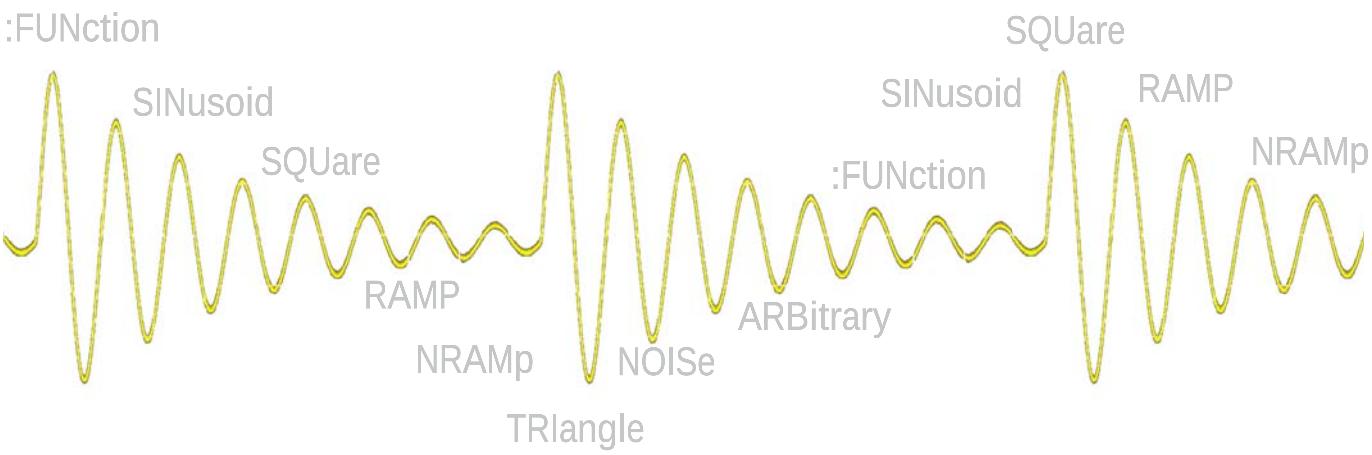
SCPI-Commands HMF series



Content

Introduction

The HMF series is basically supplied with an USB/RS-232 interface. The respective drivers are available on the enclosed Product CD or can be downloaded at http://www.hameg.com. To establish a basic communication a serial cable (1:1) as well as a terminal program like Windows HyperTerminal is required. The Windows HyperTerminal program is part of any Windows operating system. A detailed instruction how to setup a basic communication using HyperTerminal is available at the HAMEG Knowledge Base at http://www.hameg.com/hyperterminal.

If the instrument is being addressed via the interface (remote control), the LED of the Remote button will light up white. Press the Remote button in order to return to local control. This will not function if the instrument's local control is locked out, in this case the instrument can not be operated via the front panel.

Program Commands

A tree structure is used for the programming of the instrument. The program commands contain all commands needed for the control of the instrument. Paths are given following the rules of the SCPI syntax. SCPI commands must be sent line by line, i.e. each line has to be terminated by a Line feed. It is not allowed to concatenate commands by a semicolon. Commands which cause an answer from the instrument are called a query. This applies as well to queries of the instrument status, parameters or limits. The commands which ask for an immediate answer use a question mark (?). Key words in parentheses ( ) may left out. The various levels are marked with a colon (:). This indicates that the following characters are elements of the level ROOT. SCPI commands may be sent long or short. Long commands use up to 12 characters (example: MEASure). Short commands consist of the first 4 characters of the long command (example: MEAS). If the 4th character happens to be a vowel and if these 4 characters are not the long command, the short command will only consist of 3 characters (example: long = ARBitrary, short = ARB).

The upper key characters are mandatory and they constitute the short commands, the lower key characters are optional. It is important to adhere to the prescribed formats of the SCPI commands, except the short and long formats of the commands all other formats are illegal. Upper or lower key characters need not be observed. In this manual upper key characters are used to indicate the short form commands, the long format is indicated by the additional characters in lower ley.

NOTE
In order to prevent communication errors we recommend not to use concatenation of commands and to terminate each command by a line feed.

Also make sure that the instrument generates readable data which can be used by the listener (computer).

Sources of errors could be:

- instruments are switched off, cable loose, interface not activated
- wrong instrument address
- false or incomplete strings of commands
- wrong measuring range

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

The HMF2525/2550 uses SCPI (= Standard Commands for Programmable Instruments) based commands for remote control.

Remote control is possible via the built-in dual interface USB/RS-232 (options: Ethernet/USB, IEEE-488). This allow access to nearly all functions which are available on the front panel.

