,01.000
Usage:	Query only

***OPC**
OPeration Complete

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

***RST**
ReSeT

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

Usage: Setting only

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

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

Parameters:	
<Contents>	Contents of the service request enable register in decimal form. Bit 6 (MSS mask bit) is always 0.
Range:	0 to 255

***STB?**
STatus Byte query

Returns the contents of the status byte in decimal form.

Usage: Query only

***TRG**
TRiGger

Triggers all actions waiting for a trigger event. In particular, *TRG generates a manual trigger signal (Manual Trigger). This common command complements the commands of the `TRIGger` subsystem.

Usage: Event

***TST?**
self TeST query

Triggers selftests of the instrument and returns an error code in decimal form (see Service Manual supplied with the instrument). „0“ indicates no errors occured.

Usage: Query only

***WAI**
WAIt to continue

Prevents servicing of the subsequent commands until all preceding commands have been executed and all signals have settled (see also command synchronization and `*OPC`).

Usage: Event

2.2 System related commands

FETCh?
Query the actual measurement value on the display in auto trigger, single trigger or manual trigger mode. In contrast to the READ? command the FETC? command does not initialize a trigger. By sending the *TRG command before the FETC? command you get the actual triggered measurement value.

Usage: Query only

READ?
Query the actual measurement value in auto trigger mode. In the single trigger mode the READ? command initialize a trigger with the settings for trigger count or interval. In the trigger manual mode the READ? command initialize a trigger. By sending the READ? command again the trigger mode stops. Please notice that the READ? command only queries the first measurement value of the trigger system.

Usage: Query only

SYSTem:BEEPer:STATe <State>
Switches the front panel control beeper on or off. For example, if the control beeper is disabled, you don't get a control beep during the manual or signgle trigger.

Parameters:

<State>	ON I OFF	
	*RST:	ON

SYSTem:BEEPer:STATe?
Returns the state of the front panel control beeper. Returns "0" for deactivated (OFF) and "1" for activated (ON) control beeper.

	*RST:	1

SYSTem:BEEPer[:IMMediate]
The instrument returns a single beep immediately.

Usage: Setting only

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

Usage: Query only

SYSTem:LOCal
Sets the system to front panel control. The front panel control is unlocked.

Usage: Setting only

SYSTem:REMote
Sets the system to remote state. The front panel control is locked and can be unlock via soft menu key „Unlock keys" (front panel) or SCPI command SYSTem:LOCal.

Usage: Setting only

SYSTem:RWLock
Sets the system to remote state. The front panel control is locked and can not be unlocked via soft menu key „Unlock keys" (front panel). You are only able to unlock the front panel control via SCPI command SYSTem:LOCal.

Usage: Setting only

SYSTem:VERSion?
Returns the version of the SCPI (= Standard Commands for Programmable Instruments) standard.

Usage: Query only

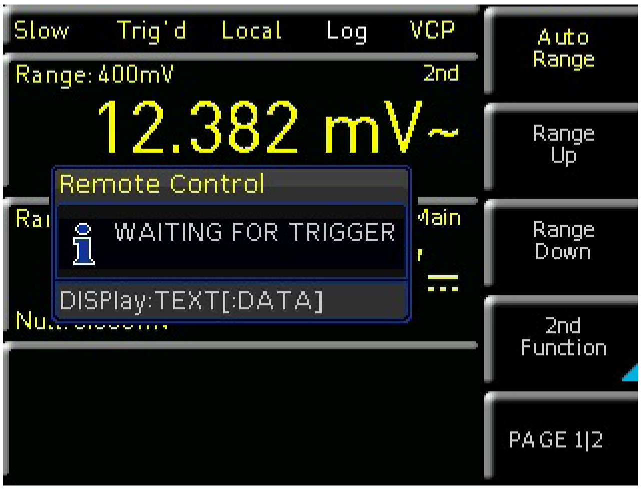
2.3 Display commands

DISPlay:TEXT:CLEar
Clears the text message box on the front display.

Usage: Setting only

DISPlay:TEXT[:DATA] „<string>“
Displays a text message box on the front display.

Example: DISP:TEXT „WAITING FOR TRIGGER“



Remote Control

WAITING FOR TRIGGER

DISPlay:TEXT[:DATA]

Fig. 2.1: Display text example

2.4 Trigger commands

TRIGger:COUNt {<Count>|MIN|MAX|DEFault}
Selects the number of triggers for the single trigger mode before returning to the „idle“ trigger state.

Parameters:

<Count> Trigger count value in single trigger mode.

<Count>:	1 to 50,000
MIN:	1
MAX:	50,000
DEFault:	1
*RST:	1

TRIGger:COUNt? [MINimum|MAXimum]
Returns the trigger count of the single trigger mode.

Return values:

<Count> 1 to 50,000

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

TRIGger:INTerval {<Seconds>|MIN|MAX|DEF}
Selects the trigger time interval for the single trigger mode before returning to the „idle“ trigger state.

Parameters:

<Interval> Trigger time interval value in single trigger mode.

<Seconds>:	0 to 3,600s
MIN:	0s
MAX:	3,600s
DEFault:	0s
*RST:	0.0E+00s

TRIGger:INTerval? [{MIN|MAX}]
Returns the trigger time interval for the single trigger mode before returning to the „idle“ trigger state.

Return values:

<Interval>	Trigger time interval value in single trigger mode.

<Seconds>:	0 to 3,600s
MIN:	0.0E+00s
MAX:	3.6000E+03s

TRIGger:LEVel {<Level>|MIN|MAX|DEF}
Selects the trigger level (threshold) for the auto trigger mode.

Parameters:

<Level>	Trigger threshold value in auto trigger mode.

<Level>:	-750V to 750V
MIN:	-750V
MAX:	750V
DEFault:	0V
*RST:	0.0E+00V

TRIGger:LEVel? [{MIN|MAX}]
Selects the trigger level (threshold) for the auto trigger mode.

Return values:

<Level>	Trigger threshold value in auto trigger mode.

<Level>:	-750V to 750V
MIN:	-7.5000E+02
MAX:	7.5000E+02

TRIGger:LEVel:MODe {CONTinue | ABOVe|BELow}
Selects the trigger level mode in the auto trigger mode.

Parameters:

‹Level Mode›	Trigger level in auto mode

CONTinue:	Continuous mode
ABOVe:	Upper threshold trigger level
BELow:	Lower threshold trigger level
*RST:	CONT

TRIGger:LEVel:MODe?
Returns the trigger level mode in the auto trigger mode.

Return values:

‹Level Mode›	Trigger level in auto mode

CONTinue:	Continuous mode
ABOVe:	Upper threshold trigger level
BELow:	Lower threshold trigger level

TRIGger:MODE {<Mode>}
Selects the trigger mode.

Parameters:

<Mode>	AUTO I MANual I SINGle	
	AUTO:	Automatic trigger mode
	MANual:	Manual trigger mode
	SINGle:	Single trigger mode
	*RST:	AUTO

TRIGger:MODE?
Returns the trigger mode.

Return values:

<Mode>	AUTO I MANual I SINGle	
	AUTO:	Automatic trigger mode
	MANual:	Manual trigger mode
	SINGle:	Single trigger mode

2.4 Configuration and Measurement Commands

2.4.1 Measurement Commands

MEASure:CAPacitance? [{<Range>|AUTO|MIN|MAX|DEF}]
Configures the instrument for capacitance measurements. The displayed values (including statistic values) will be reset and the instrument immediately triggers a measurement.

Return values:

<Range>	5nF, 50nF, 500nF, 5µF, 50µF, 500µF	
	AUTO:	Auto range
	MIN:	5nF
	MAX:	500µF
	DEFault:	5nF

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

*RST: AUTO

Usage: Query only

MEASure:CONTinuity?
Configures the instrument for continuity measurements. The displayed value will be reset and the instrument immediately triggers a measurement.

Return values:

<Value>	The command returns a single reading. Range: 4000Ω
Usage:	Query only

MEASure:CURRent:AC? [{←Range→|AUTO|MIN|MAX|DEF}]
Configures the instrument for AC I measurements. The displayed values (including statistic values) will be reset and the instrument immediately triggers a measurement.

