



S Series SGA Signal Generator



GPIO Command Reference Manual

Document part no. 47090/071 (PDF version)

S Series Signal Sources

SGA SIGNAL GENERATORS

GPIB Command Reference Manual

© Aeroflex Ltd. 2011
Longacres House
Six Hills Way
Stevenage SG1 2AN
UK

*No part of this document may be reproduced or transmitted in any form
or by any means, electronic or mechanical, including photocopying,
or recorded by any information storage or retrieval system,
without permission in writing by Aeroflex Ltd.
(hereafter referred to throughout the document as 'Aeroflex').*

Document no. 47090/060 (PDF)
Issue 1
9 March 2011

Associated publications

If you want to...	Refer to...
View safety information and basic setup and operation instructions in pdf format.	S Series SGA Signal Generator Getting Started Part no. 47090/066 On the CD-ROM and at www.aeroflex.com/ .
View safety information and basic setup and operation instructions in printed format.	S Series SGA Signal Generator Getting Started Part no. 47000/066 Supplied with the instrument.
View operating information for the instrument in html Help format	S Series SGA Signal Generator Help Part no. 47090/067 On the CD-ROM and at www.aeroflex.com/
View operating information for the instrument in pdf format.	S Series SGA Signal Generator Operating Manual Part no. 467090/068 On the CD-ROM and at www.aeroflex.com/
View operating information for the instrument in a printed format.	S Series SGA Signal Generator Operating Manual Part no. 467000/068 Available as an optional extra

Contents

REMOTE OPERATION COMMANDS.....	1
Purpose	1
Introduction	1
Where to find commands.....	1
CONVENTIONS USED IN THIS MANUAL	2
Abbreviations	2
Bracketed elements	2
Case	2
Choices	3
Compound program headers.....	3
Program data.....	3
Response data	6
Remote command structure	7
COMMON COMMANDS	8
IEEE OPTIONAL COMMANDS	13
The [SOURce] subsystem — an introduction	14
RF output frequency commands	15
Modulation commands.....	24
Power commands.....	26
Measurement unit commands	30
Instrument system-level commands.....	32
Status commands	41
Output control commands	44
AM commands	47
FM commands	54
Phase modulation commands	63
Sweep commands.....	70

REMOTE OPERATION COMMANDS

Purpose

This document describes GPIB remote commands that are supported on the Aeroflex S-Series SGA Signal Generator. This document describes various SCPI (Standard Commands for Programmable Instrument) mandatory and common commands required to support the instrument, together with IEEE optional commands

Introduction

This instrument may be operated remotely via an interface that conforms to:

- IEEE Std 488.1-1987, which defines the electrical, mechanical and low-level protocol characteristics of the bus structure, the GPIB (General Purpose Interface Bus)

- IEEE Std 488.2-1987, which defines standard codes, formats, protocols and common commands for use with IEEE Std 488.1.

The instrument is not fully compliant with SCPI (Standard Commands for Programmable Instruments) because many product features are not covered by that standard, and modern software trends favor the use of instrument drivers as a means of achieving interchangeability.

However, we recognize that SCPI is in common use by system developers and a number of SCPI features that make system integration easier have been implemented. These include the command mnemonic derivation rules (long and short form) and many of the most frequently used commands themselves. Refer to SCPI 1999 (standard available from the IVI Foundation) for details.

Where to find commands

Commands are grouped into particular subsystems on the following pages, as shown in the Contents. Under each heading is an overview of the commands within that subsystem, which will help you quickly locate commands by function. Commands are arranged alphabetically within subsystems.

CONVENTIONS USED IN THIS MANUAL

Abbreviations

Long and short forms

The elements of compound and query headers have a long and a short form, as defined by SCPI. Either the long or the short form may be entered as a command; other abbreviations are not permissible.

Example:

```
STATus:OPERation:EVENT?
```

is interpreted the same as

```
STAT:OPER:EVEN
```

The short form is marked by upper-case letters, the long form corresponds to the complete word. Upper-case and lower-case serve the above purpose only, as the instrument itself does not make any distinction between upper-case and lower-case letters.

Queries always return the short form, or a numeric response in those cases where the command provides a choice of numeric or character data.

Bracketed elements

Square brackets []

Elements within the compound common program header structure that are enclosed within square brackets are optional and therefore may be omitted; the instrument processes the command in the same manner whether the bracketed element is included or not.

Example:

```
[SOURCE:]POWER[:LEVEL][:IMMEDIATE][:AMPLITUDE]
```

is interpreted the same as

```
POWER
```

This applies to parameters also. The ability to recognize the full command length ensures that the instrument complies with the SCPI standard.

Angle brackets < >

Text within angle brackets represents an actual value that needs to be inserted: for example, <freq> shows that you need to insert a frequency value in the command at this point.

Case

The software is not case-sensitive. Upper- and lower-case characters are interchangeable. There is no conflict between milli (m) and mega (M) as both cannot be applied to the same data.

Choices

The vertical bar (|)

- separates a choice of parameters:
for example, 0 | 1 means '0 or 1'
or
- separates a choice of commands:
for example, the vertical bar in [SOURce][:MODulation]:AM[1]|2[:DEPth] means that you can set the AM depth for either path 1 or path 2 (path 1 is the default): the short-form versions of the commands are AM or AM2.

Compound program headers

Compound program headers allow a complex set of commands to be built up from a smaller set of basic elements in a tree structure. The elements of a compound program header are separated by a colon (:), each colon representing a change of level in the hierarchy. Each subsystem in this instrument is organized as a separate tree structure.

The compound program header may, optionally, be followed by one or more parameters encoded as program data functional elements.

Example:

```
CALibration:IQUser:ADJust
```

Note: A leading colon is optional

Program data

Program data functional elements contain the parameters related to the program header(s). The following program data functional elements are accepted by the instrument:

<CPD>	(also known as <CHARACTER PROGRAM DATA>)
<NRf>	(also known as <DECIMAL NUMERIC PROGRAM DATA>)
<ARBITRARY BLOCK PROGRAM DATA>	

These functional elements are defined in IEEE 488.2 and the SCPI Syntax and Style handbook.

A white space must separate the command header(s) and the program data.

<white space>, as defined in IEEE Std 488.2, can be any number of ASCII characters in the range 0–9, 11–32 decimal.

<white space> is also allowed at other points in a message.

<CPD>

CPD (character program data) sets a parameter to one of a number of states that are best described by short alphanumeric strings.

Example:

ON

<NRf>

NRf (numeric representation, flexible) covers integer and floating-point representations.

Examples:

-466	Integer value
4.91	Explicitly-placed decimal point
59.5E+2	Mantissa and exponent representation

The format is known as 'flexible' because any of the three representations may be used for any type of numeric parameter.

Examples:

Where a parameter requires an integer value in the range 1 to 100, and you need to set its value to 42, the following values are accepted by the instrument:

42	Integer
42.0	Floating point
4.2E1, 4200E-2	Floating point – mantissa/exponent
41.5	Rounded up to 42
42.4	Rounded down to 42

<STRING PROGRAM DATA>

String program data consists of a number of ASCII characters enclosed in quotes. Use either pairs of single (ASCII 39) or double (ASCII 34) quotes, but do not mix single and double in a string. A quote within a string must be enclosed within an extra pair of quotes.

Example:

'This string contains the word 'Hello' '

is interpreted as

This string contains the word 'Hello'

and

"This string contains the word "Hello" "

is interpreted as

This string contains the word "Hello".

<Boolean>

<Boolean> is used as shorthand for the form ON | OFF | <NRf>. Boolean parameters have a value of 0 or 1 and are unitless.

On input, an <NRf> is rounded to an integer, and a nonzero result is interpreted as 1.

<CPD> elements ON and OFF are accepted as inputs, with ON corresponding to 1 and OFF corresponding to 0. Queries return 1 or 0, never ON or OFF.

Examples:

ON is interpreted as 1

0.4 is interpreted as 0

2.8 is interpreted as 1

Response data

The following response data functional elements are generated by the instrument:

<CRD> (also known as <CHARACTER RESPONSE DATA>)
<NR1>
<NR2>
<STRING RESPONSE DATA>

<CRD>

CRD (character response data) is returned when reading the value of a parameter that can take a number of discrete states. States are represented by short alphanumeric strings.

Example:

ON

<NR1>

This type of NR (numeric response) returns the value of integer parameters, such as an averaging number or the number of measurement points.

Examples:

15
+3
-57

<NR2>

This type of NR (numeric response) includes an explicitly placed decimal point, but no exponent.

Examples:

17.91
-18.27
+18.83

Extended numeric parameters

Most subsystems use extended numeric parameters to specify physical quantities. Extended numeric parameters accept all numeric parameter values and other special values as well.

The following are examples of extended numeric parameters:

100	any simple numeric value
-100mV	negative 100 millivolts
10DEG	10 degrees

Extended numeric parameters also include the following special parameters:

MINimum | MAXimum

The special form numeric parameters MINimum and MAXimum assume the limit values for the parameter. The maximum and minimum may be queried by sending <header>? MAXimum|MINimum. The MAXimum value refers to the largest value to which the function can currently be set, and MINimum refers to the value closest to negative infinity to which the function can currently be set.

Remote command structure

This section describes the way remote commands are used in this document. IEEE optional commands for the SGA consist of the following structures:

SCPI subsystem	System name	Instance	Setting/result
(1 mnemonic)	(2 mnemonics)	(numeric suffix)	(1 or more mnemonics)
READ	:ANALyzer :MEAS	1 2	:POWer

The purpose of the four parts of the command are:

SCPI subsystem

The root mnemonic or command subsystem as specified in SCPI; for example, READ, FETCh, SOURce.

COMMON COMMANDS

Commands recognized by all IEEE 488.2 instruments

All SCPI instruments must implement the common commands declared mandatory by the IEEE 488.2 standard. These commands have the same effect on any instrument that conforms to the standard. The headers of these commands consist of an asterisk (*) followed by three letters. Many common commands refer to the status reporting system.

The most important of the common commands is *RST, which places the instrument in a defined state. It is good practice to send *RST at the start of any program.

*CLS
*ESE\?
*ESR?
*IDN?
*OPC\?
*OPT?
*RST
*SRE\?
*STB?
*TST?
*WAI

***CLS**

Description: Clear status clears the standard event register, the error/event queue, the operation status register and the questionable status register.

Parameters: None

***ESE**

Description: The event status enable command sets the standard event status enable register to the value specified. This is an eight-bit register.