SCPI commands

Symbol	Description
< >	variable, predefined element
=	equality, identical to
|	or
( )	group of elements, commentary
[ ]	optional elements
{ }	multitude of elements
' '	example

Overview of SCPI Commands

APPLY Commands

```
APPLy
       :SINusoid [<frequency> [,<amplitude>] [,<offset>]]             with release 1.2
       :SQUare [<frequency> [,<amplitude>] [,<offset>]]
       :RAMP [<frequency> [,<amplitude>] [,<offset>]]
       :PULSe [<frequency> [,<amplitude>] [,<offset>]]
       :ARB [<frequency> [,<amplitude>] [,<offset>]]
APPLy?
```

Configuration of the Output

```
FUNCtion {SINusoid|SQUare|RAMP|PULSe|ARBitrary}
FUNCtion?

FREQuency {<frequency>|MINimum|MAXimum}
FREQuency? [MINimum|MAXimum]

PERiode {<period in sec.>|MINimum|MAXimum}
PERiode? [MINimum|MAXimum]

VOLTage {<amplitude>|MINimum|MAXimum}
VOLTage? [MINimum|MAXimum]

VOLTage
       :HIGH {<voltage>|MINimum|MAXimum}
       :HIGH? [MINimum|MAXimum]
       :LOW {<voltage>|MINimum|MAXimum}
       :LOW? [MINimum|MAXimum]

VOLTage:OFFSet {<offset>|MINimum|MAXimum}
VOLTage:OFFSet? [MINimum|MAXimum]

FUNCtion:SQUare:DCYCle {<percent>|MINimum|MAXimum}
FUNCtion:SQUare:DCYCle? [MINimum|MAXimum]

FUNCtion:RAMP:SYMMetry {<percent>|MINimum|MAXimum}
FUNCtion:RAMP:SYMMetry? [MINimum|MAXimum]

OUTPut {OFF|ON}
OUTPut?

OUTPut:LOAD {TERMinated|INFinity}
OUTPut:LOAD?

OUTPut:POLarity {NORMal|INVerted}
OUTPut:POLarity?
```

Pulse configuration

```
FUNCtion:PULSe
       :WIDTh:HIGH {<seconds>|MINimum|MAXimum}
       :WIDTh:HIGH? [MINimum|MAXimum]

       :WIDTh:LOW {<seconds>|MINimum|MAXimum}
       :WIDTh:LOW? [MINimum|MAXimum]

       :DCYCle {<percent>|MINimum|MAXimum}
       :DCYCle? [MINimum|MAXimum]

       :ETIMe {<seconds>|MINimum|MAXimum}
       :ETIMe? [MINimum|MAXimum]
```

Modulation commands

AM-Commands

```
AM:INTernal
       :FUNCtion {SINusoid|SQUare|RAMP|NRAMp|TRIangle|NOISe|ARBitrary*}
       :FUNCtion?                                        *with release 1.2

AM:INTernal
       :FREQuency {<frequency>|MINimum|MAXimum}
       :FREQuency? [MINimum|MAXimum]

AM:DEPTh {<modulation in percent>|MINimum|MAXimum}
AM:DEPTh? [MINimum|MAXimum]

AM:SOURce {INTernal|EXTernal}
AM:SOURce?

AM:STATe {OFF|ON}
AM:STATe?
```

FM-Commands

```
FM:INTernal
       :FUNCtion {SINusoid|SQUare|RAMP|NRAMp|TRIangle|NOISe|ARBitrary*}
       :FUNCtion?                                        *with release 1.2

FM:INTernal
       :FREQuency {<frequency>|MINimum|MAXimum}
       :FREQuency? [MINimum|MAXimum]

FM:DEViation {<peak-frequency in Hz>|MINimum|MAXimum}
FM:DEViation? [MINimum|MAXimum]

FM:SOURce {INTernal|EXTernal}
FM:SOURce?

FM:STATe {OFF|ON}
FM:STATe?
```

PM-Commands

```
PM:INTernal
      :FUNCtion {SINusoid|SQUare|RAMP|NRAMp|TRIangle|NOISe|ARBitrary*}
      :FUNCtion?                                         *with release 1.2

PM:INTernal
      :FREQuency {<frequency>|MINimum|MAXimum}
      :FREQuency? [MINimum|MAXimum]

PM:DEViation {<phase in degree>|MINimum|MAXimum}
PM:DEViation? [MINimum|MAXimum]

PM:SOURce {INTernal|EXTernal}
PM:SOURce?

PM:STATe {OFF|ON}
PM:STATe?
```

FSK-Commands

```
FSKey:FREQuency {<frequency>|MINimum|MAXimum}
FSKey:FREQuency? [MINimum|MAXimum]

FSKey:INTernal:RATE {<rate in Hz>|MINimum|MAXimum}
FSKey:INTernal:RATE? [MINimum|MAXimum]

FSKey:SOURce {INTernal|EXTernal}
FSKey:SOURce?

FSKey:STATe {OFF|ON}
FSKey:STATe?
```

Wobble commands

```
FREQuency
      :STARt {<frequency>|MINimum|MAXimum}
      :STARt? [MINimum|MAXimum]

      :STOP {<frequency>|MINimum|MAXimum}
      :STOP? [MINimum|MAXimum]

FREQuency
      :CENTer {<frequency>|MINimum|MAXimum}
      :CENTer? [MINimum|MAXimum]

      :SPAN {<frequency>|MINimum|MAXimum}
      :SPAN? [MINimum|MAXimum]

SWEep
      :SPACing {LINear|LOGarithmic}
      :SPACing?

      :TIME {<seconds>|MINimum|MAXimum}
      :TIME? [MINimum|MAXimum]

SWEep:STATe {OFF|ON}
SWEep:STATe?
```

```
MARKer:FREQuency {<frequency>|MINimum|MAXimum}
MARKER:FREQuency? [MINimum|MAXimum]
MARKer {OFF|ON}
MARKer?
```