Return values:

<Range> 20mA, 200mA, 2A, 10A

AUTO:	Auto range
MIN:	20mA
MAX:	10A
DEFault:	20mA

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

*RST: AUTO

Usage: Query only

MEASure:CURRent:DC? [{<Range>|AUTO|MIN|MAX|DEF}]
Configures the instrument for DC I measurements. The displayed values (including statistic values) will be reset and the instrument immediately triggers a measurement.

Return values:

<Range> 20mA, 200mA, 2A, 10A

AUTO:	Auto range
MIN:	20mA
MAX:	10A
DEFault:	20mA

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

*RST: AUTO

Usage: Query only

MEASure:DIODe?

Configures the instrument for diode tests. The displayed value will be reset and the instrument immediately triggers a measurement.

Return values:

<Value> The command returns a single reading.
Range: 5V

Usage: Query only

MEASure:FREQuency[:VOLTAGE]? [{<Range>|AUTO|MIN|MAX|DEF}]

Configures the instrument for frequency measurements with main function AC V. The displayed values (including statistic values) will be reset and the instrument immediately triggers a measurement. Display of frequency is only available as 2nd function for the main functions AC V and AC I.

Return values:

<Range>

AC voltage:	400mV, 4V, 40V, 400V, 750V (5Hz to 700kHz)
AUTO:	Auto range
MIN:	400mV
MAX:	750V
DEFault:	400mV

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

Usage: Query only

MEASure:FREQuency:CURRent [{<Range>|AUTO|MIN|MAX|DEF}]

Configures the instrument for frequency measurements with main function AC I. The displayed values (including statistic values) will be reset and the instrument immediately triggers a measurement. Display of frequency is only available as 2nd function for the main functions AC V and AC I.

Return values:

<Range>

AC current:	20mA, 200mA (5Hz to 10kHz) 2A, 10A (5Hz to 5kHz)
AUTO:	Auto range
MIN:	20mA
MAX:	10A
DEFault:	20mA

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

Usage: Query only

MEASure:FRESistance? [{<Range>|AUTO|MIN|MAX|DEF}]
Configures the instrument for 4-wire resistance measurements. The displayed values (including statistic values) will be reset and the instrument immediately triggers a measurement.

Return values:

<Range>	400Ω, 4kΩ, 40kΩ, 400kΩ, 4MΩ
	AUTO: Auto range MIN: 400Ω MAX: 4MΩ DEFault: 400Ω
	If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.
Usage:	Query only

MEASure:RESistance? [{<Range>|AUTO|MIN|MAX|DEF}]
Configures the instrument for 2-wire resistance measurements. The displayed values (including statistic values) will be reset and the instrument immediately triggers a measurement.

Return values:

<Range>	400Ω, 4kΩ, 40kΩ, 400kΩ, 4MΩ, 40MΩ, 250MΩ
	AUTO: Auto range MIN: 400Ω MAX: 250MΩ DEFault: 400Ω
	If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.
Usage:	Query only

MEASure:TEMPerature? [{<Probe_Type>|DEF}[,{<Type>|DEF}]
Configures the instrument for temperature measurements. The displayed values (including statistic values) will be reset and the instrument immediately triggers a measurement.

Return values:

<Probe_Type>	FRTD I RTD
	FRTD: 4-wire temperature value (4W) RTD: 2-wire temperature value (2W) DEFault: RTD (2W)
<Type>	PT100 I PT500 I PT1000 DEFault: PT100
Example:	MEAS:TEMP? FRTD,PT500
Usage:	Query only

MEASure[:VOLTage]:AC? [{<Range>|AUTO|MIN|MAX|DEF}]

Configures the instrument for AC V measurements. The displayed values (including statistic values) will be reset and the instrument immediately triggers a measurement.

Return values:

<Range> 400mV, 4V, 40V, 400V, 750V

AUTO:	Auto range
MIN:	400mV
MAX:	750V
DEFault:	400mV

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

Usage: Query only

MEASure[:VOLTage][:DC]? [{<Range>|AUTO|MIN|MAX|DEF}]

Configures the instrument for DC V measurements. The displayed values (including statistic values) will be reset and the instrument immediately triggers a measurement.

Return values:

<Range> 400mV, 4V, 40V, 400V, 1000V

AUTO:	Auto range
MIN:	400mV
MAX:	1000V
DEFault:	400mV

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

Usage: Query only

2.4.2 Configuration commands

Capacitance configuration commands

CONFigure:CAPacitance [{<Range>|AUTO|MIN|MAX|DEF}]
Configures the instrument for capacitance measurements, but does not initiate a measurement. Use the READ? command to query the capacitance measurement value with specified range or the CONF? command to query the capacitance function configuration.

Parameters:
<Range> 5nF, 50nF, 500nF, 5µF, 50µF, 500µF

AUTO:	Auto range
MIN:	5nF
MAX:	500µF
DEFault:	5nF

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

*RST: AUTO

[SENSe:]CAPacitance:NULL[:STATe] {ON|OFF}
Turns the null function for capacitance measurements on or off. In this case, the instrument does not switch into capacitance mode. For activating the capacitance measurement function use the FUNC command.

Parameters:
<State> ON | OFF

*RST: OFF

[SENSe:]CAPacitance:NULL[:STATe]?
Returns the NULL function state of the capacitance measurement function.

Return values:
<State> 1 | 0

1	ON - NULL function of the capacitance measurement function is activated
0	OFF - NULL function of the capacitance measurement function is disabled

[SENSe:]CAPacitance:NULL:VALue {<Value>|MIN|MAX}
Sets the null value of the capacitance measurement function.

Parameters:

<Value>	0 to 500µF (adjustable in 1fF steps)	
	MIN:	0F
	MAX:	500µF
	*RST:	0F

[SENSe:]CAPacitance:NULL:VALue? [{MIN|MAX}]
Returns the null value of the capacitance measurement function.

Return values:

<Value>	0 to 500µF	
	MIN:	0.0E+00
	MAX:	5.000000E-04

[SENSe:]CAPacitance:RANGe:AUTO <Mode>
Activates or disables the auto range mode of the capacitance measurement function.

Parameters:

<Mode>	ON I OFF I 1 I 0	
	ON:	Activates the auto range mode of the capacitance measurement function
	OFF:	Disables the auto range mode of the capacitance measurement function
	1:	Activates the auto range mode of the capacitance measurement function
	0:	Disables the auto range mode of the capacitance measurement function
	*RST:	ON

[SENSe:]CAPacitance:RANGe:AUTO?
Returns the auto mode state of the capacitance measurement function.

Return values:

<State>	1 I 0	
	1:	Auto range of the capacitance measurement function is activated
	0:	Auto range of the capacitance measurement function is disabled

[SENSe:]CAPacitance:RANGe[:UPPer] {<Range>|MIN|MAX|DEF}
Sets the capacitance measurement function range.

Parameters:

<Range>	5nF, 50nF, 500nF, 5µF, 50µF, 500µF	
	MIN:	5nF
	MAX:	500µF
	DEFault:	5nF
	*RST:	AUTO

[SENSe:]CAPacitance:RANGe[:UPPer]? [{MIN|MAX|DEF}]

Returns the state of the capacitance measurement function range.

Return values:

<Range>	5nF, 50nF, 500nF, 5µF, 50µF, 500µF
	MIN: 5.00000000E-09 MAX: 5.00000000E-04 DEF: 5.00000000E-09

Continuity configuration commands

CONFigure:CONTinuity

Configures the instrument for continuity measurements, but does not initiate a measurement. Use the READ? command to query the continuity measurement value or the CONF? command to query the general function configuration.

[SENSe:]CONTinuity:THReshold {<Threshold>|MIN|MAX|DEF}

Sets the threshold of the continuity measurement function. In this case, the instrument does not switch into continuity mode. For activating the continuity measurement function use the FUNC command.

Parameters:

<Threshold>	0Ω to 1MΩ (adjustable in 1Ω steps)
	MIN: 0Ω MAX: 1MΩ DEFault: 200Ω
	*RST: 200Ω

[SENSe:]CONTinuity:THReshold? [{MIN|MAX|DEF}]

Returns the threshold of the continuity measurement function.