Parameters: <NRf>
Mask

Valid values: Mask: integer. Valid values are 0 to 255. Values outside the range are rejected and an error generated.

***ESE?**

Description: Reads the event status enable register. This is an eight-bit register. The contents of the event status enable register are returned in decimal form.

Parameters: None

Response: <NR1>
Mask

Returned values: Mask: integer. Values are in the range 0 to 255.

***ESR?**

Description: Reads the value of the standard event status register. This is an eight-bit register. The contents of the register are returned in decimal form. Subsequently the register is set to zero.

Parameters: None

Response: <NR1>
Register contents

Returned values: Register contents: integer. Values are in the range 0 to 255.

***IDN?**

Description: The identification query command allows information about the instrument to be read.

Parameters: None

Response: <arbitrary ASCII response data>
Manufacturer, model, serial number, software part number and issue number

Returned values: Manufacturer: string
Always returns 'Aeroflex'.

Model: string
This is the instrument's model number in the form 'NSGA Radio Test Set'

Serial number: string
This is in the form ssssss/sss where s is an ASCII digit in the range 0 to 9.

Software part number and issue number: string
This is in the form ppppp/ppp/ii.ii where p and i are ASCII digits in the range 0 to 9.

***OPC**

Description: The operation complete command sets the operation complete bit (bit 0) in the standard event status register when execution of the preceding operation is complete. This bit can be used to initiate a service request.

*OPC should be the final <program message unit> of the <program message>.

Parameters: None

***OPC?**

Description: The operation complete query returns a '1' when the preceding operation has completed.

*OPC? should be the final <query message unit> of the <program message>.

Parameters: None

Response: <NR1>
Operation complete

Returned values: Operation complete: integer. Value is 1.

***OPT?**

Description: Reads hardware options present. If no options are present a single '0' is returned, otherwise the response is up to six strings separated by commas.

Parameters: None

Response: <arbitrary ASCII response data>
Options

Returned values: Options: string

***RST**

Description: Resets the instrument to a known configuration appropriate for remote operation.

Parameters: None

***SRE**

Description: Sets the service request enable register. This is an eight-bit register.

Parameters: <NRf>
Mask

Valid values: Mask: integer. Valid values are 0 to 255. Values outside range are rejected and an error is generated.

***SRE?**

Description: Reads the service request enable register. This is an eight-bit register.

Parameters: None

Response: <NR1>
Mask

Returned values: Mask: integer. Values are in the range 0 to 255.

***STB?**

Description: Reads the status byte. This is an eight-bit register.

Parameters: None

Response: <NR1>
Status byte

Returned values: Status byte: integer. Values are in the range 0 to 255.

***TST?**

Description: Self test query. Returns a '0' when the remote operation interface and processor are operating correctly.

Parameters: None

Response: <NR1>
Self test completed

Returned values: Self test completed: integer. Value is 0.

***WAI**

Description Wait-to-Continue command. Prevents servicing of subsequent commands until all preceding commands have been executed and all signals have settled.

Parameters None

IEEE OPTIONAL COMMANDS

IEEE optional commands, also referred to as instrument-control commands, are based on a hierarchical structure and can be represented in a command tree. The command headers are built with one or several mnemonics (keywords). The first-level (root-level) mnemonic identifies a complete command system, for example:

SOURce: this mnemonic identifies the SOURce command system, which provides generator settings.

The same mnemonics may be used on different command levels, not necessarily with the same meaning. The actual meaning of a mnemonic depends on its position in the command header.

The [SOURce] subsystem — an introduction

The SOURce subsystem contains commands that cover all aspects of modulation, frequency, power and pulse generation

The [SOURce] subsystem consists of:

- The [MODulation] subsystem, which controls all aspects of modulation
- The [FREQuency] subsystem, which controls frequency parameters of the carrier and sweep signals
- The [POWer] subsystem, which sets all aspects of carrier and sweep levels
- The [PULSe] subsystem, which controls external and internal pulses and their profiles

Each of these subsystems is dealt with separately in the following sections.

The [SOURce] subsystem effectively controls the switching and configuration of internal and external signal sources and modulation paths within the instrument.

The menu structure of the [SOURce] subsystem is as follows:

[SOURce]	
[SIGGen]	
[:GENerator]	
<i>(alias :SOURce)</i>	
:FREQuency	Carrier frequency
[:MODulation]	Carrier modulation...
:POWer	RF level
:PULSe	Pulse setup

RF output frequency commands

([SOURce][:SIGGen][:GENerator]:FREQuency subsystem)

Carrier frequency, phase, sweep

Commands for:

- Setting carrier frequency, phase, phase reference, phase noise optimization and sensitivity
- Setting carrier frequency mode
- Setting carrier frequency step size
- Setting carrier sweep mode operating frequency
- Setting carrier sweep step size, spacing and mode
- Setting carrier sweep stop and start frequencies.

```
[SOURce]
  [:SIGGen]
    [:GENerator]
      (alias :SOURce)
        :FREQuency
          :CENTer\?
          [:CW|:FIXed]\?
            :AUTo\?
            :STEP
              [:INCRement]\?
          :MODE\?
          :PHASe
            [:ADJust]\?
            :OPTimization\?
            :REFerence\?
            :SENSitivity\?
          :SWEep
            :DWEL\?
            :SPACing\?
            :STARt\?
            :STEP
              [:LINear]\?
              :LOGarithmic\?
            :STOP\?
```

[SOURce][:SIGGen][:GENerator]:FREQuency:CENTer

Description: Sets the center frequency.

Parameters: <numeric_value>

Valid values: <NRf>(Hz)

Example: SOUR:SIGG:GEN:FREQ:CENT

[SOURce][:SIGGen][:GENerator]:FREQuency:CENTer?

Description: Queries the center frequency.

Parameters: None

Response: <NR2>

Returned values: Center frequency in Hz

Example: SOUR:SIGG:GEN:FREQ:CENT?

[SOURce][:SIGGen][:GENerator]:FREQuency[:CW]:FIXed]

Description: Sets the carrier frequency by value, to maximum or minimum, stepping up or down, returning to the last full setting, or setting the current value to be the new setting.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum | UP | DOWN | RETurn | REFerence

Example: SOUR:SIGG:GEN:FREQ:CW:MAX

[SOURce][:SIGGen][:GENerator]:FREQuency[:CW]:FIXed]?

Description: Queries the carrier frequency by value.

Parameters: None

Response: <NR2>

Returned values: Carrier frequency in Hz

Example: SOUR:SIGG:GEN:FREQ:CW?

[SOURce][:SIGGen][:GENerator]:FREQuency[:CW|:FIXed]:AUTo

Description: Couples the CW frequency to center frequency. Explicitly setting a value for FREQuency:CW sets AUTO OFF.

Parameters: <Boolean>

Valid values: OFF | ON | 0 | 1

Example: SOUR:SIGG:GEN:FREQ:CW:AUT ON

[SOURce][:SIGGen][:GENerator]:FREQuency[:CW|:FIXed]:AUTo?

Description: Queries the carrier frequency state.

Parameters: None

Response: <Boolean>

Returned values: OFF | ON | 0 | 1

Example: SOUR:SIGG:GEN:FREQ:CW:AUT?

[SOURce][:SIGGen][:GENerator]:FREQuency[:CW|:FIXed]:STEP[:INCRement]

Description: Sets the carrier frequency step size.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:FREQ:CW:STEP:INCR MAX

[SOURce][:SIGGen][:GENerator]:FREQuency[:CW|:FIXed]:STEP[:INCRement]?

Description: Queries the carrier frequency step size by value.

Parameters: None

Response: <NR2>

Returned values: Carrier frequency step size in Hz

Example: SOUR:SIGG:GEN:FREQ:CW:STEP:INCR?

[SOURce][:SIGGen][:GENerator]:FREQuency:MODE

Description: Sets the mode of operation of the carrier frequency.

Parameters: <CPD>

Valid values: CW | FIXed | SWEep | LIST

Example: SOUR:SIGG:GEN:FREQ:MOD SWE

[SOURce][:SIGGen][:GENerator]:FREQuency:MODE?

Description: Queries the mode of operation of the carrier frequency.

Parameters: None

Response: <CRD>

Returned values: CW | FIX | SWE | LIST

Example: SOUR:SIGG:GEN:FREQ:MOD?

[SOURce][:SIGGen][:GENerator]:FREQuency:PHASe[:ADJust]

Description: Sets the carrier frequency phase.

Parameters: <NRf>

Valid values: -360° to 0° to +360°

Example: SOUR:SIGG:GEN:FREQ:PHAS:ADJ 180

[SOURce][:SIGGen][:GENerator]:FREQuency:PHASe[:ADJust]?

Description: Queries the carrier frequency phase.

Parameters: None

Response: <NR2>

Returned values: Degrees

Example: SOUR:SIGG:GEN:FREQ:PHAS:ADJ?

[SOURce][:SIGGen][:GENerator]:FREQuency:PHASe:OPTimization

Description: Sets the phase noise performance.

Parameters: <CPD>

Valid values: LTEN less than 10 kHz: optimizes phase noise less than 10 kHz away from carrier (gives faster synthesizer settling)
GTEN greater than 10 kHz: optimizes phase noise more than 10 kHz away from carrier (gives slower synthesizer settling)

Example: SOUR:SIGG:GEN:FREQ:PHAS:OPT LTEN

[SOURce][:SIGGen][:GENerator]:FREQuency:PHASe:OPTimization?

Description: Queries the phase noise setting.

Parameters: None

Response: <CRD>

Returned values: LTEN | GTEN

Example: SOUR:SIGG:GEN:FREQ:PHAS:OPT?

[SOURce][:SIGGen][:GENerator]:FREQuency:PHASe:REFerence

Description: Sets the current carrier frequency phase as a zero reference.

Parameters: None

Valid values: None

Example: SOUR:SIGG:GEN:FREQ:PHAS:REF

[SOURce][:SIGGen][:GENerator]:FREQuency:PHASe:REFerence?

Description: Queries the carrier frequency's phase relative to the zero reference.

Parameters: None

Response: <NR2>

Returned values: Degrees

Example: SOUR:SIGG:GEN:FREQ:PHAS:REF?

[SOURce][:SIGGen][:GENerator]:FREQuency:PHASe:SENSitivity

Description: Sets the sensitivity of the controls when setting up carrier phase shift.