Burst Commands

```
BURSt:MODE {TRIGgered|GATed}
BURSt:MODE?

BURSt:GATE:POLarity {NORMal|INVerted}
BURSt:GATE:POLarity?

BURSt:NCYCles {<number of cycles>|INFinity|MINimum|MAXimum}
BURSt:NCYCles? [MINimum|MAXimum]

BURSt:INTernal:PERiod {<seconds>|MINimum|MAXimum}
BURSt:INTernal:PERiod? [MINimum|MAXimum]

BURSt:PHASe {<angle>|MINimum|MAXimum}
BURSt:PHASe? [MINimum|MAXimum]

BURSt:STATe {OFF|ON}
BURSt:STATe?
```

Arbitrary commands

```
DATA {<value>,<value>|<binary block>}

DATA:COPY <Arb-name>

DATA:DELete {<Arb-name>}
DATA:DELete:ALL

DATA:SWAP {NORMal|SWAPped}
DATA:SWAP?

FUNCtion:ARBitrary {<Arb-name>}
FUNCtion:ARBitrary?
```

Trigger commands

```
TRIGger
*TRG

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

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

TRIGger {OFF|ON}
TRIGger?
```

Save and recall of instrument settings

```
*SAV {0|1|2|3|4} Power off settings are stored at instrument setting 0
*RCL {0|1|2|3|4} Settings 1, 2, 3 und 4 are user defined settings.
```

with release 1.2

System and register commands

```
*IDN?
*RST
*TST?
*OPC
*OPC?
*WAI
*LRN?
*STB?
*SRE <activation value>
*SRE?
*ESR?
*ESE <activation value>
*ESE?
*CLS
*PSC {0|1}
*PSC?

STATus
       :QUEStionable:CONDition?
       :QUEStionable[:EVENt]?
       :QUEStionable:ENABle <activation value>
       :QUEStionable:ENABle?

SYSTem:ERRor?

STATus:PRESet

SYSTem
       :BEEPer:STATe {OFF|ON}
       :BEEPer:STATe?
```

with release 1.2

Detailed description of SCPI commands

1 Function and output commands

APPLy

:SINusoid [<*frequency*> [,<*amplitude*>] [,<*offset*>]] with release 1.2
:SQUare [<*frequency*> [,<*amplitude*>] [,<*offset*>]]
:RAMP [<*frequency*> [,<*amplitude*>] [,<*offset*>]]
:PULSe [<*frequency*> [,<*amplitude*>] [,<*offset*>]]
:ARB [<*frequency*> [,<*amplitude*>] [,<*offset*>]]

APPLy?

The APPL command provides the most straightforward method to program the function generator over the remote interface. You can select the function, frequency, amplitude, and offset all in one command as shown in the syntax statement above.

Because of the use of optional parameters in the APPLy commands (enclosed in square brackets), you must specify frequency to use the amplitude parameter, and you must specify both frequency and amplitude to use the offset parameter. For example, the following command string is valid (frequency and amplitude are specified but offset is omitted and therefore uses a default value).

APPL:SIN 4.0E+2, 5.0

However, you cannot specify an amplitude or offset without specifying a frequency.
You can substitute "MIN" or "MAX" in place of specific values for the frequency, amplitude, and offset parameters.
For example, the following statement outputs a 5 V_{pp} sine wave at 50 MHz (the maximum frequency of the HMF2550 for sine) with a -1.2 Volt offset.

APPL:SIN MAX, 5.0, -1.2

FUNCtion { **SINus** | **SQUare** | **RAMP** | **PULSe** | **NOISe** | **ARBitrary** }
FUNCtion?

Select the output function. The selected waveform uses the previously selected frequency, amplitude, and offset voltage settings.
The FUNC? query returns the current function (SIN, SQU, RAMP, PULS, NOIS, ARB).

OUTPut { **OFF** | **ON** }
OUTPut?

Activates or deactivates the output.
The OUTP? query returns the current status (ON, OFF).

OUTPut:LOAD { **TERMinated** | **INFinity** }
OUTPut:LOAD?

Select the output load. Terminated is defined as 50Ω, infinity is defined as open circuit (without any load).
The LOAD? query returns the current status (terminated, infinity).

OUTPut:POLarity { **NORMal** | **INVerted** } *only valid for Pulse and Arbitrary function*
OUTPut:POLarity?

Inverts the current output waveform (pulse or arbitrary only).
The POL? query returns the current status (normal, inverted).

1.1 Sine function

FREQuency { <***frequency***> | **MINimum** | **MAXimum** }
FREQuency?

Set the output frequency. MIN selects the lowest frequency allowed for the sine function and MAX selects the highest frequency allowed. The FREQ? query returns the frequency setting in Hertz for the sine function.

PERiod { <***period***> | **MINimum** | **MAXimum** }
PERiod?

Set the output period. MIN selects the lowest period allowed for the sine function and MAX selects the highest period allowed. The PER? query returns the period setting in sec. for the sine function.