Return values:

<Threshold>	0Ω to 1MΩ
	MIN: 0.00000000E+00 MAX: 1.00000000E+06 DEFault: 2.00000000E+02

[SENSe:]CONTinuity:BEEPer[:STATe] {ON|OFF}

Activates or disables the beeper of the continuity measurement function.

Parameters:

<State>	ON I OFF	
	ON:	Activates the beeper function of the continuity measurement function
	OFF:	Disables the beeper function of the continuity measurement function
	*RST:	OFF

[SENSe:]CONTinuity:BEEPer[:STATe]?

Returns the beeper state of the continuity measurement function.

Return values:

<State>	1 I 0	
	1:	Beeper of the continuity measurement function is activated
	0:	Beeper of the continuity measurement function is disabled

AC I configuration commands

CONFigure:CURRent:AC [{<Range>|AUTO|MIN|MAX|DEF}]

Configures the instrument for AC I measurements, but does not initiate a measurement. Use the READ? command to query the AC I measurement value with specified range or the CONF? command to query the AC I function configuration.

Parameters:

<Range>	20mA, 200mA, 2A, 10A	
	AUTO:	Auto range
	MIN:	20mA
	MAX:	10A
	DEFault:	20mA

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

*RST: AUTO

[SENSe:]CURRent:AC:BANDwidth {<Threshold>|MIN|MAX|DEF}
Sets the AC filter of the AC I measurement function.

Parameters:

<Threshold>	10 I 50 I 400	
	10:	10Hz filter (Slow)
	50:	50Hz filter (Medium)
	400:	400Hz filter (Fast)
	MIN:	Slow
	MAX:	Fast
	DEF:	Slow
	*RST:	5.00000000E+01

[SENSe:]CURRent:AC:BANDwidth? [{MIN|MAX}]
Returns the AC filter state of the AC I measurement function.

Return values:

<State>	1.00000000E+01 I 5.0000000E+1 I 4.00000000E+02	
	MIN:	1.00000000E+01 (Slow)
	MAX:	4.00000000E+02 (Fast)

[SENSe:]CURRent:AC:NULL[:STATe] {ON|OFF}
Turns the null function for AC I measurements on or off. In this case, the instrument does not switch into AC I mode. For activating the AC I measurement function use the FUNC command.

Parameters:

<State>	ON I OFF	
	ON:	Null function of AC I measurements is activated
	OFF:	Null function of AC I measurements is disabled
	*RST:	OFF

[SENSe:]CURRent:AC:NULL[:STATe]?
Returns the NULL function state of the AC I measurement function.

Return values:

<State>	1 I 0	
	1	ON - NULL function activated
	0	OFF - NULL function disabled

[SENSe:]CURRent:AC:NULL:VALue {<Value>|MIN|MAX}

Sets the null value of the AC I measurement function.

Parameters:

<Value>	-10V to 10A (adjustable in 10pA steps)	
	MIN:	-10A
	MAX:	10A
	*RST:	0A

[SENSe:]CURRent:AC:NULL:VALue? [{MIN|MAX}]

Returns the null value of the AC I measurement function.

Return values:

<Value>	-10A to 10A	
	MIN:	-1.000000E+01
	MAX:	1.000000E+01

[SENSe:]CURRent:AC:RANGe:AUTO <Mode>

Activates or disables the auto range mode of the AC I measurement function.

Parameters:

<Mode>	ON I OFF I 1 I 0	
	ON:	Activates the auto range mode of the AC I measurement function
	OFF:	Disables the auto range mode of the AC I measurement function
	1:	Activates the auto range mode of the AC I measurement function
	0:	Disables the auto range mode of the AC I measurement function
	*RST:	ON

[SENSe:]CURRent:AC:RANGe:AUTO?

Returns the auto mode state of the AC I measurement function.

Return values:

<State>	1 I 0	
	1:	Auto range of the AC I measurement function is activated
	0:	Auto range of the AC I measurement function is disabled

[SENSe:]CURRent:AC:RANGe[:UPPer] {<Range>|MIN|MAX|DEF}

Sets the AC I measurement function range.

Parameters:

<Range>	20mA, 200mA, 2A, 10A	
	MIN:	20e-3
	MAX:	1e+1
	DEFault:	20e-3
	*RST:	AUTO

[SENSe:]CURRent:AC:RANGe[:UPPer]? [{MIN|MAX}]
Returns the state of the AC I measurement function range.

Return values:

<Range>	20mA, 200mA, 2A, 10A
	MIN: 2.00000000E-02
	MAX: 1.00000000E+01
	DEFault: 2.00000000E-02

AC V configuration commands

CONFigure[:VOLTage]:AC [{<Range>|AUTO|MIN|MAX|DEF}]
Configures the instrument for AC V measurements, but does not initiate a measurement. Use the READ? command to query the AC V measurement value with specified range or the CONF? command to query the AC V function configuration.

Parameter:

<Range>	400mV, 4V, 40V, 400V, 750V
	AUTO: Auto range
	MIN: 400mV
	MAX: 750V
	DEFault: 400mV

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

*RST: AUTO

[SENSe:]VOLTage:AC:BANDwidth {<Filter>|MIN|MAX|DEF}
Sets the AC filter of the AC V measurement function. In this case, the instrument does not switch into AC V mode. For activating the AC V measurement function use the FUNC command.

Parameters:

<Filter>	10 | 50 | 400
	10: 10Hz filter (Slow)
	50: 50Hz filter (Medium)
	400: 400Hz filter (Fast)
	MIN: Slow
	MAX: Fast
	DEFault: Slow
	*RST: 5.00000000E+01

[SENSe:]VOLTage:AC:BANDwidth? [{MIN|MAX}]

Returns the AC filter of the AC V measurement function.

Return values:

<Filter>	1.00000000E+01 I 5.00000000E+01 I 4.00000000E+02	
	MIN:	1.00000000E+01
	MAX:	4.00000000E+02

[SENSe:]VOLTage:AC:NULL[:STATe] {ON|OFF}

Turns the null function for AC V measurement function on or off.

Parameters:

<State>	ON I OFF	
	ON:	Null function of AC V measurement function is activated
	OFF:	Null function of AC V measurement function is disabled
	*RST:	OFF

[SENSe:]VOLTage:AC:NULL[:STATe]?

Returns the null function state of the AC V measurement function.

Return values:

<State>	1 I 0	
	1	ON - Null function of the AC V measurement function is activated
	0	OFF - Null function of the AC V measurement function is disabled

[SENSe:]VOLTage:AC:NULL:VALue {<Value>|MIN|MAX}

Sets the null value of the AC V measurement function.

Parameters:

<Value>	-750V to 750V (adjustable in 1nV steps)	
	MIN:	-750V
	MAX:	750V
	*RST:	0Ω

[SENSe:]VOLTage:AC:NULL:VALue? [{MIN|MAX}]

Returns the null value of the AC V measurement function.

Return values:

<Value>	-750V to 750V	
	MIN:	-7.500000E+02
	MAX:	7.500000E+02

[SENSe:]VOLTage:AC:RANGe:AUTO <Mode>
Activates or disables the auto range mode of the AC V measurement function.

Parameters:

<Mode>	ON I OFF I 1 I 0	
	ON:	Activates the auto range mode of the AC V measurement function
	OFF:	Disables the auto range mode of the AC V measurement function
	1:	Activates the auto range mode of the AC V measurement function
	0:	Disables the auto range mode of the AC V measurement function

[SENSe:]VOLTage:AC:RANGe:AUTO?
Returns the auto mode state of the AC V measurement function.

Return values:

<State>	1 I 0	
	1:	Auto range of the AC V measurement function is activated
	0:	Auto range of the AC V measurement function is disabled

[SENSe:]VOLTage:AC:RANGe[:UPPer] {<Range>|MIN|MAX|DEF}
Sets the AC V measurement function range.