Parameters: <CPD>

Valid values: FINE (0.036°)
MEDium (0.360°)
COARse (1.44°)

Example: SOUR:SIGG:GEN:FREQ:PHAS:SENS FIN

[SOURce][:SIGGen][:GENerator]:FREQuency:PHASe:SENSitivity?

Description: Queries the sensitivity of the controls.

Parameters: None

Response: <CRD>

Returned values: FIN | MED | COAR

Example: SOUR:SIGG:GEN:FREQ:PHAS:SENS?

[SOURce][:SIGGen][:GENerator]:FREQuency:SWEep:DWELI

Description: Sets the time per sweep step for the carrier frequency.

Parameters: <NRf>

Valid values: <NRf> (s)

Example: SOUR:SIGG:GEN:FREQ:SWE:DWEL 5s

[SOURce][:SIGGen][:GENerator]:FREQuency:SWEep:DWELI?

Description: Queries the time per sweep step for the carrier frequency.

Parameters: None

Response: <NR2>

Returned values: Time in s.

Example: SOUR:SIGG:GEN:FREQ:SWE:DWEL?

[SOURce][:SIGGen][:GENerator]:FREQuency:SWEep:SPACing

Description: Sets the carrier sweep step points to either linear or logarithmic spacing.

Parameters: <CPD>

Valid values: LINear | LOGarithmic

Example: SOUR:SIGG:GEN:FREQ:SWE:SPAC LIN

[SOURce][:SIGGen][:GENerator]:FREQuency:SWEep:SPACing?

Description: Queries whether carrier sweep step points have linear or logarithmic spacing.

Parameters: None

Response: <CRD>

Returned values: LIN | LOG

Example: SOUR:SIGG:GEN: FREQ:SWE:SPAC?

[SOURce][:SIGGen][:GENerator]:FREQuency:SWEep:STARt

Description: Sets the start frequency for a carrier sweep.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:FREQ:SWE:STAR MAX

[SOURce][:SIGGen][:GENerator]:FREQuency:SWEep:STARt?

Description: Queries the start frequency for a carrier sweep.

Parameters: None

Response: <NR2>

Returned values: Start frequency in Hz

Example: SOUR:SIGG:GEN: FREQ:SWE:STAR?

[SOURce][:SIGGen][:GENerator]:FREQuency:SWEep:STEP[:LINear]

Description: Sets the size of linear carrier sweep steps.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:FREQ:SWE:STEP:LIN 5000Hz

[SOURce][:SIGGen][:GENerator]:FREQuency:SWEep:STEP[:LINear]?

Description: Queries the size of linear carrier sweep steps.

Parameters: None

Response: <NR2>

Returned values: Sweep step size in Hz

Example: SOUR:SIGG:GEN: FREQ:SWE:STEP:LIN?

[SOURce][:SIGGen][:GENerator]:FREQuency:SWEep:STEP:LOGarithmic

Description: Sets the size of logarithmic carrier sweep steps.

Parameters: <numeric_value>

Valid values: <NRf>(PCT) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:FREQ:SWE:STEP:LOG 3 PCT

[SOURce][:SIGGen][:GENerator]:FREQuency:SWEep:STEP[:LOGarithmic]?

Description: Queries the size of logarithmic carrier sweep steps.

Parameters: None

Response: <NR2>

Returned values: Sweep step size as a percentage

Example: SOUR:SIGG:GEN:FREQ:SWE:STEP:LOG?

[SOURce][:SIGGen][:GENerator]:FREQuency:SWEep:STOP

Description: Sets the stop frequency for the carrier sweep.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:FREQ:SWE:STOP MAX

[SOURce][:SIGGen][:GENerator]:FREQuency:SWEep:STOP?

Description: Queries the carrier sweep's stop frequency.

Parameters: None

Response: <NR2>

Returned values: Sweep stop frequency in Hz

Example: SOUR:SIGG:GEN: FREQ:SWE:STOP?

Modulation commands

([SOURce][:SIGGen][:GENerator][:MODulation] subsystem)

AM, FM, PM, pulse modulation

Commands for:

- Setting modulation mode

**[SOURce]
[:SIGGen]
[:GENerator]
(*alias* :SOURce)
[:MODulation]
:MODE\?**

[SOURce][:SIGGen][:GENerator][:MODulation]:MODE

Description: Sets the modulation mode.

Parameters: <CPD>

Valid values: AM, FM, PM or PULSE

Example: SOUR:SIGG:GEN:MOD:MOD AM

[SOURce][:SIGGen][:GENerator][:MODulation]:MODE?

Description: Returns the modulation mode.

Parameters: None

Response: <CRD>

Returned values: AM, FM, PM or PULSE

Example: SOUR:SIGG:GEN:MOD:MOD?

Power commands

([SOURce][:SIGGen][:GENerator]:POWer subsystem)

ALC, carrier level, carrier level sweeping, level steps, offsets, max. RF level

Commands for:

- Setting carrier level and step size
- Setting compensation for external losses (offsets)
- Setting an RF output limit

```
[SOURce]
  [:SIGGen]
    [:GENerator]
      (alias :SOURce)
        :POWer
          :ATTenuation
            [:LEVel]
              [:IMMediate]
                [:AMPLitude]\?
                  :STEP
                    [:INCRement]\?
                      :OFFSet
                        :STATe\?
          :LIMit
            [:AMPLitude]\?
            [:STATe]\?
```

[SOURce][:SIGGen][:GENerator]:POWer:ATTenuation

Description: Sets the attenuation level. Note that when increasing the level by 10 dB, the magnitude of the outgoing signal, as well as the LEVel, is decreased by 10 dB. Default units are as determined in the unit subsystem.

Parameters: <numeric_value>

Valid values: <NRf>dB | UP | DOWN

Example: SOUR:SIGG:GEN:POW:ATT 2dB

[SOURce][:SIGGen][:GENerator]:POWer:ATTenuation?

Description: Queries the RF attenuation level.

Parameters: None

Response: <NR2>

Returned values: Attenuation level, in units set under the unit subsystem.

Example: SOUR:SIGG:GEN:POW:ATT?

[SOURce][:SIGGen][:GENerator]:POWer[:LEVel][:IMMediate] [:AMPLitude]

Description: Sets the carrier level.

Parameters: <numeric_value>

Valid values: <NRf> | MAXimum | MINimum | UP | DOWN | RETurn | REFerence

Set by value, to maximum or minimum, stepping up or down, returning to the last full setting, or setting the current value to the last full setting.

<NRf> is in units set by :UNIT:POW.

Example: SOUR:SIGG:GEN:POW:LEV:IMM:AMP MAX

[SOURce][:SIGGen][:GENerator]:POWer[:LEVel][:IMMediate] [:AMPLitude]?

Description: Queries the carrier level by value.

Parameters: None

Response: <NR2>

Returned values: Carrier level, in units set by :UNIT:POW

Example: SOUR:SIGG:GEN:POW:LEV:IMM:AMP?

[SOURce][:SIGGen][:GENerator]:**POWer[:LEVel][:IMMediate]**
[:AMPLitude]:**STEP[:INCRement]**

Description: Sets the step size for carrier level.

Parameters: <numeric_value>

Valid values: <NRf>(dB) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:POW:LEV:IMM:AMP:STEP:INCR MAX

[SOURce][:SIGGen][:GENerator]:**POWer[:LEVel][:IMMediate]**
[:AMPLitude]:**STEP[:INCRement]?**

Description: Queries the step size for carrier level.

Parameters: None

Response: <NR2>

Returned values: Carrier level step size in dB

Example: SOUR:SIGG:GEN:POW:LEV:IMM:AMP:STEP:INCR?

[SOURce][:SIGGen][:GENerator]:**POWer[:LEVel][:IMMediate]:OFFSet**
:STATe

Description: Sets the carrier level offset on or off.

Parameters: <Boolean>

Valid values: OFF | ON | 0 | 1

Example: SOUR:SIGG:GEN:POW:LEV:IMM:OFFS:STAT ON

[SOURce][:SIGGen][:GENerator]:**POWer[:LEVel][:IMMediate]:OFFSet**
:STATe?

Description: Queries whether the carrier level offset is off (0) or on (1).

Parameters: None

Response: <Boolean>

Returned values: 0 | 1

Example: SOUR:SIGG:GEN:POW:LEV:IMM:OFFS:STAT?

[SOURce][:SIGGen][:GENerator]:POWER:LIMit[:AMPLitude]

Description: Sets the maximum RF level limit.

Parameters: <numeric_value>

Valid values: <NRf>(dB) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:POW:LIM:AMPL 4dB

[SOURce][:SIGGen][:GENerator]:POWER:LIMit[:AMPLitude]?

Description: Queries the maximum RF level limit.

Parameters: None

Response: <NR2>

Returned values: Power level limit, in the units set under unit subsystem.

Example: SOUR:SIGG:GEN:POW:LIM:AMPL?

[SOURce][:SIGGen][:GENerator]:POWER:LIMit:STATe

Description: Sets the RF level limit state to enabled or disabled.

Parameters: <Boolean>

Valid values: 0 | 1

Example: SOUR:SIGG:GEN:POW:LIM:STAT 1

[SOURce][:SIGGen][:GENerator]:POWER:LIMit:STATe?

Description: Queries the RF level limit state.

Parameters: None

Response: <Boolean>

Returned values: 0 | 1

Example: SOUR:SIGG:GEN:POW:LIM:STAT?

Measurement unit commands

([SOURce][:SIGGen][:GENerator]:UNIT subsystem)

Output level/voltage units

Commands for:

- Setting the units for output level
- Setting the voltage type for absolute/relative units.

**[SOURce]
[:SIGGen]
[:GENerator]
(*alias* :SOURce)
:UNIT
:POWER\
:Voltage**

[SOURce][:SIGGen][:GENerator]:UNIT:POWer

Description: Sets the units for the output level, for the remote interface only. Local measurement units remain as set on the instrument's front panel.

Parameters: <CPD>

Valid values: DBM | DBV | DBMV | DBUV | V | MV | UV |

Example: SOUR:SIGG:GEN:UNIT:POW DBMV

[SOURce][:SIGGen][:GENerator]:UNIT:POWer?

Description: Queries the units used for output level.

Parameters: None

Response: <CRD>

Returned values: DBM | DBV | DBMV | DBUV | V | MV | UV |

Example: SOUR:SIGG:GEN:UNIT:POW?