VOLTage { <***amplitude***> | **MINimum** | **MAXimum** }
VOLTage?

Set the output amplitude. MIN selects the lowest amplitude allowed for the sine function and MAX selects the highest amplitude allowed. The VOLT? query returns the amplitude setting in Volt for the sine function.

VOLTage
:HIGH { <***voltage***> | **MINimum** | **MAXimum** }
:HIGH? [**MINimum** | **MAXimum**]
:LOW { <***voltage***> | **MINimum** | **MAXimum** }
:LOW? [**MINimum** | **MAXimum**]

Set the upper and the lower level of the sine function amplitude. MIN selects the lowest amplitude allowed for the sine function and MAX selects the highest amplitude allowed.
This command offers one possibility to specify a sine wave including a offset.
The HIGH / LOW? query returns the current value in Volt.

VOLTage
:OFFSet { <***offset***> | **MINimum** | **MAXimum** }
:OFFSet?

Set the output offset. MIN selects the lowest offset allowed for the sine function and MAX selects the highest offset allowed. The OFFS? query returns the current offset setting in Volt for the sine function.

1.2 Square function

FREQuency { <***frequency***> | **MINimum** | **MAXimum** }
FREQuency?

Set the output frequency. MIN selects the lowest frequency allowed for the square function and MAX selects the highest frequency allowed. The FREQ? query returns the frequency setting in Hertz for the square function.

PERiod { <***period***> | **MINimum** | **MAXimum** }
PERiod?

Set the output period. MIN selects the lowest period allowed for the square function and MAX selects the highest period allowed. The PER? query returns the period setting in sec. for the square function.

VOLTage { <***amplitude***> | **MINimum** | **MAXimum** }
VOLTage?

Set the output amplitude. MIN selects the lowest amplitude allowed for the square function and MAX selects the highest amplitude allowed. The VOLT? query returns the amplitude setting in Volt for the square function.

VOLTage

:HIGH { <***voltage***> | **MINimum** | **MAXimum** }
:HIGH? [**MINimum** | **MAXimum**]
:LOW { <***voltage***> | **MINimum** | **MAXimum** }
:LOW? [**MINimum** | **MAXimum**]

Set the upper and the lower level of the square function amplitude. MIN selects the lowest amplitude allowed for the square function and MAX selects the highest amplitude allowed.
This command offers one possibility to specify a square function including a offset.
The HIGH / LOW? query returns the current value in Volt.

VOLTage

:OFFSet { <***offset***> | **MINimum** | **MAXimum** }
:OFFSet?

Set the output offset. MIN selects the lowest offset allowed for the square function and MAX selects the highest offset allowed. The OFFS? query returns the offset setting in V for the square function.

FUNCtion:SQUare

:DCYCle { <***duty cycle in %***> | **MINimum** | **MAXimum** }
:DCYCle?

Set the output duty cycle. MIN selects the lowest duty cycle allowed for the square function and MAX selects the highest duty cycle allowed. The DCYC? query returns the amplitude setting in % for the square function.

1.3 Ramp function

FREQuency { <***frequency***> | **MINimum** | **MAXimum** }
FREQuency?

Set the output frequency. MIN selects the lowest frequency allowed for the ramp function and MAX selects the highest frequency allowed. The FREQ? query returns the frequency setting in Hertz for the ramp function.

PERiod { <***period***> | **MINimum** | **MAXimum** }
PERiod?

Set the output period. MIN selects the lowest period allowed for the ramp function and MAX selects the highest period allowed. The PER? query returns the period setting in sec. for the ramp function.

VOLTage { <***amplitude***> | **MINimum** | **MAXimum** }
VOLTage?

Set the output amplitude. MIN selects the lowest amplitude allowed for the ramp function and MAX selects the highest amplitude allowed. The VOLT? query returns the amplitude setting in Volt for the ramp function.

VOLTage

:HIGH { <***voltage***> | **MINimum** | **MAXimum** }
:HIGH? [**MINimum** | **MAXimum**]
:LOW { <***voltage***> | **MINimum** | **MAXimum** }
:LOW? [**MINimum** | **MAXimum**]

Set the upper and the lower level of the ramp function amplitude. MIN selects the lowest amplitude allowed for the ramp function and MAX selects the highest amplitude allowed.
This command offers one possibility to specify a ramp function including a offset.
The HIGH / LOW? query returns the current value in Volt.

VOLtage

:OFFSet { <***offset***> | **MINimum** | **MAXimum** }
:OFFSet?

Set the output offset. MIN selects the lowest offset allowed for the ramp function and MAX selects the highest offset allowed. The OFFS? query returns the offset setting in Volt for the ramp function.

FUNCtion:RAMP

:SYMmetry { <***value in %*** > | **MINimum** | **MAXimum** }

:SYMmetry?

Set the output symmetry. MIN selects the lowest symmetry allowed for the ramp function and MAX selects the highest symmetry allowed. The SYM? query returns the symmetry setting in % for the square function.

1.4 Pulse function

FREQuency { <***frequency***> | **MINimum** | **MAXimum** }

FREQuency?