Parameters:

<Range>	400mV, 4V, 40V, 400V, 750V	
	MIN:	400mV
	MAX:	750V
	DEFault:	400mV
	*RST:	AUTO

[SENSe:]VOLTage:AC:RANGe[:UPPer]? [{MIN|MAX}]
Returns the state of the AC V measurement function range.

Return values:

<Range>	400mV, 4V, 40V, 400V, 750V	
	MIN:	4.00000000E-01
	MAX:	7.50000000E+02

DC I configuration commands

CONFigure:CURRent[:DC] [{<Range>|AUTO|MIN|MAX|DEF}]

Configures the instrument for DC I measurements, but does not initiate a measurement. Use the READ? command to query the DC I measurement value with specified range or the CONF? command to query the DC I function configuration.

Parameters:

<Range> 20mA, 200mA, 2A, 10A

AUTO:	Auto range
MIN:	20mA
MAX:	10A
DEFault:	20mA

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

*RST: AUTO

[SENSe:]CURRent[:DC]:NULL[:STATe] {ON|OFF}

Turns the null function for DC I measurements on or off. In this case, the instrument does not switch into DC I mode. For activating the DC I measurement function use the FUNC command.

Parameters:

<State> ON | OFF

*RST: OFF

[SENSe:]CURRent[:DC]:NULL[:STATe]?

Returns the null function state of the DC I measurement function.

Return values:

<State> 1 | 0

1	ON - NULL function activated
0	OFF - NULL function disabled

[SENSe:]CURRent[:DC]:NULL:VALue {<Value>|MIN|MAX}

Sets the null value of the DC I measurement function.

Parameters:

<Value>	-10A to 10A (adjustable in 10pA steps)	
	MIN:	-10A
	MAX:	10A
	*RST:	0A

[SENSe:]CURRent[:DC]:NULL:VALue? [{MIN|MAX}]

Returns the null value of the DC I measurement function.

Return values:

<Value>	-10A to 10A	
	MIN:	-1.000000E+01
	MAX:	1.000000E+01

[SENSe:]CURRent[:DC]:RANGe:AUTO <Mode>

Activates or disables the auto range mode of the DC I measurement function.

Parameters:

<Mode>	ON I OFF I 1 I 0	
	ON:	Activates the auto range mode of the DC I measurement function
	OFF:	Disables the auto range mode of the DC I measurement function
	1:	Activates the auto range mode of the DC I measurement function
	0:	Disables the auto range mode of the DC I measurement function
	*RST:	ON

[SENSe:]CURRent[:DC]:RANGe:AUTO?

Returns the auto mode state of the DC I measurement function.

Return values:

<State>	1 I 0	
	1:	Auto range of the AC I measurement function is activated
	0:	Auto range of the AC I measurement function is disabled

[SENSe:]CURRent[:DC]:RANGe[:UPPer] {<Range>|MIN|MAX|DEF}

Sets the DC I measurement function range.

Parameters:

<Range>	20mA, 200mA, 2A, 10A	
	MIN:	20e-3
	MAX:	1e+1
	DEFault:	20e-3
	*RST:	AUTO

[SENSe:]CURRent[:DC]:RANGe[:UPPer]? [{MIN|MAX}]

Returns the state of the DC I measurement function range.

Return values:

<Range>	20mA, 200mA, 2A, 10A
	MIN: 2.00000000E-02 MAX: 1.00000000E+01 DEFault: 2.00000000E-02

DC V configuration commands

CONFigure[:VOLTage][:DC] [{<Range>|AUTO|MIN|MAX|DEF}]

Configures the instrument for DC V measurements, but does not initiate a measurement. Use the READ? command to query the DC V measurement value with specified range or the CONF? command to query the DC V function configuration.

Parameter:

<Range>	400mV, 4V, 40V, 400V, 1000V
	AUTO: Auto range MIN: 400mV MAX: 1000V DEFault: 400mV
	If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.
	*RST: AUTO

[SENSe:]VOLTage[:DC]:NULL[:STATe] {ON|OFF}

Turns the null function for DC V measurement function on or off.

Parameters:

<State>	ON I OFF
	ON: Null function of DC V measurement function is activated OFF: Null function of DC V measurement function is disabled
	*RST: OFF

[SENSe:]VOLTage[:DC]:NULL[:STATe]?

Returns the null function state of the DC V measurement function.

Return values:

<State>	1 I 0	
	1	ON - Null function of the DC V measurement function is activated
	0	OFF - Null function of the DC V measurement function is disabled

[SENSe:]VOLTage[:DC]:NULL:VALue {<Value>|MIN|MAX}

Sets the null value of the DC V measurement function.

Parameters:

<Value>	-1000V to 1000V (adjustable in 1nV steps)	
	MIN:	-1000V
	MAX:	1000V
	*RST:	0V

[SENSe:]VOLTage[:DC]:NULL:VALue? [{MIN|MAX}]

Returns the null value of the DC V measurement function.

Return values:

<Value>	-1000V to 1000V	
	MIN:	-1.000000E+03
	MAX:	1.000000E+03

[SENSe:]VOLTage[:DC]:RANGe:AUTO <Mode>

Activates or disables the auto range mode of the DC V measurement function.

Parameters:

<Mode>	ON I OFF I 1 I 0	
	ON:	Activates the auto range mode of the DC V measurement function
	OFF:	Disables the auto range mode of the DC V measurement function
	1:	Activates the auto range mode of the DC V measurement function
	0:	Disables the auto range mode of the DC V measurement function

[SENSe:]VOLTage[:DC]:RANGe:AUTO?

Returns the auto mode state of the DC V measurement function.

Return values:

<State>	1 I 0	
	1:	Auto range of the DC V measurement function is activated
	0:	Auto range of the DC V measurement function is disabled

[SENSe:]VOLTage[:DC]:RANGe[:UPPer] {<Range>|MIN|MAX|DEF}

Sets the DC V measurement function range.

Parameters:

<Range>	400mV, 4V, 40V, 400V, 1000V	
	MIN:	400mV
	MAX:	1000V
	DEFault:	1000V
	*RST:	AUTO

[SENSe:]VOLTage[:DC]:RANGe[:UPPer]? [{MIN|MAX}]

Returns the state of the DC V measurement function range.

Return values:

<Range>	400mV, 4V, 40V, 400V, 1000V	
	MIN:	4.00000000E-01
	MAX:	1.00000000E+03

[SENSe:]VOLTage[:DC]:ZERO:AUTO <Mode>

Activates or disables the auto zero mode of the DC V measurement function.

Parameters:

<Mode>	ON I OFF I 1 I 0	
	ON:	Activates the auto zero mode of the DC V measurement function
	OFF:	Disables the auto zero mode of the DC V measurement function
	1:	Activates the auto zero mode of the DC V measurement function
	0:	Disables the auto zero mode of the DC V measurement function

[SENSe:]VOLTage[:DC]:ZERO:AUTO?

Returns the auto mode state of the DC V measurement function.

Return values:

<State>	1 I 0	
	1:	Auto zero mode of the DC V measurement function is activated
	0:	Auto zero mode of the DC V measurement function is disabled

Diode configuration commands

CONFigure:DIODe
Configures the instrument for diode measurements, but does not initiate a measurement. Use the READ? command to query the diode measurement value or the CONF? command to query the general function configuration.

[SENSe:]DIODe:THReshold {<Threshold>|MIN|MAX|DEF}
Sets the threshold of the diode measurement function. In this case, the instrument does not switch into continuity mode. For activating the continuity measurement function use the FUNC command.

Parameters:

<Threshold>	0V to 5V (adjustable in 1µV steps)
	MIN: 0V MAX: 5V DEFault: 700mV
	*RST: 700mV

[SENSe:]DIODe:THReshold? [{MIN|MAX|DEF}]
Returns the threshold of the diode measurement function.

Return values:

<Threshold>	0V to 5V
	MIN: 0.00000000E+00 MAX: 4.94999981E+00 DEFault: 6.99999988E-01

[SENSe:]DIODe:BEEPer[:STATe] {ON|OFF}
Turns the beeper function for diode measurements on or off.