[SOURce][:SIGGen][:GENerator]:UNIT:VOLTage

Description: Sets the voltage type to be used for absolute and relative voltage units: DBV, DBMV, DBUV, V, MV, UV.

Parameters: <CPD>

Valid values: PD | EMF

Example: SOUR:SIGG:GEN:UNIT:VOLT MV

[SOURce][:SIGGen][:GENerator]:UNIT:VOLTage?

Description: Queries the voltage type used for voltage units.

Parameters: None

Response: <CRD>

Example: SOUR:SIGG:GEN:UNIT VOLT?

Instrument system-level commands

([SOURce][:SIGGen][:GENerator]:SYSTem subsystem)

Ethernet setup, GPIB address, error queue, keyboard locking, power-up and memory handling, touch screen on/off, SCPI version

Commands for:

- Setting the instrument's Ethernet address, DHCP and hostname
- Setting the instrument's GPIB address, baud rate and serial interface parameters
- Setting keyboard locking
- Setting the default command set
- Setting power-on memory location parameters
- Setting the default store locations for save/recall operations

SYSTem

:COMMunicate

:ETHernet

:ADDRess\?

Ethernet address

:AUTO\?

Enable DHCP

:HNAME\?

Hostname

:GPIB

[:SELF]

:ADDRess\?

GPIB address

:REMote\?

:ERRor

:ALL?

:CODE

:ALL?

[:NEXT]?

:COUNT?

[:NEXT]?

:HELP

HEADers?

:KLOCK\?

Keyboard locking

:PON

:MEMory\?

Power-on memory store

:TYPE\?

Power-on memory location

:PRESet

:SETTings

:FULL

:CLEar

:ALL

:RECALL

:SAVE

SYSTem:COMMunicate:ETHernet:ADDRess

Description: Sets the instrument's Ethernet address.

This command is rejected with a 'settings conflict' if DHCP is enabled.

Parameters: <string parameter data>,<string parameter data>

Valid values: NetMask and IP address, both in dotted quad format (for example, 255.255.255.0)

Example: SYST:COMM:ETH:ADDR '255.255.255.0','10.113.10.38'

SYSTem:COMMunicate:ETHernet:ADDRess?

Description: Returns the current NetMask and IP address in use, even if DHCP is enabled.

Parameters: None

Response: <string>,<string>

Returned values: Current NetMask and IP addresses

Example: SYST:COMM:ETH:ADDR?

SYSTem:COMMunicate:ETHernet:AUTO

Description: Enables or disables the use of DHCP to set network parameters.

Parameters: <Boolean>

Valid values: ON | OFF | 1 | 0

Example: SYST:COMM:ETH:AUTO ON

SYSTem:COMMunicate:ETHernet:AUTO?

Description: Returns the DHCP state.

Parameters: None

Response: <Boolean>

Returned values: 1 | 0

Example: SYST:COMM:ETH:AUTO?

SYSTem:COMMunicate:ETHernet:HNAME

Description: Sets the host name that appears in DHCP server logs.

Parameters: <string parameter data>

Valid values: Host name

Example: SYST:COMM:ETH:HNAM 'HOSTNAME'

SYSTem:COMMunicate:ETHernet:HNAME?

Description: Returns the instrument's host name.

Parameters: None

Response: <string>

Returned values: Host name

Example: SYST:COMM:ETH:HNAM?

SYSTem:COMMunicate:ETHernet:MADDress?

Description: Returns the Ethernet MAC address.

Parameters: None

Response: <string>

Returned values: For example, "00:50:31:04:01:02"

Example: SYST:COMM:ETH:MADD?

SYSTem:COMMunicate:GPIB[:SELF]:ADDRess

Description: Sets the instrument's GPIB address.

This command is only actioned once the EOM at the end of the message has been received and all outstanding query responses have been read.

Parameters: <numeric_value>

Valid values: Valid GPIB address
SYST:COMM:GPIB:ADDR 10

Example: SYST:COMM:GPIB:SELF:ADDR 10

SYSTem:COMMunicate:GPIB[:SELF]:ADDRess?

Description: Returns the instrument's GPIB address.

Parameters: None

Response: <NR1>

Returned values: Integer

Example: SYST:COMM:GPIB:ADDR?

SYSTem:COMMunicate:REMOte

Description: Selects the remote operation interface.

This command is only actioned once the EOM at the end of the message has been received and all outstanding query responses have been read.

Parameters: <CPD>

Valid values: GPIB | RS232 | EThernet

Example: SYST:COMM:REM 'GPIB'

SYSTem:COMMunicate:REMOte?

Description: Returns the remote operation interface that the instrument uses.

Parameters: None

Response: <CRD>

Returned values: GPIB | RS232 | ETH

Example: SYST:COMM:REM?

SYSTem:ERRor:ALL?

Description: Queries the error queue for all unread items, and removes them from the queue.

Parameters: None

Response: <NR1>,<CRD>

Returns a comma-separated list of number, string pairs in FIFO order. If the queue is empty, the response is 0, 'No error'.

Example: SYST:ERR:ALL?

SYSTem:ERRor:CODE[:ALL]?

Description: Queries the error queue for all unread items, and removes them from the queue.

Parameters: None

Response: <NR1>,...,<NR1>

Returns a comma-separated list of only the error/event code numbers in FIFO order. If the queue is empty, the response is 0.

Example: SYST:ERR:CODE:ALL?

SYSTem:ERRor:CODE[:NEXT]?

Description: Queries the error queue for the next item, and removes it from the queue.

Parameters: None

Response: <NR1>

Returns the error code only, as an integer. If the queue is empty, the response is 0.

Example: SYST:ERR:CODE?
SYST:ERR:CODE:NEXT?

SYSTem:ERRor:COUNt?

Description: Queries the error queue for the number of unread items.

Parameters: None

Response: <NR1>

If the queue is empty, the response is 0.

Example: SYST:ERR:COUN?

SYSTem:ERRor[:NEXT]?

Description: Queries the error queue for the next unread item, and removes it from the queue.

Parameters: None

Response: <NR1>,<CRD>

Returns a number and string. If the queue is empty, the response is 0, 'No error'.

Example: SYST:ERR?
SYST:ERR:NEXT?

SYSTem:HELP:HEADers?

Description: Returns a list of the instrument command headers.

Parameters: None

Response: <arbitrary block response data>

Example: SYST:HELP:HEAD?

SYSTem:KLOCK

Description: Locks and unlocks the touchscreen.

Parameters: <Boolean>

Valid values: ON | OFF | 1 | 0

*RST sets: OFF

Example: SYST:KLOC 1

SYSTem:KLOCK?

Description: Queries whether the touchscreen is locked (1) or unlocked (0).

Parameters: None

Response: <Boolean>

Returned values: 0 | 1

Example: SYST:KLOC?

SYSTem:PON:MEMory

Description: Specifies a user-defined power-on memory store number.

Parameters: <numeric_value>

Valid values: Valid store number.

Example: SYST:PON:MEM 2000

SYSTem:PON:MEMory?

Description: Returns the power-on memory number.

Parameters: None

Response: <NR1>

Returned values: Store number.

Example: SYST:PON:MEM?

SYSTem:PON:TYPE

Description: Selects power-on either from the default memory location (factory-preset) or one specified by :SYSTem:PON:MEMory above.

Parameters: <CPD>

Valid values: DEFault | MEMory

*RST sets: No effect on the language set.

Example: SYST:PON:TYPE 'DEF'

SYSTem:PON:TYPE?

Description: Queries whether the instrument powers up from the default memory location or one specified by :SYSTem:PON:MEMory above.

Parameters: None

Response: <CRD>

Returned values: DEF | MEM

Example: SYST:PON:TYPE?

SYSTem:PRESet

Description: Returns the instrument to its default state.

Parameters: None

Example: SYST:PRES

SYSTem:SETTings:FULL:CLEar:ALL

Description: Clears all user-defined memory locations.

Parameters: none

Example: SYST:SETT:FULL:CLE:ALL

SYSTem:SETTings:FULL:RECall

Description: Recalls the contents of the specified memory location.

Parameters: <numeric_value>

Valid values: Valid store number | UP | DOWN

Example: SYST:SETT:FULL:REC UP

SYSTem:SETTings:FULL:SAVE

Description: Save the current configuration to the memory location.

Parameters: <numeric_value>

Valid values: Valid store number.

Example: SYST:SETT:FULL:SAVE

Status commands

(STATus subsystem)

Commands for determining the state of the instrument

This subsystem controls the SCPI-defined status-reporting structures. SCPI defines QUEStionable, OPERation, Instrument SUMmary and INSTRument registers, in addition to those in *IEEE 488.2*. These registers conform to the *IEEE 488.2* specification, and each may consist of a condition register, an event register, an enable register, and negative and positive transition filters.

STATus

- <StatReg>
 - :CONDition?
 - :ENABle\?
 - :EVENT?
 - :NTRansition\?
 - :PTRansition\?
- :PRESet

where <StatReg> is:

- :OPERation
- :OPERation:TRIGger
- :QUEStionable
- :QUEStionable:FREQuency
- :QUEStionable:MODulation
- :QUEStionable:MODulation:AM
- :QUEStionable:MODulation:FM
- :QUEStionable:POWER

STATus:<StatReg>:CONDition?

Description: Reads the contents of the status register.

Parameters: None.

Response: <NR1> Status register contents.

Example: STAT:OPER:COND?
STAT:QUES:COND?

STATus:<StatReg>:ENABLE

Description: Sets the enable mask, which allows true conditions in the status event register to be reported in the summary bit. If a bit is '1' in the enable register and its associated event bit makes a transition to true, a positive transition occurs in the associated summary bit.

Parameters: <NRf> Mask

Valid values: 0–7FFFH

Example: STAT:OPER:ENAB 2000
STAT:QUES:ENAB 1536

STATus:<StatReg>:ENABLE?

Description: Reads the enable mask for the status register.

Parameters: [<NRf>] [Mask]

Response: <NR1> Mask

Returned values: 0–7FFFH

Example: STAT:OPER:ENAB?

STATus:<StatReg>:EVENT?

Description: Reads the contents of the event register associated with the operation status register.

Parameters: None.

Response: <NR1> Event register contents.

Returned values: 0–7FFFH

Example: STAT:OPER:EVEN?

STATus:<StatReg>:NTRansition

Description: Sets the negative transition filter in the status register. Setting a bit in the negative transition filter causes a 1 to 0 transition in the corresponding bit of the associated condition register, causing a '1' to be written in the associated bit of the corresponding event register.