Set the output frequency. MIN selects the lowest frequency allowed for the pulse function and MAX selects the highest frequency allowed. The RAMP? returns the frequency setting in Hertz for the pulse function.

PERiod { <***period***> | **MINimum** | **MAXimum** }

PERiod?

Set the output period. MIN selects the lowest period allowed for the pulse function and MAX selects the highest period allowed. The PER? query returns the period setting in sec. for the pulse function.

:VOLTage { <***amplitude***> | **MINimum** | **MAXimum** }

:VOLTage?

Set the output amplitude. MIN selects the lowest amplitude allowed for the pulse function and MAX selects the highest amplitude allowed. The VOLT? query returns the amplitude setting in Volt for the pulse function.

VOLTage

:HIGH { <***voltage***> | **MINimum** | **MAXimum** }

:HIGH? [**MINimum** | **MAXimum**]

:LOW { <***voltage***> | **MINimum** | **MAXimum** }

:LOW? [**MINimum** | **MAXimum**]

Set the upper and the lower level of the pulse function amplitude. MIN selects the lowest amplitude allowed for the pulse function and MAX selects the highest amplitude allowed. This command offers one possibility to specify a pulse function including a offset. The HIGH / LOW? query returns the current value in Volt.

VOLTage

:OFFSet { <***offset***> | **MINimum** | **MAXimum** }

:OFFSet?

Set the output offset. MIN selects the lowest offset allowed for the pulse function and MAX selects the highest offset allowed. The OFFS? query returns the offset setting in Volt for the pulse function.

FUNction:PULSe

WIDTh:HIGH { <***seconds***> | **MINimum** | **MAXimum** }

WIDTh:HIGH? { **MINimum** | **MAXimum** }

WIDTh:LOW { <***seconds***> | **MINimum** | **MAXimum** }

WIDTh:LOW? { **MINimum** | **MAXimum** }

Set the output width for the upper or lower part of the pulse function. MIN selects the lowest width allowed for the pulse function and MAX selects the highest width allowed. The WIDT? query returns the respective width setting in sec. for the upper or lower part of the pulse function.

FUNction:PULSe

:DCYCle { <***duty cycle in %*** > | **MINimum** | **MAXimum** }

:DCYCle?

Set the output duty cycle. MIN selects the lowest duty cycle allowed for the pulse function and MAX selects the highest amplitude allowed. The DCYC? query returns the duty cycle setting in % for the pulse function.

FUNction:PULSe

:EDGEtime { <***seconds***> | (**MINimum** | **MAXimum** }

:EDGEtime? { **MINimum** | **MAXimum** }

Set the output edge times. MIN selects the lowest edge time allowed for the pulse function and MAX selects the highest edge time allowed. The EDGE? query returns the current edge time setting in sec. for the pulse function.

2 Modulations commands

2.1 AM modulation

AM:INTernal

:FUNCtion { **SINusoid** | **SQUare** | **RAMP** | **NRAMp** | **TRIangle** | **NOISe** | **ARBitrary*** }

:FUNCtion? **with release 1.2*

Set the shape of the AM modulation signal. All above mentioned signal shapes can be selected.
The FUNC? query returns the current setting of the modulation shape.

AM:INTernal

:FREQuency { <***frequency***> | **MINimum** | **MAXimum** }

:FREQuency? [**MINimum** | **MAXimum**]

Set the frequency of the AM modulation signal. MIN selects the lowest frequency allowed for the AM modulation signal and MAX selects the highest frequency allowed. The FREQ? query returns the current frequency setting for the modulation signal in Hertz.

AM:DEPTh { <***modulation in %***> | **MINimum** | **MAXimum** }

AM:DEPTh? [**MINimum** | **MAXimum**]

Set the AM modulation depth in %. MIN selects the lowest modulation depth for the AM modulation signal and MAX selects the highest modulation depth allowed.
The DEPT? query returns the current setting in %.

AM:SOURce { **INTernal** | **EXTernal** }

AM:SOURce?

Set the modulation source. INT selects the internal modulation source for the AM modulation and EXT selects a external source as reference for the modulation. This external source needs to be connected to the rear "modulation input" connector.
The SOUR? query returns the current setting of the modulation source (internal, external).

AM:STATe { **OFF** | **ON** }

AM:STATe?

Set the AM modulation status. Activates or deactivates the AM modulation of the main signal.
The STAT? query returns the current status of the AM modulation state (OFF, ON).

2.2 FM modulation

FM:INTernal

:FUNCtion { **SINusoid** | **SQUare** | **RAMP** | **NRAMp** | **TRIangle** | **NOISe** | **ARBitrary*** }

:FUNCtion? **with release 1.2*

Set the shape of the FM modulation signal. All above mentioned signal shapes can be selected.
The FUNC? query returns the current setting of the modulation shape.

FM:INTernal

:FREQuency { <***frequency***> | **MINimum** | **MAXimum** }

:FREQuency? [**MINimum** | **MAXimum**]

Set the frequency of the FM modulation signal. MIN selects the lowest frequency allowed for the FM modulation signal and MAX selects the highest frequency allowed. The FREQ? query returns the current frequency setting for the modulation signal in Hertz.