Parameters:

<State>	ON I OFF
	ON: Beeper of the diode measurement function is activated OFF: Beeper of the diode measurement function is disabled
	*RST: OFF

[SENSe:]DIODe:BEEPer[:STATe]?

Returns the beeper function state of the diode measurement function.

Return values:

<State>	1 | 0	
	1	ON - beeper of the diode measurement function is activated
	0	OFF - beeper of the diode measurement function is disabled

Frequency configuration commands

CONFigure:FREQuency[:VOLTAGE]

Configures the instrument for frequency measurements, but does not initiate a measurement. Use the READ? command to query the frequency measurement value or the CONF? command to query the frequency function configuration in AC V mode.

CONFigure:FREQuency:CURRent

Configures the instrument for frequency measurements, but does not initiate a measurement. Use the READ? command to query the frequency measurement value or the CONF? command to query the frequency function configuration in AC I mode.

[SENSe:]FREQuency:APERture {<Seconds>|MIN|MAX|DEF}

Sets the gate time of the frequency measurement function in AC V / AC I mode.

Parameters:

<Seconds>	10ms, 100ms, 1s	
	MIN:	10ms
	MAX:	1s
	DEFault:	1s
	*RST:	1s

[SENSe:]FREQuency:APERture? [{MIN|MAX}]

Returns the gate time of the frequency measurement function in AC V / AC I mode.

Return values:

<GateTime>	1.00000000E-02, 1.00000000E-01, 1.00000000E+00	
	MIN:	1.00000000E-02
	MAX:	1.00000000E+00

[SENSe:]FREQuency:CURRent:RANGe:AUTO <Mode>

Activates or disables the auto range mode of the AC I measurement function for frequency measurements.

Parameters:

<Mode>	ON I OFF I 1 I 0	
	ON:	Activates the auto range mode of the AC I measurement function
	OFF:	Disables the auto range mode of the AC I measurement function
	1:	Activates the auto range mode of the AC I measurement function
	0:	Disables the auto range mode of the AC I measurement function
	*RST:	ON

[SENSe:]FREQuency:CURRent:RANGe:AUTO?

Returns the auto range state of the AC I measurement function for frequency measurements.

Parameters:

<Mode>	1 I 0	
	1:	Auto range mode of the AC I measurement function is activated
	0:	Auto range mode of the AC I measurement function is disabled

[SENSe:]FREQuency:CURRent:RANGe[:UPPer] {<Current_Range>|MIN|MAX|DEF}

Sets the current range of the AC I measurement function for frequency measurements.

Parameters:

<Current_Range>	20mA, 200mA, 2A, 10A	
	MIN:	20mA
	MAX:	10A
	DEFault:	20mA
	*RST:	AUTO

[SENSe:]FREQuency:CURRent:RANGe[:UPPer]? [{MIN|MAX}]

Returns the current range of the AC I measurement function for frequency measurements.

Return values:

<Current_Range>	20mA, 200mA, 2A, 10A	
	MIN:	2.00000000E-02
	MAX:	1.00000000E+01
	DEFault:	2.00000000E-02

[SENSe:]FREQuency:VOLTage:RANGe:AUTO <Mode>
Activates or disables the auto range mode of the AC V measurement function.

Parameters:
<Mode> ON | OFF | 1 | 0

ON: Activates the auto range mode of the AC V measurement function
OFF: Disables the auto range mode of the AC V measurement function
1: Activates the auto range mode of the AC V measurement function
0: Disables the auto range mode of the AC V measurement function

*RST: ON

[SENSe:]FREQuency:VOLTage:RANGe:AUTO?
Returns the auto range state of the AC V measurement function for frequency measurements.

Parameters:
<Mode> 1 | 0

1: Auto range mode of the AC V measurement function is activated
0: Auto range mode of the AC V measurement function is disabled

[SENSe:]FREQuency:VOLTage:RANGe[:UPPer] {<Voltlage_Range>|MIN|MAX|DEF}
Sets the voltage range of the AC V measurement function for frequency measurements.

Parameters:
<Voltlage_Range> 400mV, 4V, 40V, 400V, 750V

MIN: 400mV
MAX: 750V
DEFault: 400mV

*RST: AUTO

[SENSe:]FREQuency:VOLTage:RANGe[:UPPer]? [{MIN|MAX}]
Returns the voltage range of the AC V measurement function for frequency measurements.

Return values:
<Voltlage_Range> 400mV, 4V, 40V, 400V, 750V

MIN: 4.00000000E-01
MAX: 7.50000000E+02
DEFault: 4.00000000E-01

4-wire resistance configuration commands

CONFigure:FRESistance [{←Range→|AUTO|MIN|MAX|DEF}]

Configures the instrument for 4-wire resistance measurements, but does not initiate a measurement. Use the READ? command to query the 4-wire resistance measurement value with specified range or the CONF? command to query the 4-wire resistance function configuration.

Parameter:

<Range>	400Ω, 4kΩ, 40kΩ, 400kΩ, 4MΩ

AUTO:	Auto range
MIN:	400Ω
MAX:	4MΩ
DEFault:	400Ω

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

*RST: AUTO

[SENSe:]FRESistance:NULL[:STATe] {ON|OFF}

Turns the null function for 4-wire (4w) resistance measurement function on or off. In this case, the instrument does not switch into resistance mode. For activating the Ω measurement function use the FUNC command.

Parameters:

<State>	ON I OFF

ON:	Null function of 4w resistance measurement function is activated
OFF:	Null function of 4w resistance measurement function is disabled

*RST: OFF

[SENSe:]FRESistance:NULL[:STATe]?

Returns the null function state of the 4-wire (4w) resistance measurement function.

Return values:

<State>	1 I 0

1	ON - Null function of the 4w resistance measurement function is activated
0	OFF - Null function of the 4w resistance measurement function is disabled

[SENSe:]FRESistance:NULL:VALue {<Value>|MIN|MAX}
Sets the null value of the 4-wire (4w) resistance measurement function.

Parameters:

<Value>	0Ω to 4MΩ (adjustable in 1Ω steps)	
	MIN:	0Ω
	MAX:	4MΩ
	*RST:	0Ω

[SENSe:]FRESistance:NULL:VALue? [{MIN|MAX}]
Returns the null value of the 4-wire (4w) resistance measurement function.

Return values:

<Value>	0Ω to 4MΩ	
	MIN:	0.0E+00
	MAX:	4.000000E+06

[SENSe:]FRESistance:RANGe:AUTO <Mode>
Activates or disables the auto range mode of the 4-wire (4w) resistance measurement function.

Parameters:

<Mode>	ON I OFF I 1 I 0	
	ON:	Activates the auto range mode of the 4w resistance measurement function
	OFF:	Disables the auto range mode of the 4w resistance measurement function
	1:	Activates the auto range mode of the 4w resistance measurement function
	0:	Disables the auto range mode of the 4w resistance measurement function
	*RST:	ON

[SENSe:]FRESistance:RANGe:AUTO?
Returns the auto mode state of the 4-wire (4w) resistance measurement function.

Return values:

<State>	1 I 0	
	1:	Auto range of the 4w resistance measurement function is activated
	0:	Auto range of the 4w resistance measurement function is disabled

[SENSe:]FRESistance:RANGe[:UPPer] {<Range>|MIN|MAX|DEF}
Sets the 4-wire (4w) resistance measurement function range.

Parameters:

<Range>	400Ω, 4kΩ, 40kΩ, 400kΩ, 4MΩ	
	MIN:	400Ω
	MAX:	4MΩ
	DEFault:	400Ω
	*RST:	AUTO

[SENSe:]FRESistance:RANGe[:UPPer]? [{MIN|MAX}]
Returns the state of the 4-wire (4w) resistance measurement function range.

Return values:

<Range>	400Ω, 4kΩ, 40kΩ, 400kΩ, 4MΩ
	MIN: 4.00000000E+02
	MAX: 4.00000000E+06

2-wire resistance configuration commands

CONFigure:RESistance [{<Range>|AUTO|MIN|MAX|DEF}]
Configures the instrument for 2-wire resistance measurements, but does not initiate a measurement. Use the READ? command to query the 2-wire resistance measurement value with specified range or the CONF? command to query the 2-wire resistance function configuration.