Parameters: <NRf> Mask

Valid values: 0–7FFFH

Example: STAT:OPER:NTR 2000
STAT:QUES:NTR 2000

STATus:<StatReg>:NTRansition?

Description: Reads the negative transition mask for the status register.

Parameters: [<NRf>] [Mask]

Response: <NR1> Mask

Returned values: 0–7FFFH

Example: STAT:OPER:NTR?

STATus:<StatReg>:PTRansition

Description: Sets the positive transition filter in the status register. Setting a bit in the positive transition filter causes a 0 to 1 transition in the corresponding bit of the associated condition register, causing a '1' to be written in the associated bit of the corresponding event register.

Parameters: <NRf> Mask

Valid values: 0–7FFFH

Example: STAT:OPER:PTR 2000
STAT:QUES:PTR 2000

STATus:<StatReg>:PTRansition?

Description: Reads the positive transition mask for the status register.

Parameters: [<NRf>] [Mask]

Response: <NR1> Mask

Returned values: 0–7FFFH

Example: STAT:OPER:PTR?

Output control commands

(OUTPut subsystem)

The OUTput subsystem effectively controls the switching of modulation paths within the instrument.

OUTPut		
:MODulation	Enable/disable modulation...	
:AM[1] 2		
[:STATe]\?		...AM
:FM[1] 2		
[:STATe]\?		...FM
:PM[1] 2		
[:STATe]\?		...phase
[:STATe]\?		...all

OUTPut:MODulation:AM[1]|2[:STATe]

Description: Turns the source feeding the AM1 or AM2 modulator on or off; other active modulators are not affected.

Parameters: <Boolean>

Valid values: OFF | ON | 0 | 1

Example: OUT:MOD:AM1 ON
OUT:MOD:AM2:STAT ON

OUTPut:MODulation:AM[1]|2[:STATe]?

Description: Queries the state of the amplitude modulation source.

Parameters: None

Response: <Boolean>

Returned values: 0 | 1

Example: OUT:MOD:AM1?
OUT:MOD:AM2:STAT?

OUTPut:MODulation:FM[1]|2[:STATe]

Description: Turns the source feeding the FM1 or FM2 modulator on or off; other active modulators are not affected.

Parameters: <Boolean>

Valid values: OFF | ON | 0 | 1

Example: OUT:MOD:FM1 ON
OUT:MOD:FM2:STAT ON

OUTPut:MODulation:FM[1]|2[:STATe]?

Description: Queries the state of the frequency modulation source.

Parameters: None

Response: <Boolean>

Returned values: 0 | 1

Example: OUT:MOD:FM1?
OUT:MOD:FM2:STAT?

OUTPut:MODulation:PM[1]|2[:STATe]

Description: Turns the source feeding the PM1 or PM2 modulator on or off; other active modulators are not affected.

Parameters: <Boolean>

Valid values: OFF | ON | 0 | 1

Example: OUT:MOD:PM1 ON
OUT:MOD:PM1:STAT ON

OUTPut:MODulation:PM[1]|2[:STATe]?

Description: Queries the state of the phase modulation source.

Parameters: None

Response: <Boolean>

Returned values: 0 | 1

Example: OUT:MOD:PM1?
OUT:MOD:PM1:STAT?

OUTPut:MODulation[:STATe]

Description: Enables or disables all the active modulation outputs.

When ON, this command causes each modulation output to adopt the state set by its relevant [SOURce][:MODulation]:<modn>:STATe command.

Corresponds to the MOD ON/OFF button.

Parameters: <Boolean>

Valid values: OFF | ON | 0 | 1

Example: OUT:MOD ON
OUT:MOD:STAT ON

OUTPut:MODulation[:STATe]?

Description: Queries the state of the active modulation outputs.

Parameters: None

Response: <Boolean>

Returned values: 0 | 1

Example: OUT:MOD?
OUT:MOD:STAT?

AM commands

([SOURce][SOURce][:SIGGen][:GENerator][:MODulation]:AM subsystem)

AM depth, source, frequency, waveshape, phase, input parameters

Commands for:

- Setting AM frequency and frequency step size
- Setting AM depth and depth step size
- Setting AM mode (fixed or sweep)
- Setting AM waveshape and time per sweep
- Setting internal/external source on/off
- Setting phase relationship of AM2 with respect to AM1.

```
[SOURce]
  [:SIGGen]
    [:GENerator]
      (alias :SOURce)
        [:MODulation]
          :AM[1]|2
            [:DEPTH]\?
              :STEP
                [:INCRement]\?
            :EXTernal
              :COUPling\?
            :INTernal
              :FREQuency\?
                :STEP
                  [:INCRement]\?
            :PHASe\?
              :ADJust\?
              :OPTimization\?
              :REFerence\?
              :SENSitivity\?
            :SHAPE\?
          :SOURce\?
        :STATe\?
```

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2[:DEPT]h

Description: Sets the AM depth as a percentage.

Parameters: <numeric_value>

Valid values: <NRf>(PCT) | MAXimum | MINimum | UP | DOWN | RETurn | REFerence

Set by value, to maximum or minimum, stepping up or down, returning to the last full setting, or setting the current value to the last full setting.

Example: SOUR:SIGG:GEN:MOD:AM1:DEPT MAX

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2[:DEPT]h?

Description: Queries the AM depth.

Parameters: None

Response: <NR2>

Returned values: AM depth as a percentage

Example: OUR:SIGG:GEN:MOD:AM1:DEPT?

**[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2[:DEPT]h
:STEP[:INCR]ement**

Description: Sets the AM depth step size as a percentage.

Parameters: <numeric_value>

Valid values: <NRf> (PCT) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:MOD:AM1:DEPT:STEP:INCR MAX

**[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2[:DEPT]h
:STEP[:INCR]ement?**

Description: Queries the AM depth step size.

Parameters: None

Response: <NR2>

Returned values: AM depth step size as a percentage

Example: SOUR:SIGG:GEN:MOD:AM1:DEPT:STEP:INCR?

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2:EXTernal:COUPLing

Description: Selects AC or DC coupling for the external source.

Parameters: <CPD>

Valid values: AC | DC

Example: SOUR:SIGG:GEN:MOD:AM1:EXT:COUP AC

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2:EXTernal:COUPLing?

Description: Queries whether the external source is AC- or DC-coupled.

Parameters: None

Response: <CRD>

Returned values: AC | DC

Example: SOUR:SIGG:GEN:MOD:AM1:EXT:COUP?

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2:INTernal:FREQuency

Description: Sets the internal AM frequency.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum | UP | DOWN | RETurn | REFerence
Set by value to maximum or minimum, stepping up or down, returning to the last full setting, or setting the current value to the last full setting.

Example: SOUR:SIGG:GEN:MOD:AM1:INT:FREQ MIN

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2:INTernal:FREQuency?

Description: Queries the internal AM frequency.

Parameters: None

Response: <NR2>

Returned values: AM frequency in Hz

Example: SOUR:SIGG:GEN:MOD:AM1:INT:FREQ?

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2:INTernal:FREQuency:STEP[:INCRement]

Description: Sets the internal AM frequency step.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:MOD:AM1:INT:FREQ:STEP:INCR MAX

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2:INTernal:FREQuency:STEP[:INCRement]?

Description: Queries the internal AM frequency step size.

Parameters: None

Response: <NR2>

Returned values: AM frequency step size in Hz

Example: SOUR:SIGG:GEN:MOD:AM1:INT:FREQ:STEP:INCR?

[SOURce][:SIGGen][:GENerator][:MODulation]:AM2:INTernal:PHASe:ADJust

Description: Controls the phase offset value relative to the reference. A negative value for adjust causes the output signal to lag the reference.

Parameters: <numeric_value>

Valid values: <NRf> | UP | DOWN

Example: SOUR:SIGG:GEN:MOD:AM1:INT:PHAS:ADJ UP

[SOURce][:SIGGen][:GENerator][:MODulation]:AM2:INTernal:PHASe:ADJust?

Description: Queries the phase offset value relative to the reference.

Parameters: None

Response: <NR2>

Returned values: Phase angle (degrees)

Example: SOUR:SIGG:GEN:MOD:AM1: INT:PHAS:ADJ?

[SOURce][:SIGGen][:GENerator][:MODulation]:AM2:INTernal:PHASe:OPTimization

Description: Sets phase noise performance.

Parameters: <CPD>

Valid values: LTEN | GTEN

Example: SOUR:SIGG:GEN:MOD:AM1:INT:PHAS:OPT LTEN

[SOURce][:SIGGen][:GENerator][:MODulation]:AM2:INTernal:PHASe:OPTimization?

Description: Queries the phase noise performance.

Parameters: None

Response: <CRD>

Returned values: LTEN | GTEN

Example: SOUR:SIGG:GEN:MOD:AM1:INT:PHAS:OPT?

[SOURce][:SIGGen][:GENerator][:MODulation]:AM2:INTernal:PHASe:REFerence

Description: Sets the current carrier frequency phase as a zero reference.

Parameters: None

Valid values: None

Example: SOUR:SIGG:GEN:MOD:AM1:INT:PHAS:REF

[SOURce][:SIGGen][:GENerator][:MODulation]:AM2:INTernal:PHASe:REFerence?

Description: Queries the current carrier frequency phase reference.

Parameters: None

Response: <NR2> degrees

Returned values: Phase angle (degrees)

Example: SOUR:SIGG:GEN:MOD:AM1:INT:PHAS:REF?

**[SOURce][:SIGGen][:GENerator][:MODulation]:AM2:INTernal:PHASe
:SENSitivity**

Description: Selects the sensitivity of the control when setting up the phase offset of AM2 relative to AM1.

Parameters: <CPD>

Valid values: FINE (0.01° resolution)
MEDIUM (0.1° resolution)
COARSE (1.0° resolution)

Example: SOUR:SIGG:GEN:MOD:AM1:INT:PHAS:SENS MED

**[SOURce][:SIGGen][:GENerator][:MODulation]:AM2:INTernal:PHASe
:SENSitivity?**

Description: Queries the sensitivity of the control when setting up the phase offset of AM2 relative to AM1.

Parameters: None

Response: <CRD>

Returned values: FIN | MED | COAR

Example: SOUR:SIGG:GEN:MOD:AM1:INT:PHAS:SENS?