FM:DEViation { <***peak-frequency in Hz***> | **MINimum** | **MAXimum** }
FM:DEViation? [**MINimum** | **MAXimum**]

Set the peak frequency deviation in Hertz. This value represents the variation in frequency of the modulating waveform from the carrier frequency. MIN selects the lowest frequency allowed for the deviation and MAX selects the highest frequency allowed.

FM:SOURce { **INTernal** | **EXTernal** }
FM:SOURce?

Set the modulation source. INT selects the internal modulation source for the FM modulation and EXT selects a external source as reference for the modulation. This external source needs to be connected to the rear "modulation input" connector.
The SOUR? query returns the current setting of the modulation source (internal, external).

FM:STATe { **OFF** | **ON** }
FM:STATe?

Set the FM modulation status. Activates or deactivates the FM modulation of the main signal.
The STAT? query returns the current status of the AM modulation state (OFF, ON).

2.3 PM modulation

PM:INTernal
:FUNCtion { **SINusoid** | **SQUare** | **RAMP** | **NRAMp** | **TRIangle** | **NOISe** | **ARBitrary*** }
:FUNCtion?

*with release 1.2

Set the shape of the PM modulation signal. All above mentioned signal shapes can be selected.
The FUNC? query returns the current setting of the modulation shape.

PM:INTernal
:FREQuency { <***frequency***> | **MINimum** | **MAXimum** }
:FREQuency? [**MINimum** | **MAXimum**]

Set the frequency of the PM modulation signal. MIN selects the lowest frequency allowed for the PM modulation signal and MAX selects the highest frequency allowed. The FREQ? query returns the current frequency setting for the modulation signal in Hertz.

PM:DEViation { <***phase in degree***> | **MINimum** | **MAXimum** }
PM:DEViation? [**MINimum** | **MAXimum**]

Set the deviation of the phase in degree (-180°...+180°). This value represents the variation in degree of the modulating waveform from the carrier waveform. MIN selects the lowest angle allowed for the deviation and MAX selects the highest angle allowed.

PM:SOURce { **INTernal** | **EXTernal** }
PM:SOURce?

Set the modulation source. INT selects the internal modulation source for the PM modulation and EXT selects a external source as reference for the modulation. This external source needs to be connected to the rear "modulation input" connector.
The SOUR? query returns the current setting of the modulation source (internal, external).

PM:STATe { **OFF** | **ON** }
PM:STATe?

Set the PM modulation status. Activates or deactivates the PM modulation of the main signal.
The STAT? query returns the current status of the PM modulation state (OFF, ON).

2.4 FSK modulation

FSKey:FREQuency { <***frequency***> | **MINimum** | **MAXimum** }
FSKey:FREQuency? [**MINimum** | **MAXimum**]

Set the frequency of the FSK modulation signal. MIN selects the lowest frequency allowed for the FSK modulation signal and MAX selects the highest frequency allowed. The FREQ? query returns the current frequency setting for the modulation signal in Hertz.

FSKey:INTernal

:RATE { <***rate in Hz***> | **MINimum** | **MAXimum** }
:RATE? [**MINimum** | **MAXimum**]

Set the rate (frequency) of the FSK modulation in order to define the timeframe between the frequency shifts. MIN selects the lowest rate allowed for the FSK modulation signal and MAX selects the highest rate allowed.
The RATE? query returns the current rate setting for the modulation signal in Hertz.

FSKey:SOURce { **INTernal** | **EXTernal** }
FSKey:SOURce?

Set the modulation source. INT selects the internal modulation source for the FSK modulation and EXT selects a external source as reference for the modulation. This external source needs to be connected to the rear "modulation input" connector.
The SOUR? query returns the current setting of the modulation source (internal, external).

FSKey:STATe { **OFF** | **ON** }
FSKey:STATe?

Set the FSK modulation status. Activates or deactivates the FSK modulation of the main signal.
The STAT? query returns the current status of the FSK modulation state (OFF, ON).

3 Sweep and Burst commands

3.1 Sweep commands

FREQuency

:STARt { <***frequency***> | **MINimum** | **MAXimum** }
:STARt? [**MINimum** | **MAXimum**]

Set the start frequency of the sweep. MIN selects the lowest frequency allowed for the sweep function and MAX selects the highest frequency allowed. The FREQ? query returns the current start frequency setting for the sweep function in Hertz.

:STOP { <***frequency***> | **MINimum** | **MAXimum** }
:STOP? [**MINimum** | **MAXimum**]

Set the stop frequency of the sweep. MIN selects the lowest stop frequency allowed for the sweep function and MAX selects the highest frequency allowed. The FREQ? query returns the current stop frequency setting for the sweep function in Hertz.

FREQuency

:CENTer { <***frequency***> | **MINimum** | **MAXimum** }
:CENTer? [**MINimum** | **MAXimum**]

Set the center frequency of the sweep. MIN selects the lowest center frequency allowed for the sweep function and MAX selects the highest frequency allowed. The FREQ? query returns the current center frequency setting for the sweep function in Hertz.

:SPAN { <***frequency***> | **MINimum** | **MAXimum** }
:SPAN? [**MINimum** | **MAXimum**]

Set the span frequency of the sweep. MIN selects the lowest span frequency allowed for the sweep function and MAX selects the highest frequency allowed. The FREQ? query returns the current span frequency setting for the sweep function in Hertz.