Parameter:

<Range> 400Ω, 4kΩ, 40kΩ, 400kΩ, 4MΩ, 40MΩ, 250MΩ

AUTO:	Auto range
MIN:	400Ω
MAX:	250MΩ
DEFault:	400Ω

If the input signal is greater than can be measured on the selected range (manual ranging), the instrument returns 9.90000000E+37.

*RST: AUTO

[SENSe:]RESistance:NULL[:STATe] {ON|OFF}
Turns the null function for 2-wire (2w) resistance measurement function on or off. In this case, the instrument does not switch into resistance mode. For activating the Ω measurement function use the FUNC command.

Parameters:

<State> ON | OFF

ON:	Null function of 2w resistance measurement function is activated
OFF:	Null function of 2w resistance measurement function is disabled

*RST: OFF

[SENSe:]RESistance:NULL[:STATe]?

Returns the null function state of the 2-wire (2w) resistance measurement function.

Return values:

<State>	1 I 0	
	1	ON - Null function of the 2w resistance measurement function is activated
	0	OFF - Null function of the 2w resistance measurement function is disabled

[SENSe:]RESistance:NULL:VALue {<Value>|MIN|MAX}

Sets the null value of the 2-wire (2w) resistance measurement function.

Parameters:

<Value>	0Ω to 250MΩ (adjustable in 1Ω steps)	
	MIN:	0Ω
	MAX:	250MΩ
	*RST:	0Ω

[SENSe:]RESistance:NULL:VALue? [{MIN|MAX}]

Returns the null value of the 2-wire (2w) resistance measurement function.

Return values:

<Value>	0Ω to 250MΩ	
	MIN:	0.0E+00
	MAX:	2.500000E+08

[SENSe:]RESistance:RANGe:AUTO <Mode>

Activates or disables the auto range mode of the 2-wire (2w) resistance measurement function.

Parameters:

<Mode>	ON I OFF I 1 I 0	
	ON:	Activates the auto range mode of the 2w resistance measurement function
	OFF:	Disables the auto range mode of the 2w resistance measurement function
	1:	Activates the auto range mode of the 2w resistance measurement function
	0:	Disables the auto range mode of the 2w resistance measurement function
	*RST:	ON

[SENSe:]RESistance:RANGe:AUTO?

Returns the auto mode state of the 2-wire (2w) resistance measurement function.

Return values:

<State>	1 I 0	
	1:	Auto range of the 2w resistance measurement function is activated
	0:	Auto range of the 2w resistance measurement function is disabled

[SENSe:]RESistance:RANGe[:UPPer] {<Range>|MIN|MAX|DEF}

Sets the 2-wire (2w) resistance measurement function range.

Parameters:

<Range>	400Ω, 4kΩ, 40kΩ, 400kΩ, 4MΩ, 40MΩ, 250MΩ	
	MIN:	400Ω
	MAX:	250MΩ
	DEFault:	400Ω
	*RST:	AUTO

[SENSe:]RESistance:RANGe[:UPPer]? [{MIN|MAX}]

Returns the state of the 2-wire (2w) resistance measurement function range.

Return values:

<Range>	400Ω, 4kΩ, 40kΩ, 400kΩ, 4MΩ, 40MΩ, 250MΩ	
	MIN:	4.00000000E+02
	MAX:	2.50000000E+08

Temperature configuration commands

CONFigure:TEMPerature [{<Probe_Type>|DEF}][,{<Type>|DEF}][,1]

Configures the instrument for 2-wire / 4-wire temperature measurements, but does not initiate a measurement. Use the READ? command to query the 2-wire / 4-wire temperature measurement value with specified range or the CONF? command to query the 2-wire resistance function configuration.

Parameter:

<Probe_Type>	FRTD I RTD	
	FRTD:	4-wire temperature value (4W)
	RTD:	2-wire temperature value (2W)
	DEF:	RTD (2W)
<Type>	PT100 I PT500 I PT1000	
	DEF:	PT100
Example:	CONF:TEMP FRTD,PT500	

[SENSe:]TEMPerature:NULL[:STATe] {ON|OFF}
Turns the null function for temperature measurement function on or off. In this case, the instrument does not switch into temperature mode (SENSOR). For activating the SENSOR measurement function use the FUNC command.

Parameters:

<State>	ON I OFF	
	ON:	Null function of temperature measurement function is activated
	OFF:	Null function of temperature measurement function is disabled
	*RST:	OFF

[SENSe:]TEMPerature:NULL[:STATe]?
Returns the null function state of the temperature measurement function.

Return values:

<State>	1 I 0	
	1	ON - Null function of the temperature measurement function is activated
	0	OFF - Null function of the temperature measurement function is disabled

[SENSe:]TEMPerature:NULL:VALue {<Value>|MIN|MAX}
Sets the null value of the temperature measurement function.

Parameters:

<Value>	-273,1°C to 999,9°C (adjustable in 1µ°C steps)	
	MIN:	-273,1°C
	MAX:	999,9°C
	*RST:	0°C

[SENSe:]TEMPerature:NULL:VALue? [{MIN|MAX}]
Returns the null value of the temperature measurement function.

Return values:

<Value>	-273,1°C to 999,9°C	
	MIN:	-2.731000E+02
	MAX:	9.999000E+02

[SENSe:]TEMPerature:TRANsducer:RTD:TYPE <Type>
Sets the RTD type (PT type) of the temperature measurement function.

Parameters:

<Type>	PT100 I PT500 I PT1000	
	*RST:	PT100

[SENSe:]TEMPerature:TRANsducer:RTD:TYPE?

Returns the RTD type (PT type) of the temperature measurement function.

Return values:

<Type>	PT100 I PT500 I PT1000

[SENSe:]TEMPerature:TRANsducer:TYPE <Probe_Type>

Sets the probe type of the temperature measurement function.

Parameters:

<Probe_Type>	FRTD I RTD	
	FRTD:	4-wire temperature probe type (4W)
	RTD:	2-wire temperature probe type (2W)
	*RST:	RTD

[SENSe:]TEMPerature:TRANsducer:TYPE?

Returns the probe type of the temperature measurement function.

Return values:

<Probe_Type>	FRTD I RTD	
	FRTD:	4-wire temperature probe type (4W)
	RTD:	2-wire temperature probe type (2W)

UNIT:TEMPerature {C I K I F}

Selects the unit of the temperature measurement function.

Parameters:

<Unit>	C:	°C
	K:	Kelvins
	F:	°F

UNIT:TEMPerature?

Selects the unit of the temperature measurement function.

Return values:

<Unit>	C:	°C is activated
	K:	Kelvins is activated
	F:	°F is activated

ADC rate configuration commands

[SENSe:]ADCRate {SLOW I MEDium I FAST}

Selects the ADC rate for the activated measurement function. The reading rate per second of the ADC rate function is depending on the activated measurement function.

Parameters:

<Function>	SLOW I MEDium I FAST	
	*RST:	SLOW

[SENSe:]ADCRate?

Returns the ADC rate for the activated measurement function.

Return values:

<Function>	SLOW I MED I FAST

Miscellaneous

CONFigure?

Returns the actual instrument configuration.

Example:	TEMP, PT100, RTD
Usage:	Query only

[SENSe:]FUNCtion[:ON] <Function>

Selects the measurement function.

Parameters:

<Function>	VOLTage[:DC]	DC V measurement function
	VOLTage:AC	AC V measurement function
	CURRent[:DC]	DC I measurement function
	CURRent:AC	AC I measurement function
	RESistance	2-wire resistance measurement function
	FRESistance	4-wire resistance measurement function
	FREQuency[:VOLTage]	Frequency measurement function in AC V mode
	FREQuency:CURRent	Frequency measurement function in AC I mode
	CONTinuity	Continuity measurement function
	DIODe	Diode measurement function
	SENSor	Sensor measurement function
	*RST:	VOLT[:DC]
Example:	FUNC VOLT:AC	

[SENSe:]FUNCtion[:ON]?
Returns the selected measurement function.