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2:INTernal:SHAPE

Description: Selects the shape of the internally-generated AM waveform.

Parameters: <CPD>

Valid values: SINE | SQUARE | TRIANGLE | RAMP

Example: SOUR:SIGG:GEN:MOD:AM1:INT:SHAP SINE

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2:INTernal:SHAPE?

Description: Queries the shape of the internally generated AM.

Parameters: None

Response: <CRD>

Returned values: SINE | SQU | TRI | RAMP

Example: SOUR:SIGG:GEN:MOD:AM1:INT:SHAP?

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2:SOURce

Description: Selects either an internal or external source to generate AM.

Parameters: <CPD>

Valid values: INTernal | EXTernal

Example: SOUR:SIGG:GEN:MOD:AM1:SOUR INT

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2:SOURce?

Description: Queries whether the source for AM is internal or external.

Parameters: None

Response: <CRD>

Returned values: INT | EXT

Example: SOUR:SIGG:GEN:MOD:AM1:SOUR?

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2:STATe

Description: Adds AM1 or AM2 to the set of active modulations, or removes AM1 or AM2 from it.

Parameters: <Boolean>

Valid values: OFF | ON | 0 | 1

Example: SOUR:SIGG:GEN:MOD:AM1:STAT ON

[SOURce][:SIGGen][:GENerator][:MODulation]:AM[1]|2:STATe?

Description: Queries whether the AM path is on (1) or off (0).

Parameters: None

Response: <Boolean>

Returned values: 0 | 1

Example: SOUR:SIGG:GEN:MOD:AM1:STAT?

FM commands

([SOURce][:SIGGen][:GENerator][:MODulation]:FM subsystem)

FM deviation, source, frequency, waveshape, mod. sweep, phase, input parameters, DC null

Commands for:

- Setting FM frequency and frequency step size
- Setting FM depth and depth step size
- Setting FM coupling
- Setting DC null
- Setting FM mode (fixed or sweep)
- Setting FM waveshape and time per sweep
- Setting FM sweep parameters
- Setting internal/external source on/off
- Setting phase relationship of FM2 with respect to FM1.

```
[SOURce]
  [:SIGGen]
    [GENerator]
      (alias :SOURce)
        [:MODulation]
          :FM[1]|2
            [:DEViation]\?
              :STEP
                [:INCRement]\?
            :EXTernal
              :COUPling\?
              :DNULI
            :INTernal
              :FREQUency\?
                :STEP
                  [:INCRement]\?
            :PHASe\?
              :ADJJust\?
              :OPTimization\?
              :REFerence
              :SENSitivity\?
```

:SHAPE\?
:SOURCE\?
:STATE\?

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2[:DEViation]

Description: Sets the FM deviation.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum | UP | DOWN | RETurn | REFerence

Set by value, to maximum or minimum, stepping up or down, returning to the last full setting, or setting the current value to the last full setting.

Example: SOUR:SIGG:GEN:MOD:FM1:DEPT MAX

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2[:DEViation]?

Description: Queries the FM deviation.

Parameters: None

Response: <NR2>

Returned values: FM deviation as a percentage

Example: SOUR:SIGG:GEN:MOD:FM1:DEPT?

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2[:DEViation]:STEP[:INCRement]

Description: Sets the FM deviation step size.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:MOD:FM1:DEPT:STEP:INCR MAX

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2[:DEViation]:STEP[:INCRement]?

Description: Queries the FM deviation step size.

Parameters: None

Response: <NR2>

Returned values: FM deviation step size

Example: SOUR:SIGG:GEN:MOD:FM1:DEPT:STEP:INCR?

**[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:EXTernal
:COUPling**

Description: Selects AC or DC coupling for the external source.

Parameters: <CPD>

Valid values: AC | DC

Example: SOUR:SIGG:GEN:MOD:FM1:EXT:COUP AC

**[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:EXTernal
:COUPling?**

Description: Queries whether the external source is AC- or DC-coupled.

Parameters: None

Response: <CRD>

Returned values: AC | DC

Example: SOUR:SIGG:GEN:MOD:FM1:EXT:COUP?

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:EXTernal:DNULI

Description: Performs a DC FM null.

Reminder: you need to apply a ground reference to the external modulation input.

Parameters: None

Valid values: None

Example: SOUR:SIGG:GEN:MOD:FM1:EXT:DNUL

[SOURce][:SIGGen][:GENerator][:MODulation]FM[1]|2:INTernal :FREQuency

Description: Sets the internal FM frequency.

Parameters: <numeric_value>

Valid values: <NRf> (Hz) | MAXimum | MINimum | UP | DOWN | RETurn | REFerence

Set by value, to maximum or minimum, stepping up or down, returning to the last full setting, or setting the current value to the last full setting.

Example: SOUR:SIGG:GEN:MOD:FM1:INT:FREQ MIN

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:INTernal :FREQuency?

Description: Queries the internal FM frequency.

Parameters: None

Response: <NR2>

Returned values: FM frequency step size in Hz

Example: SOUR:SIGG:GEN:MOD:FM1:INT:FREQ?

[SOURce][:SIGGen][:GENerator][:MODulation]FM[1]|2:INTernal :FREQuency:STEP[:INCRement]

Description: Sets the internal FM frequency step.

Parameters: <numeric_value>

Valid values: <NRf> (Hz) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:MOD:FM1:INT:FREQ:STEP:INCR MAX

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:INTernal :FREQuency:STEP[:INCRement]?

Description: Queries the internal FM frequency step size.

Parameters: None

Response: <NR2>

Returned values: FM frequency step size in Hz

Example: SOUR:SIGG:GEN:MOD:FM1:INT:FREQ:STEP:INCR?

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:INTernal:PHASe:ADJust

Description: Controls the phase offset value relative to the reference. A negative value causes the output signal to lag the reference.

Parameters: <numeric_value>

Valid values: <NRf> | UP | DOWN

Example: SOUR:SIGG:GEN:MOD:FM1:INT:PHAS:ADJ UP

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:INTernal:PHASe:ADJust?

Description: Queries the phase offset value relative to reference.

Parameters: None

Response: <NR2>

Returned values: Phase angle (degrees)

Example:Example: SOUR:SIGG:GEN:MOD:FM1: INT:PHAS:ADJ?

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:INTernal:PHASe:OPTimization

Description: Sets phase noise performance.

Parameters: <CPD>

Valid values: LTEN | GTEN

Example: SOUR:SIGG:GEN:MOD:FM1:INT:PHAS:OPT LTEN

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:INTernal:PHASe:OPTimization?

Description: Queries the phase noise performance.

Parameters: None

Response: <CRD>

Returned values: LTEN | GTEN

Example: SOUR:SIGG:GEN:MOD:FM1:INT:PHAS:OPT?

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]2:INTernal:PHASe:REFerence

Description: Sets current carrier frequency phase as a zero reference.

Parameters: None

Valid values: None

Example: SOUR:SIGG:GEN:MOD:FM1:INT:PHAS:REF

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]2:INTernal:PHASe:SENSitivity

Description: Selects the sensitivity of the controls when setting up the phase offset of FM2 relative to FM1.

Parameters: <CPD>

Valid values: FINE (0.01° resolution)
MEDIUM (0.1° resolution)
COARSE (1.0° resolution)

Example: SOUR:SIGG:GEN:MOD:FM1:INT:PHAS:SENS MED

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]2:INTernal:PHASe:SENSitivity?

Description: Queries the sensitivity of the controls when setting up the phase offset of FM2 relative to FM1.

Parameters: None

Response: <CRD>

Returned values: FIN | MED | COAR

Example: SOUR:SIGG:GEN:MOD:FM1:INT:PHAS:SENS?

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:INTernal:SHAPE

Description: Selects the shape of the internally generated FM waveform.

Parameters: <CPD>

Valid values: SINE | SQUare | TRIangle | RAMP

Example: SOUR:SIGG:GEN:MOD:FM1:INT:SHAP SINE

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:INTernal:SHAPE?

Description: Queries the shape of the internally generated FM waveform.

Parameters: None

Response: <CRD>

Returned values: SINE | SQU | TRI | RAMP

Example: SOUR:SIGG:GEN:MOD:FM1:INT:SHAP?

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:SOURce

Description: Selects either an internal or external source to generate FM.

Parameters: <CPD>

Valid values: INTernal | EXTernal

Example: SOUR:SIGG:GEN:MOD:FM1:SOUR INT

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:SOURce?

Description: Queries whether the source for FM is internal or external.

Parameters: None

Response: <CRD>

Returned values: INT | EXT

Example: SOUR:SIGG:GEN:MOD:FM1:SOUR?

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:STATe

Description: Adds FM1 or FM2 to the set of active modulations, or removes FM1 or FM2 from it.

Parameters: <Boolean>

Valid values: OFF | ON | 0 | 1

Example: SOUR:SIGG:GEN:MOD:FM1:STAT ON

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:STATe?

Description: Queries whether the FM path is on (1) or off (0).

Parameters: None

Response: <Boolean>

Returned values: 0 | 1

Example: SOUR:SIGG:GEN:MOD:FM1:STAT?

Phase modulation commands

([SOURce][:SIGGen][:GENerator][:MODulation]:PM subsystem)

Phase modulation deviation, source, frequency, waveshape, phase, input parameters

Commands for:

- Setting phase modulation frequency and frequency step size
- Setting phase modulation deviation and deviation step size

```
[SOURce]
  [:SIGGen]
    [:GENerator]
      (alias :SOURce)
        [:MODulation]
          :PM[1]|2
            [:DEViation]\?
              :STEP
                [:INCRement]\?
            :EXTernal
              :COUPling\?
            :INTernal
              :FREQuency\?
                :STEP
                  [:INCRement]\?
            :PHASe\?
              :ADJust\?
              :OPTimization\?
              :REFerence
              :SENSitivity\?
          :SOURce\?
          :STATe\?
```

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2[:DEViation]

Description: Sets the phase modulation deviation.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum | UP | DOWN | RETurn | REFerence

Set by value, to maximum or minimum, stepping up or down, returning to the last full setting, or setting the current value to the last full setting.

Example: SOUR:SIGG:GEN:MOD:PM1:DEPT MAX

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2[:DEViation]?

Description: Queries the phase modulation deviation.

Parameters: None

Response: <NR2>

Example: SOUR:SIGG:GEN:MOD:PM1:DEPT?