SWEep

:SPACing { **LINear** | **LOGarithmic** }
:SPACing?

Set the spacing of the sweep to linear or logarithmic.
The SPAC? query returns the current setting of the spacing (linear, logarithmic).

SWEep

:TIME { <***seconds***> | **MINimum** | **MAXimum** }

:TIME? [**MINimum** | **MAXimum**]

Set the time required to sweep from the start frequency to the stop frequency. MIN selects the lowest sweep time allowed for the sweep function and MAX selects the highest sweep time allowed.The TIME? query returns the sweep time in seconds.

SWEep:STATe { **OFF** | **ON** }

SWEep:STATe?

Set the sweep status. Activates or deactivates the sweep function of the signal.
The STAT? query returns the current status of the sweep function state (OFF, ON).

MARKer:FREQuency { <***frequency***> | **MINimum** | **MAXimum** }

MARKER:FREQuency? [**MINimum** | **MAXimum**]

Set the frequency of the marker. MIN selects the lowest marker frequency allowed for the marker function and MAX selects the highest frequency allowed.
The FREQ? query returns the current setting of the marker frequency.

MARKer { **OFF** | **ON** }

MARKer?

Disable or enable the frequency marker. The MARK? query returns the current setting of the marker function (OFF,ON).

3.2 Burst commands

BURSt:MODE { **TRIGgered** | **GATed** }

BURSt:MODE?

Set the BURST mode. In TRIG mode the signal will be started after the pos. or neg. Slope of the trigger due to current trigger settings. In GAT mode the signal will be started with the pos. slope and will be interrupted by the neg. slope of the gate signal.
The MODE? query returns the current setting of the mode function (triggered, gated).

BURSt:GATE:POLarity { **NORMal** | **INVerted** }

BURSt:GATE:POLarity?

Select whether the function generator uses true-high or true-low logic levels for the gated burst mode. The default is NORM (true-high logic).
The POL? query returns the current status of the gate polarity setting (normal, inverted).

BURSt:NCYCles { <***number of cycles***> | **INFinity** | **MINimum** | **MAXimum** }

BURSt:NCYCles? [**MINimum** | **MAXimum**]

Set the number of repetition cycles of the BURST mode. INF sets the number of repetitions to infinite. MIN selects the lowest number of repetitions allowed for the cycle function and MAX selects the highest repetitions allowed.
The NCYC? query returns the current setting of the burst repetition function.

BURSt:INTernal:PERiod { <***seconds***> | **MINimum** | **MAXimum** }

BURSt:INTernal:PERiod? [**MINimum** | **MAXimum**]

Set the internal period time of the BURST mode. MIN selects the lowest period time allowed for the burst function and MAX selects the highest period time allowed.
The PER? query returns the period setting in sec. for the burst function.

BURSt:PHASe { <***angle***> | **MINimum** | **MAXimum** }

BURSt:PHASe? [**MINimum** | **MAXimum**]

Set the phase of the BURST mode in degree. MIN selects the angle allowed for the burst function and MAX selects the highest angle allowed.
The PHAS? query returns the current setting of the phase angle.

BURSt:STATe { **OFF** | **ON** }

BURSt:STATe?

Set the burst state. Activates or deactivates the burst function.
The STAT? query returns the current status of the burst function (OFF, ON).

4 Arbitrary commands

DATA { <*value*>,<*value*> | <*binary block*> }

Use this function to define a arbitrary waveform. The function generator can handle data either as floating point values or as binary values and is able to distinguish between these two formats automatically.

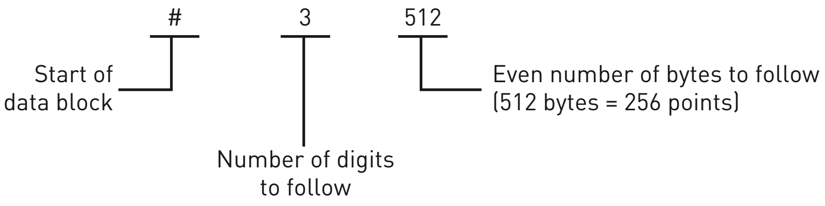
Create floating-point values from -1 to +1.
The values -1 and +1 correspond to the peak values of the waveform (if the offset is 0 volts).
For example, if you set the amplitude to 10 V_{pp} (0 V offset), "+1" corresponds to +5 V and "-1" corresponds to -5 V.

The DATA command will overwrite any previous generated waveform .
Use the DATA:COPY command to copy the waveform to non-volatile memory.

The following example shows how to use the DATA command to create seven arbitrary points using floating-point values:

DATA 1, .67, .33, 0, -.33, -.67, -1

In the binary block format, a block header precedes the waveform data.
The block header has the following format:



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

DATA:COPY <*Arb-name*>

You can output a arbitrary waveform directly from the volatile memory or you can copy the waveform to non-volatile memory using the DATA:COPY command and a unique name for the arbitrary waveform crated. The arbitrary name may contain up to 10 characters.

DATA:DELete { <*Arb-name*> }
DATA:DELete:ALL

Deletes the selected user defined waveform.
DATA:DEL:ALL deletes all stored uder defined waveforms in the function generator.