Return values:

<Function>	VOLT	DC V measurement function
	VOLT:AC	AC V measurement function
	CURR	DC I measurement function
	CURR:AC	AC I measurement function
	RES	2-wire resistance measurement function
	FRES	4-wire resistance measurement function
	FREQ	Frequency measurement function in AC V mode
	FREQ:CURR	Frequency measurement function in AC I mode
	CONT	Continuity measurement function
	DIOD	Diode measurement function
	SENS	Sensor measurement function
Example:	FREQ:CURR	

2.5 Mathematic Functions

CALCulate:FUNCtion {NULL | DB | DBM | AVERage | LIMit | POWer}
Sets the calculation function, but does not activate the function. Use the CALC ON command to activate the calculation function.

Parameters:

<Function>	NULL:	Null function
	DB:	dB function
	DBM:	dBm function
	AVERage:	Statistic measurements
	LIMit:	Limit lines
	POWer:	Power display

The following table shows the possible math/measurement function combinations.

Measurement function	Mathematic functions					
	Stats (AVER)	Limits	NULL	dB	dBm	Power
DC V	●	●	●	●	●	● (only with 2nd function DC I)
AC V	●	●	●	●	●	–
DC I	●	●	●	●	●	● (only with 2nd function DC I)
AC I	●	●	●	●	●	–
Ω	●	●	●	–	–	–
CAP	●	●	●	–	–	–
SENSOR	●	–	●	–	–	–
▶| •)))	–	–	–	–	–	–

Tab. 2.1: Measurement functions with possible math options

CALCulate:FUNCtion?
Returns the calculation function.

Return values:

<Function>	NULL:	Null function
	DB:	dB function (available in DC V AC V, DC I, AC I)
	DBM:	dBm function (available in DC V AC V, DC I, AC I)
	AVER:	Statistic measurements
	LIM:	Limit lines
	POW:	DC power value (available in DC V/DC I or DC I/DC V mode)

CALCulate[:STATe] {OFF | ON}
Turns with the CALC FUNC command selected calculation function on or off.

Parameters:

<State>	ON:	The selected calculation function is activated.
	OFF:	The selected calculation function is disabled.

CALCulate[:STATe]?
Returns the state of the CALC FUNC command selected calculation function.

Return values:

<State>	1 | 0	
	1	ON - Calculation function is activated.
	0	OFF - Calculation function is disabled.

CALCulate:POWer?
Returns the DC power value. To activate the power display function use the CALC:FUNC POW resp. the CALC ON command.

Return values:

<Value> If the HMC8012 is not configured in DCI/DCV or DCV/DCI mode „-1“ will be returned.

CALCulate:LIMit:LOWer {<Value> | MINimum | MAXimum}
Sets the low limit value of the limit line function. The possible maximum and minimum low limit value is depending on the activated measurement function. To activate the limit lines use the CALC:FUNC LIM resp. the CALC ON command

Parameters:

<Value>	Depending on the activated measurement function

CALCulate:LIMit:LOWer? {MINimum | MAXimum}
Retuns the low limit value of the limit line function.

Return values:

<State>	e.g. -7.500E+02

CALCulate:LIMit:UPPer {<Value> | MINimum | MAXimum}
Sets the high limit value of the limit line function. The possible maximum and minimum high limit value is depending on the activated measurement function. To activate the limit lines use the CALC:FUNC LIM resp. the CALC ON command

Parameters:

<Value>	Depending on the activated measurement function

CALCulate:LIMit:UPPer? {MINimum | MAXimum}
Retuns the high limit value of the limit line function.

Return values:

<Value>	e.g. -7.500E+02

CALCulate:DB:REFerence {<Value> | MINimum | MAXimum}
Stores a relative value which is used for the dB function. The possible dB reference value is depending on the activated measurement function. To activate the dB function use the CALC:FUNC DB resp. the CALC ON command

Parameters:

<Value>	Depending on the activated measurement function

CALCulate:DB:REFerence? {MINimum | MAXimum}
Retuns the dB reference value.

Return values:

<Value>	e.g. 1.0E-06

CALCulate:DBM:REFerence {<Value> | MINimum | MAXimum}
Sets the user defined dBm reference resistance value. To activate the dBm function use the CALC:FUNC DBM resp. the CALC ON command.

Parameters:

<Value>	1Ω to 65,5kΩ (adjustable in 1Ω steps) MIN: 1Ω MAX: 65,5kΩ

CALCulate:DBM:REFerence? {MINimum | MAXimum}

Returns the user defined dBm reference resistance value.

Return values:

<Value>	e.g. 6.000E+02

CALCulate:NULL:OFFSet {<Value> | MINimum | MAXimum}

Sets the maximum null value depending on the activated measurement function.

Parameters:

<Value>	Depending on the activated measurement function

CALCulate:NULL:OFFSet? {MINimum | MAXimum}

Returns the maximum null value depending on the activated measurement function.

Return values:

<Value>	e.g. 1.00E+01

CALCulate:AVERage:AVERage?

Returns the mean value of the statistic function depending on the activated measurement function. To activate the statistic function use the CALC:FUNC AVER resp. the CALC ON command.

Return values:

<Value>	e.g. 1.82852E-07
Usage:	Query only

CALCulate:AVERage:CLEar

Resets all statistic function values .

Usage:	Setting only

CALCulate:AVERage:COUNt?

Returns the number of statistic measurement counts. You can read the statistical values at any time.

Return values:

<Value>	e.g. 1.000E+02 (Count = 100)
Usage:	Query only

CALCulate:AVERage:MAXimum?

Returns the maximum value (Max) of the statistic function depending on the activated measurement function. To activate the statistic function use the CALC:FUNC AVER resp. the CALC ON command.

Return values:

<Value>	e.g. 1.55606E-07
Usage:	Query only

CALCulate:AVERage:MINimum?
Returns the minimum value (Min) of the statistic function depending on the activated measurement function. To activate the statistic function use the CALC:FUNC AVER resp. the CALC ON command.

Return values:	
<Value>	e.g. 2.24768E-07
Usage:	Query only

CALCulate:AVERage:PTPeak?
Returns the peak to peak value (Pk to Pk) of the statistic function depending on the activated measurement function. To activate the statistic function use the CALC:FUNC AVER resp. the CALC ON command.

Return values:	
<Value>	e.g. 6.91621E-08
Usage:	Query only

CALCulate:AVERage:SDEViation?
Returns the standard deviation value (StdDev) of the statistic function depending on the activated measurement function. To activate the statistic function use the CALC:FUNC AVER resp. the CALC ON command.

Return values:	
<Value>	e.g. 1.50020E-08
Usage:	Query only

2.6 Data and File Management

DATA:LOG[:STATe] {0 | 1 | OFF | ON}

Turns the data logging function on or off.

Parameters:

<State>	ON / 1:	Data logging function is activated.
	OFF / 0:	Data logging function is disabled.

DATA:LOG[:STATe]?

Returns the state of the data logging function.

Return values:

<State>	1 I 0	
	1	ON - Data logging function is activated.
	0	OFF - Data logging function is disabled.

DATA:LOG:FNAMe {<"File_Name">},[{INT | EXT | DEF}]

Defines the file name and storage location for the logging function.

Parameters:

<File_Name>		e.g. "Test01.CSV",INT
	INT:	Internal memory
	EXT:	USB stick
	DEF:	Internal memory

DATA:LOG:FNAMe?

Returns the file name and storage location for the logging function.

Return values:

<File_Name>		e.g. "/INT/DATA/Test01.CSV"
	INT:	Internal memory
	EXT:	USB stick
	DEF:	Internal memory

DATA:LOG:FORMat {CSV | TXT}

Defines the data logging file format.

Parameters:

<Format>	CSV	Comma separated values
	TXT	Text file
	*RST:	CSV

DATA:LOG:FORMat?

Returns the data logging file format.

Return values:

<Format>	CSV I TXT

DATA:LOG:MODE {UNLimited | COUNt | TIME}
Selects the data logging mode.

Parameters:

<Mode>	UNLimited	Infinite data capture
	COUNt	Number of measurement values to be captured
	TIME	Duration of the measurement values capture
	*RST:	UNL

DATA:LOG:MODE?
Returns the data logging mode.