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2[:DEViation]:STEP[:INCRe ment]

Description: Sets the phase modulation deviation step size.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:MOD:PM1:DEPT:STEP:INCR MAX

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2[:DEViation]:STEP[:INCRe ment]?

Description: Queries the phase modulation deviation step size.

Parameters: None

Response: <NR2>

Returned values: Phase modulation step size in Hz

Example: SOUR:SIGG:GEN:MOD:PM1:DEPT:STEP:INCR?

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2:EXTernal:COUPling

Description: Selects AC or DC coupling for the external source.

Parameters: <CPD>

Valid values: AC | DC

Example: SOUR:SIGG:GEN:MOD:PM1:EXT:COUP AC

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2:EXTernal:COUPling?

Description: Queries whether the external source is AC- or DC-coupled.

Parameters: None

Response: <CRD>

Returned values: AC | DC

Example: SOUR:SIGG:GEN:MOD:PM1:EXT:COUP?

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2:INTernal:FREQuency

Description: Sets the internal phase modulation frequency.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum | UP | DOWN | RETurn | REFerence

Set by value, to maximum or minimum, stepping up or down, returning to the last full setting, or setting the current value to the last full setting.

Example: SOUR:SIGG:GEN:MOD:PM1:INT:FREQ MIN

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2:INTernal:FREQuency?

Description: Queries the internal phase modulation frequency.

Parameters: None

Response: <NR2>

Returned values: Phase modulation frequency in Hz

Example: SOUR:SIGG:GEN:MOD:PM1:INT:FREQ?

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2:INTernal:FREQuency:STEP[:INCRement]

Description: Set the internal phase modulation frequency step.

Parameters: <numeric_value>

Valid values: <NRf>(Hz) | MAXimum | MINimum

Example: SOUR:SIGG:GEN:MOD:PM1:INT:FREQ:STEP:INCR MAX

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2:INTernal:FREQuency:STEP[:INCRement]?

Description: Queries the internal phase modulation frequency step size.

Parameters: None

Response: <NR2>

Returned values: Phase modulation frequency step size in Hz

Example: SOUR:SIGG:GEN:MOD:PM1:INT:FREQ:STEP:INCR?

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2:INTernal:PHASe:ADJust

Description: Controls the phase offset value relative to the reference. A negative value causes the output signal to lag the reference.

Parameters: <numeric_value>

Valid values: <NRf> | UP | DOWN

Example: SOUR:SIGG:GEN:MOD:PM1:INT:PHAS:ADJ UP

[SOURce][:SIGGen][:GENerator][:MODulation]:FM[1]|2:INTernal:PHASe:ADJust?

Description: Queries the phase offset value relative to reference.

Parameters: None

Response: <NR2>

Returned values: Phase angle (degrees)

Example:Example: SOUR:SIGG:GEN:MOD:PM1:INT:PHAS:ADJ?

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2:INTernal:PHASe:OPTimization

Description: Sets phase noise performance.

Parameters: <CPD>

Valid values: LTEN | GTEN

Example: SOUR:SIGG:GEN:MOD:PM1:INT:PHAS:OPT LTEN

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2:INTernal:PHASe:OPTimization?

Description: Queries the phase noise performance.

Parameters: None

Response: <CRD>

Returned values: LTEN | GTEN

Example: SOUR:SIGG:GEN:MOD:PM1:INT:PHAS:OPT?

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2:INTernal:PHASe:REFerence

Description: Sets current carrier frequency phase as a zero reference.

Parameters: None

Valid values: None

Example: SOUR:SIGG:GEN:MOD:PM1:INT:PHAS:REF

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]2:INTernal:PHASe:SENSitivity

Description: Selects the sensitivity of the controls when setting up the phase offset of PM2 relative to PM1.

Parameters: <CPD>

Valid values: FINE (0.01° resolution)
MEDIum (0.1° resolution)
COARse (1.0° resolution)

Example: SOUR:SIGG:GEN:MOD:PM1:INT:PHAS:SENS MED

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]2:INTernal:PHASe:SENSitivity?

Description: Queries the sensitivity of the controls when setting up the phase offset of PM2 relative to PM1.

Parameters: None

Response: <CRD>

Returned values: FIN | MED | COAR

Example: SOUR:SIGG:GEN:MOD:PM1:INT:PHAS:SENS?

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]2:SOURce

Description: Selects either an internal or external source to generate phase modulation.

Parameters: <CPD>

Valid values: INTernal | EXTernal

Example: SOUR:SIGG:GEN:MOD:PM1:SOUR INT

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]2:SOURce?

Description: Queries whether the source for phase modulation is internal or external.

Parameters: None

Response: <CRD>

Returned values: INT | EXT

Example: SOUR:SIGG:GEN:MOD:PM1:SOUR?

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2:STATe

Description: Adds PM1 or PM2 to the set of active modulations, or removes PM1 or PM2 from it.

Parameters: <Boolean>

Valid values: OFF | ON | 0 | 1

Example: SOUR:SIGG:GEN:MOD:PM1:STAT ON

[SOURce][:SIGGen][:GENerator][:MODulation]:PM[1]|2:STATe?

Description: Queries whether the phase modulation path is on (1) or off (0).

Parameters: None

Response: <Boolean>

Returned values: 0 | 1

Example: SOUR:SIGG:GEN:MOD:PM1:STAT?

Sweep commands

([SOURce]:SWEep subsystem)

Sweep handling and triggering

Commands for:

- Controlling operation of a frequency or power sweep
- Setting the sweep trigger mode and slope.

```
[SOURce]
[:SIGGen]
[:GENerator]
(alias :SOURce)
:SWEep
:ABORt
:CONTinue
:INITiate
:OPERation\?
:PAUSe
:RESet
:TRIGger
[:MODE]\?
:SLOPe\?
```

[SOURce][:SIGGen][:GENerator]:SWEep:ABORt

Description: Stops the sweep immediately.

Parameters: None

Example: SOUR:SIGG:GEN:SWE:ABOR

[SOURce][:SIGGen][:GENerator]:SWEep:CONTinue

Description: Continues a paused sweep.

Parameters: None

Example: SOUR:SIGG:GEN:SWE:CONT

[SOURce][:SIGGen][:GENerator]:SWEep:INITiate

Description: Starts a sweep.

Parameters: None

Example: SOUR:SIGG:GEN:SWE:INIT

[SOURce][:SIGGen][:GENerator]:SWEep:OPERation

Description: Sets whether the sweep mode is single or continuous.

Parameters: <CPD>

Valid values: SINGLE | CONTinuous

Example: SOUR:SIGG:GEN:SWE:OPER SING

[SOURce][:SIGGen][:GENerator]:SWEep:OPERation?

Description: Returns whether the sweep mode is single or continuous.

Parameters: None

Response: <CRD>

Returned values: SING | CONT

Example: SOUR:SIGG:GEN:MOD:AM1:DEPT?

[SOURce][:SIGGen][:GENerator]:SWEep:PAUSE

Description: Pauses the sweep.

Parameters: None

Example: SOUR:SIGG:GEN:SWE:PAUS

[SOURce][:SIGGen][:GENerator]:SWEep:RESet

Description: Resets the sweep to its starting value of power or frequency.

Parameters: None

Example: SOUR:SIGG:GEN:SWE:RES

[SOURce][:SIGGen][:GENerator]:SWEep:TRIGger[:MODE]

Description: Sets the trigger mode to off, start, start then stop, or step.

Parameters: <CPD>

Valid values: OFF | START | SSTOP | STEP

Example: SOUR:SIGG:GEN:SWE:TRIG:MOD STAR

[SOURce][:SIGGen][:GENerator]:SWEep:TRIGger[:MODE]?

Description: Queries the trigger mode for the sweep.

Parameters: None

Response: <CRD>

Returned values: OFF | STAR | SSTOP | STEP

Example: SOUR:SIGG:GEN:SWE:TRIG:MOD?

[SOURce][:SIGGen][:GENerator]:SWEep:TRIGger:SLOPe

Description: Sets the polarity of the sweep trigger.

Parameters: <CPD>

Valid values: POSitive | NEGative

Example: SOUR:SIGG:GEN:SWE:TRIG:SLOP POS

[SOURce][:SIGGen][:GENerator]:SWEep:TRIGger:SLOPe?

Description: Queries the polarity of the sweep trigger.

Parameters: None

Response: <CRD>

Returned values: POS | NEG

Example: SOUR:SIGG:GEN:SWE:TRIG:POL?

AEROFLEX LIMITED

SOFTWARE LICENSE AND WARRANTY

This document is an Agreement between the user of this Licensed Software, the Licensee, and Aeroflex Limited ('Aeroflex'), the Licensor. By installing or commencing to use the Licensed Software you accept the terms of this Agreement. If you do not agree to the terms of this Agreement do not use the Licensed Software.

1. DEFINITIONS

The following expressions will have the meanings set out below for the purposes of this Agreement:

Add-In Application Software	Licensed Software that may be loaded separately from time to time into the Designated Equipment to improve or modify its functionality
Computer Application Software	Licensed Software supplied to run on a standard PC or workstation
Designated Equipment	means either: the single piece of equipment or system supplied by Aeroflex upon which the Licensed Software is installed; or a computer that is connected to a single piece of equipment or system supplied by Aeroflex upon which computer the Licensed Software is installed
Downloaded Software	any software downloaded from an Aeroflex web site
Embedded Software	Licensed Software that forms part of the Designated Equipment supplied by Aeroflex and without which the Equipment cannot function
License Fee	means either the fee paid or other consideration given to Aeroflex for the use of the Licensed Software on the Designated Equipment
Licensed Software	all and any programs, listings, flow charts and instructions in whole or in part including Add-in, Computer Application, Downloaded and Embedded Software supplied to work with Designated Equipment
PXI Software	Licensed Software specific to Aeroflex's 3000 Series PXI product range

2. LICENSE FEE

The Licensee shall pay the License Fee to Aeroflex in accordance with the terms of the contract between the Licensee and Aeroflex.

3. TERM

This Agreement shall be effective from the date of receipt or download (where applicable) of the Licensed Software by the Licensee and shall continue in force until terminated under the provisions of Clause 8.