DATA:SWAP { NORMal | SWAPped }
DATA:SWAP?

Invert a arbitrary waveform. Select whether the function generator provides the user defined waveform or a inverted version of this waveform . The SWAP? query returns the current status of this function (normal, swapped).

FUNCtion:ARBitrary { <*Arb-name*> }
FUNCtion:ARBitrary?

Select one of the built-in arbitrary waveforms or one of user-defined waveforms and load it into the internal memory.
The ARB? Query returns the currently selected arbitrary waveform.

5 Trigger commands

Trigger commands will only effect the wobble and burst mode.

TRIGger
*TRG

These commands are used to activate a internal trigger signal via remote control.

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

Set the trigger source. IMM selects the internal source for the trigger and EXT selects a external source as reference for the trigger. This external source needs to be connected to the front "trigger input" connector (TTL).
The SOUR? query returns the current setting of the trigger source (immediate, external).

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

Select whether the function generator uses the rising edge or falling edge of the trigger signal.
The SLOP? query returns the type of the edge.

TRIGger { **OFF** | **ON** }
TRIGger?

Acitvate or deactivate the trigger function.
The TRIG? query returns the current setting of the trigger (OFF,ON).

6 Save and recall of instrument settings

SAV** { **0** | **1** | **2** | **3** | **4** } ***Power off settings are stored at instrument setting 0
RCL** { **0** | **1** | **2** | **3** | **4** } ***Settings 1, 2, 3 und 4 are user defined settings. *with release 1.2*

Use this command to save or recall your preferred system settings.
The function generator has five storage locations in non-volatile memory to store the instrument settings. The function generator automatically uses location "0" to hold the state of the instrument at power down.

7 Status register and common commands

***CLS**

Clear the event register in all register groups. This commands also clears the error queue and cancels *OPC operation.

***ESR?**

A query of this register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

ESE** < ***activation value>
***ESE?**

Enable bits in the Standard Event Status Register to be reported in the Status Byte. The *ESE? query returns a decimal value which corresponds to the binary-weighted sum of all bits enabled by the *ESE command.

***IDN?**

Read the function generator's identification string which contains four fields separated by commas.

***LRN?**

Query the function generator and return a string of SCPI commands containing the current settings.

***OPC**
***OPC?**

Set the Operation Complete bit in the Standard Event Register after the previous commands have completed.

***RST**

Reset the function generator to its factory default state.

***STB?**

Query the summary (condition) register in this register group.

*SRE <*activation value*>
*SRE?

Enable bits in the Status Byte to generate a Service Request. To enable specific bits, you must write a decimal value which corresponds to the binary-weighted sum of the bits in the register. The *SRE? query returns a decimal value which corresponds to the binary-weighted sum of all bits enabled by the *SRE command.

*TST?

Perform a complete self-test of the function generator. Returns "+0" (PASS) or "+1" (FAIL). If the test fails, one or more error messages will be generated to provide additional information on the failure.
Use the SYST:ERR? command to read the error queue.

*WAI

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

*PSC { 0 | 1 }
*PSC?

Power-On Status Clear. Clear the Standard Event enable register and Status Byte condition register at power on (*PSC 1). When *PSC 0 is in effect, these two registers are not cleared at power on. The default is *PSC 1.
The *PSC? query returns the power-on status clear setting and returns "0" (do not clear at power on) or "1" (clear at power on).

STATus
:QUEStionable:CONDition?

Query the condition register in this group. This is a read-only register and bits are not cleared when you read the register. A query of this register returns a decimal value which corresponds to the binaryweighted sum of all bits set in the register.

:QUEStionable[:EVENt]?

Query the event register in this register group. This is a read-only register. Once a bit is set, it remains set until cleared by this command or *CLS (clear status) command. A query of this register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

:QUEStionable:ENABle <*activation value*>
:QUEStionable:ENABle?

Enable bits in the enable register in this register group. The selected bits are then reported to the Status Byte. A *CLS (clear status) will not clear the enable register but it does clear all bits in the event register. The STATus:PRESet command clears all bits in the enable register.
To enable bits in the enable register, you must write a decimal value which corresponds to the binary-weighted sum of the bits you wish to enable in the register.
The QUES:ENAB? query returns a decimal value which corresponds to the binary-weighted sum of all bits enabled by the STAT:QUES:ENAB command.

STATus:PRESet

Clear all bits in the Questionable Data enable register and the Standard Operation enable register.

SYSTem:ERRor?

Reads one error from the error queue.
After reading of an error it will be erased from the register. The next query will read the next error provided there was more than one.

SYSTem
:BEEPer:STATe { OFF | ON }

with release 1.2

:BEEPer:STATe?

This command allows to switch the button sounds on/off.
The STAT? query returns the current setting of the button sounds (OFF,ON).

HAMEG®
Instruments
A Rohde & Schwarz Company

Oscilloscopes

Spectrum Analyzer

Power Supplies

Modular System
Series 8000

Programmable Instruments
Series 8100

authorized dealer

www.hameg.com

Subject to change without notice

DQS-Certification: DIN EN ISO 9001:2000
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