Return values:

<Mode>	UNL	Infinite data capture
	COUN	Number of measurement values to be captured
	TIME	Duration of the measurement values capture

DATA:LOG:TIME <time in seconds>
Sets the data logging time.

Parameters:

<Time>	Internal:	50000h max.
	External:	Defined by USB stick capacity

DATA:LOG:TIME?
Returns the duration of the measurement values capture.

Return values:

<Time>	e.g. 5.00000E+04

DATA:LOG:COUNt <no of samples>
Sets the number of measurement values to be captured.

Parameters:

<Samples>	Internal:	50000 max.
	External:	Defined by USB stick capacity

DATA:LOG:COUNt?
Returns the number of measurement values to be captured.

Return values:

<Time>	e.g. 1.00E+01

DATA:LOG:INTerval <interval in seconds>
Selects a logging measurement interval. The measurement interval describes the time between the recorded measurements.

Parameters:

<Interval>	3600s max.
	*RST: 0s

DATA:LOG:INTerval?
Returns the selected logging measurement time.

Return values:
<Time> e.g. 1.0E+00

DATA:DATA? {<"File_Name">},[{INT| EXT | DEF}]
Returns the logging file data values of the selected storage location and file name. If no logging file is found, the message „No Logging Files found" is displayed. If no storage location is selected, the instrument queries the internal memory. Please notice that the logging function has to be activated, if you want to use the manual or single trigger mode (trigger via TRIG button). Without activating the logging function in single or manual trigger mode, the instrument is not able to save a logging file internally or on the USB stick.

Return values:
<File_Name> e.g. "LOG0029.CSV"

INT: Internal memory
EXT: USB stick
DEF: Internal memory

Example: External logging file (USB stick) of single trigger, count = 5
DATA:DATA? „LOG0029.CSV",EXT

```
# HAMEG  -Log-File ;
# Date: 2013-05-23 ;
# Start Time:, 15:09:32;
# Stop Time:, 15:10:03 ;
# Settings: ;
#‹9› ADC Rate : Fast;
#‹9› AC Filter : ------;
#‹9› Input Imp. : 10M;
DCV[V],DCI[A],Flag;
12.891854, 0.982340 ;
12.889381, 0.982539 ;
12.909376, 0.982238 ;
12.875769, 0.982418 ;
12.844559, 0.982506 ;
12.863085, 0.982463 ;
```

Usage: Query only

DATA:DELete {<"File_Name">},[{INT | EXT | DEF}]
Deletes the logging file data values of the selected storage location and file name. If no storage location is selected, the instrument uses the internal memory. Please notice that the logging function has to be activated, if you want to use the manual or single trigger mode (trigger via TRIG button). Without activating the logging function in single or manual trigger mode, the instrument is not able to save a logging file internally or on the USB stick.

Parameters:
<File_Name> e.g. "LOG0029.CSV"

INT: Internal memory
EXT: USB stick
DEF: Internal memory

Example: DATA:DEL „LOG0029.CSV",EXT

DATA:POINts? {<“File_Name“>},[{INT | EXT | DEF}]

Returns the number of log file values of the selected storage location and file name. If no storage location is selected, the instrument queries the internal memory. Please notice that the logging function has to be activated, if you want to use the manual or single trigger mode (trigger via TRIG button). Without activating the logging function in single or manual trigger mode, the instrument is not able to save a logging file internally or on the USB stick.

Return values:

<Value>	INT:	Max. 50,000 (internal memory)
	EXT:	Max. memory size of the USB stick
	DEF:	50,000 (internal memory)

Example: External logging file (USB stick) of single trigger, count = 5
DATA:POIN? „LOG0029.CSV“,EXT

Query: 5

DATA:LIST? [{INT | EXT | DEF}]

Returns all saved logging files of the selected storage location. If no storage location is selected, the instrument queries the internal memory. Please notice that the logging function has to be activated, if you want to use the manual or single trigger mode (trigger via TRIG button). Without activating the logging function in single or manual trigger mode, the instrument is not able to save a logging file internally or on the USB stick. If you store the logging file on the USB stick, the query returns all files depending on the storage format (CSV or TXT files).

Return values:

<List>	INT:	Internal memory
	EXT:	USB stick
	DEF:	Internal memory

Example: DATA:LIST? EXT

Query: „LOG0001.CSV“, “LOG0002.CSV“,“LOG0003.CSV“

HCOPy:DATA?

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

Usage: Query only

HCOPy:FORMat { BMP | PNG }

Selects the data format of the screenshot.

Parameters:

<Format> BMP I PNG

BMP:	Windows Bitmap Format
PNG:	Portable Network Graphic

*RST: BMP

HCOPy:FORMat?

Returns the current setting of the screenshot format.

Return values:

<Format>	BMP I PNG	
	BMP:	Windows Bitmap Format
	PNG:	Portable Network Graphic

HCOPy:SIZE:X?

Returns the horizontal expansion of the screenshots.

Usage: Query only

HCOPy:SIZE:Y?

Returns the vertical expansion of the screenshots.

Usage: Query only

*SAV {0|1|2|3|4}

Stores the current instrument state in the specified storage location. Any state previously stored in the same location is overwritten (no error is generated).

*RCL {0|1|2|3|4}

Recalls the current instrument state of the specified storage location.

2.7 Status Reporting

2.7.1 STATus:OPERation Register

The commands of the `STATus:OPERation` subsystem control the status reporting structures of the STATus:OPERation register:

See also:

– chapter 1.6.1, „Structure of a SCPI Status Register“, on page 14
– „STATus:OPERation Register“, on page 17
– Diagram on page 20

The following commands are available:

STATus:OPERation:CONDition?
Returns the of the CONDition part of the operational status register.

Return values:

<Condition>	Condition bits in decimal representation.	
	Range:	1 to 65535
	Increment:	1

Usage: Query only

STATus:OPERation:ENABle <Enable_Value>

Parameters:

<Enable_Value>	Range:	1 to 65535
	Increment:	1

STATus:OPERation:ENABle?
Enables the bits in the enable register for the Standard Operation Register group.

STATus:OPERation[:EVENt]?

Return values:

<Event>	Range:	1 to 65535
	Increment:	1

Usage: Query only

2.7.2 STATus:QUEStionable Registers

The commands of the STATus:QUEStionable subsystem control the status reporting structures of the STATus:QUEStionable registers:

See also:

– chapter 1.6.1, „Structure of a SCPI Status Register“, on page 14
– „STATus:QUEStionable Register“, on page 18
– Diagram on page 20

The following commands are available:

STATus:PRESet

Resets all bits of the STATUS:QUESTIONALBLE and Standard Operation enable register.

Usage: Event

STATus:QUEStionable:CONDition?

Returns the contents of the CONDition part of the status register to check for questionable instrument or measurement states. Reading the CONDition registers does not delete the contents.

Return values:

<Condition>	Condition bits in decimal representation Range: 1 to 65535 Increment: 1
Usage:	Query only

STATus:QUEStionable:ENABle <Enable_Value>

Sets the enable mask that allows true conditions in the EVENt part to be reported in the summary bit. If a bit in the enable part is set to 1 and its associated event bit transitions to true, a positive transition occurs in the summary bit and is reported to the next higher level.

Parameters:

<Enable_Value>	Bit mask in decimal representation Range: 1 to 65535 Increment: 1

STATus:QUEStionable:ENABle?

Reads the enable register and returns a decimal value which corresponds to the binary-weighted sum.

STATus:QUEStionable[:EVENt]?

Returns the contents of the EVENt part of the status register to check whether an event has occurred since the last reading. Reading an EVENt register deletes its contents.

Return values:

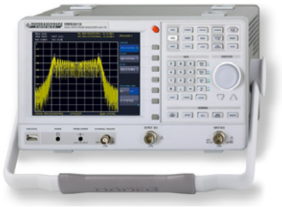
<Event>	Event bits in decimal representation	
	Range:	1 to 65535
	Increment:	1

Usage: Query only

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: January 2014
Subject to change without notice

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