4. LICENCE

- 4.1 The following rights and restrictions in this Article 4 apply to all Licensed Software unless otherwise expressly stated in other Articles of this Agreement.
- 4.2 Unless and until terminated, this License confers upon the Licensee the non-transferable and non-exclusive right to use the Licensed Software on the Designated Equipment.
- 4.3 The Licensee may not use the Licensed Software on other than the Designated Equipment, unless written permission is first obtained from Aeroflex and until the appropriate additional License Fee has been paid to Aeroflex.
- 4.4 The Licensee may not amend or alter the Licensed Software and shall have no right or license other than that stipulated herein.
- 4.5 Except as specifically permitted elsewhere in this Agreement the Licensee may make not more than two copies of the Licensed Software (but not the Authoring and Language Manuals) in machine-readable form for operational security and shall ensure that all such copies include Aeroflex's copyright notice, together with any features which disclose the name of the Licensed Software and the Licensee. Furthermore, the Licensee shall not permit the Licensed Software or any part to be disclosed in any form to any third party and shall maintain the Licensed Software in secure premises to prevent any unauthorized disclosure. The Licensee shall notify Aeroflex immediately if the Licensee has knowledge that any unlicensed party possesses the Licensed Software. The Licensee's obligation to maintain confidentiality shall cease when the Licensed Software and all copies have been destroyed or returned. The copyright in the Licensed Software shall remain with Aeroflex. The Licensee will permit Aeroflex at all reasonable times to audit the use of the Licensed Software.
- 4.6 The Licensee will not disassemble or reverse engineer the Licensed Software, nor sub-license, lease, rent or part with possession or otherwise transfer the whole or any part of the Licensed Software.

5 ADDITIONAL LICENSE RIGHTS SPECIFIC TO PXI SOFTWARE**5.1 Definitions for PXI Software**

The following expressions will have the meanings set out below for the purposes of the supplementary rights granted in this Article.

PXI Drivers	All 3000 Series PXI module device drivers including embedded firmware that are installed at runtime
PXI Executable Applications	All executable applications supplied with each 3000 Series PXI module including:- PXI Studio Soft Front Panels (manual operation graphical user interfaces) Utilities including: RF Investigator, PXI Version Information and Self Test
PXI Spectrum Analysis Library	The spectrum analysis measurement suite library .dll software supplied with each 3000 Series PXI module
PXI Optional Application Library	Individual measurement suite available from a range of optional .dll application libraries

5.2 PXI Drivers, PXI Executable Applications and PXI Spectrum Analysis Library License Rights

Subject to the License granted in Article 4 hereof notwithstanding the limitations on number of copies in Clause 4.5 hereof, the Licensee is entitled to make and distribute as many copies of the PXI Drivers and PXI Executable Applications as necessary for use with 3000 Series PXI modules acquired by the Licensee from Aeroflex or its authorized distributor or reseller provided that the Licensee may not sell or charge a fee for the PXI Drivers and PXI Executable Applications.

5.3 PXI Optional Application Library License Rights

Subject to the License granted in Article 4 hereof notwithstanding the limitations on number of copies in Clause 4.5 hereof, the Licensee is entitled to distribute as many copies of any PXI Optional Application Library as necessary for use with 3000 Series PXI modules acquired by the Licensee from Aeroflex or its authorized distributor or reseller provided that:

5.3.1 copies of the applicable PXI Optional Application Library are used solely with 3000 Series PXI modules which the customer has purchased with the corresponding option or part number for the applicable PXI Optional Application Library; and

5.3.2 the Licensee may not sell or charge a fee for the PXI Optional Application Library.

6 WARRANTY

6.1 Aeroflex certifies that the Licensed Software supplied by Aeroflex will at the time of delivery function substantially in accordance with the applicable Software Product Descriptions, Data Sheets or Product Specifications published by Aeroflex.

6.2 The warranty period (unless an extended warranty for Embedded Software has been purchased) from date of delivery in respect of each type of Licensed Software is:

PXI Drivers	24 months
Embedded Software	12 months
Add-In Application Software	90 days
Computer Application Software	90 days
Downloaded Software	No warranty

6.3 If during the appropriate Warranty Period the Licensed Software does not conform substantially to the Software Product Descriptions, Data Sheets or Product Specifications Aeroflex will provide:

6.3.1 In the case of Embedded Software and at Aeroflex's discretion either a fix for the problem or an effective and efficient work-around.

6.3.2 In the case of Add-In Application Software and Computer Application Software and at Aeroflex's discretion replacement of the software or a fix for the problem or an effective and efficient work-around.

6.4 Aeroflex does not warrant that the operation of any Licensed Software will be uninterrupted or error free.

6.5 The above Warranty does not apply to:

6.5.1 Defects resulting from software not supplied by Aeroflex, from unauthorized modification or misuse or from operation outside of the specification.

6.5.2 Third party produced proprietary software which Aeroflex may deliver with its products, in such case the third party software license agreement including its warranty terms shall apply.

6.6 The remedies offered above are sole and exclusive remedies and to the extent permitted by applicable law are in lieu of any implied conditions, guarantees or warranties whatsoever and whether statutory or otherwise as to the Licensed Software all of which are hereby expressly excluded.

7. INDEMNITY

7.1 Aeroflex shall defend, at its expense, any action brought against the Licensee alleging that the Licensed Software infringes any patent, registered design, trademark or copyright, and shall pay all Licensor's costs and damages finally awarded up to an aggregate equivalent to the License Fee provided the Licensee shall not have done or permitted to be done anything which may have been or become any such infringement and shall have exercised reasonable care in protecting the same failing which the Licensee shall indemnify Aeroflex against all claims costs and damages incurred and that Aeroflex is given prompt written notice of such claim and given information, reasonable assistance and sole authority to defend or settle such claim on behalf of the Licensee. In the defense or settlement of any such claim, Aeroflex may obtain for the Licensee the right to continue using the Licensed Software or replace it or modify it so that it becomes non-infringing.

7.2 Aeroflex shall not be liable if the alleged infringement:

- 7.2.1 is based upon the use of the Licensed Software in combination with other software not furnished by Aeroflex, or
 - 7.2.2 is based upon the use of the Licensed Software alone or in combination with other software in equipment not functionally identical to the Designated Equipment, or
 - 7.2.3 arises as a result of Aeroflex having followed a properly authorized design or instruction of the Licensee, or
 - 7.2.4 arises out of the use of the Licensed Software in a country other than the one disclosed to Aeroflex as the intended country of use of the Licensed Software at the commencement of this Agreement.
- 7.3 Aeroflex shall not be liable to the Licensee for any loss of use or for loss of profits or of contracts arising directly or indirectly out of any such infringement of patent, registered design, trademark or copyright. Notwithstanding anything in this Agreement to the contrary, the total liability of Aeroflex and its employees, in contract, tort, or otherwise (including negligence, warranty, indemnity, and strict liability) howsoever arising out of this License shall be limited to the total amount of the License Fee and total support fees actually paid to Aeroflex by the Licensee.

8. TERMINATION

- 8.1 Notwithstanding anything herein to the contrary, this License shall forthwith determine if the Licensee:
- 8.1.1 As an individual has a Receiving Order made against him or is adjudicated bankrupt or compounds with creditors or as a corporate body, compounds with creditors or has a winding-up order made against it or
 - 8.1.2 Parts with possession of the Designated Equipment.
- 8.2 This License may be terminated by notice in writing to the Licensee if the Licensee shall be in breach of any of its obligations hereunder and continue in such breach for a period of 21 days after notice thereof has been served on the Licensee.
- 8.3 On termination of this Agreement for any reason, Aeroflex may require the Licensee to return to Aeroflex all copies of the Licensed Software in the custody of the Licensee and the Licensee shall, at its own cost and expense, comply with such requirement within 14 days and shall, at the same time, certify to Aeroflex in writing that all copies of the Licensed Software in whatever form have been obliterated from the Designated Equipment.

9. THIRD PARTY LICENCES

- 9.1 The Licensed Software or part thereof may be the proprietary property of third party licensors. In such an event such third party licensors (as may be referenced on the software media, or any on screen message on start up of the software or on the order acknowledgement) and/or Aeroflex may directly enforce the terms of this Agreement and may terminate the Agreement if the Licensee is in breach of the conditions contained herein.
- 9.2 If any third party software supplied with the Licensed Software is supplied with, or contains or displays the third party's own license terms then the Licensee shall abide by such third party license terms (for the purpose of this Article the term "third party" shall include other companies within the Aeroflex group of companies).

10. EXPORT REGULATIONS

The Licensee undertakes that where necessary the Licensee will conform with all relevant export regulations imposed by the Governments of the United Kingdom and/or the United State of America.

11. U.S. GOVERNMENT RESTRICTED RIGHTS

The Licensed Software and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable.

12. NOTICES

Any notice to be given by the Licensee to Aeroflex shall be addressed to:
Aeroflex Limited, Longacres House, Six Hills Way, Stevenage, SG1 2AN, UK.

13. LAW AND JURISDICTION

This Agreement shall be governed by the laws of England and shall be subject to the exclusive jurisdiction of the English courts. This agreement constitutes the whole agreement between the parties and may be changed only by a written agreement signed by both parties.

**CHINA Beijing**

Tel: [+86] (10) 6539 1166
Fax: [+86] (10) 6539 1778

CHINA Shanghai

Tel: [+86] (21) 5109 5128
Fax: [+86] (21) 5150 6112

CHINA Shenzhen

Tel: [+86] (755) 3301 9358
Fax: [+86] (755) 3301 9356

FINLAND

Tel: [+358] (9) 2709 5541
Fax: [+358] (9) 804 2441

FRANCE

Tel: [+33] 1 60 79 96 00
Fax: [+33] 1 60 77 69 22

GERMANY

Tel: [+49] 8131 2926-0
Fax: [+49] 8131 2926-130

HONG KONG

Tel: [+852] 2832 7988
Fax: [+852] 2834 5364

INDIA

Tel: [+91] 80 [4] 115 4501
Fax: [+91] 80 [4] 115 4502

KOREA

Tel: [+82] (2) 3424 2719
Fax: [+82] (2) 3424 8620

SCANDINAVIA

Tel: [+45] 9614 0045
Fax: [+45] 9614 0047

UK Stevenage

Tel: [+44] (0) 1438 742200
Fax: [+44] (0) 1438 727601
Freephone: 0800 282388

USA

Tel: [+1] (316) 522 4981
Fax: [+1] (316) 522 1360
Toll Free: (800) 835 2352

*As we are always seeking to improve our products, the information in this document gives only a general indication of the product capacity, performance and suitability, none of which shall form part of any contract.
We reserve the right to make design changes without notice.*

web www.aeroflex.com

Email info-test@aeroflex.com

October 2010