Agilent U8903A
Audio Analyzer

Programmer's Reference

Agilent Technologies

Notices

Manual Part Number

U8903-90027

Edition

First Edition, June 5, 2009

Agilent Technologies, Inc.
5301 Stevens Creek Blvd.
Santa Clara, CA 95051 USA

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

Safety Notices

CAUTION

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Contents

1
Remote Interface Reference

This chapter describes how to configure and program the U8903A over a remote interface.

Remote Interface Configuration

This section describes how to configure the GPIB (IEEE-488), LAN, and USB remote interfaces.

NOTE

- For more information on configuring the remote interface connectivity, refer to the *Agilent Technologies USB/LAN/GPIB Interfaces Connectivity Guide*.
- If you have installed the IO Libraries Suite, you can access the Connectivity Guide via the Agilent IO Libraries Control icon. Alternatively, you can access the Connectivity Guide via the Web at www.agilent.com/find/connectivity.

You can choose to control the U8903A remotely using the GPIB, LAN, or USB interfaces.

GPIB address

Each device on the GPIB interface must have a unique address. You can set the U8903A address to any value between 0 and 30. The U8903A is shipped with a default address of 28. The GPIB address is stored in nonvolatile memory, and does not change when the U8903A is switched off, or after a remote interface reset.

The GPIB bus controller has its own address. Avoid using the bus controller address for any instrument on the interface bus. Agilent controllers generally use the address of 21.

Use the following command to set the GPIB address from the remote interface.

```
SYSTem:COMMunicate:GPIB:ADDRess
```

Use the following command to query the GPIB address from the remote interface.

```
SYSTem:COMMunicate:GPIB:ADDRess?
```

LAN configuration

The U8903A supports three LAN operating modes as follows.

- Dynamic IP (Dynamic Host Configuration Protocol or DHCP)
- Auto IP (local PC control or isolated LAN)
- Static IP (manual configuration)

Configuring the LAN remotely

The IP address, subnet mask, and default gateway can be changed manually or remotely. To remotely specify the LAN settings, use the following commands.

- IP Address: `SYSTem:COMMunicate:LAN:ADDRess`
- Subnet Mask: `SYSTem:COMMunicate:LAN:SMASk`
- Default Gateway: `SYSTem:COMMunicate:LAN:DGATeway`

The values for the IP address, subnet mask, and default gateway can range between `0.0.0.0` and `255.255.255.255`.

NOTE If you set an invalid IP address or an IP address that is used by another device or host, an error message is generated. This error can be read by using the `SYSTem:ERRor?` command.

The LAN settings are stored in nonvolatile memory.

USB interface

The USB interface does not require front panel or remote configuration.

The USB address cannot be changed as it is set at the factory and is unique for each U8903A.

NOTE

- Before connecting the USB cable, make sure that the Agilent IO Libraries software is installed on your PC.
- For more information on the Agilent IO Libraries software, refer to the *Agilent Technologies USB/LAN/GPIB Interfaces Connectivity Guide*. If you have installed other I/O software, refer to the documentation that accompanies the software.

Introduction to the SCPI Language

SCPI, also known as the Standard Commands for Programmable Instruments, is an ASCII-based instrument command language designed for test and measurement instruments. SCPI commands are based on a hierarchical structure, also known as a tree system. In this system, associated commands are grouped together under a common node or root, thus forming subsystems. A portion of the Source subsystem is shown below to illustrate the tree system.

```
[SOURce:]
    SWEep:
        MODE <mode>, (@<channel>)
```

`SOURce` is the root keyword of the command, `SWEep` is the second-level keyword, and `MODE` is the third-level keyword. A colon (:) separates a command keyword from a lower-level keyword.

SCPI Conventions and Data Formats

The following SCPI conventions are used throughout this chapter.

Angle brackets < >	Items within angle brackets are parameter abbreviations. The brackets are not sent with the command string.
Vertical bar |	Vertical bars separate alternative parameters.
Square brackets [ ]	Items within square brackets are optional. The representation of `[SOURce:]FUNCtion` means that `SOURce:` may be omitted. The brackets are not sent with the command string.
Parenthesis ( )	Items within parentheses are used to specify a channel list.
Braces { }	Braces enclose the parameter choices for a given command string. The braces are not sent with the command string.

The SCPI language defines several different data formats to be used in program messages and response messages.

Numeric Commands that require parameters to accept all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation. You can also send engineering unit suffixes with numeric parameters such as MHz or kHz.

Discrete Parameters used to program settings that have a limited number of values such as BUS, IMMediate, and EXTernal. They have a short form and a long form just like command keywords. You can mix upper- and lower-case letters. Query responses will always return the short form in all upper-case letters.

Boolean Parameters that represent a single binary condition that is either true or false. For a false condition, the U8903A will accept OFF or 0. For a true condition, the U8903A will accept ON or 1. When you query a boolean setting, the U8903A will always return 0 or 1.

String Parameters that contain virtually any set of ASCII characters. A string must begin and end with matching quotes, either with a single quote or a double quote. You can include the quote delimiter as part of the string by typing it twice without any characters in between.

Block Parameter that allows binary data (including extended ASCII codes) to be transmitted as a sequence of bytes. This is more efficient than the text format when transferring large amounts of data. Either definite length or indefinite length arbitrary data may be transmitted or returned.

IEEE-488.2 Common Commands

The IEEE-488.2 standard defines a set of common commands that perform functions such as reset, self-test, and status operation. Common commands always begin with an asterisk (*), are three characters in length, and may include one or more parameters. The command keyword is separated from the first parameter by a blank space. Use a semicolon (;) to separate multiple commands as shown below.

```
*RST; *CLS; *ESE 32; *OPC?
```

*CLS

Syntax

```
*CLS
```

Description

Clears the event registers in all register groups and also clears the error queue.

Example

The following command is used to clear all event registers and the error queue.

```
*CLS
```

*ESE

Syntax

```
*ESE <value>

*ESE?
```

Description

Sets the bits in the Standard Event enable register. The selected bits are then reported to bit 5 of the Status Byte register. The query reads the enable register and returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

Parameter

Item	Type	Range of values	Default value
value	Numeric	A decimal value which corresponds to the binary-weighted sum of the bits in the register	0

Remarks

- The bit definitions for the Standard Event register are listed in "Standard Event register" on page 23.
- Use the `<value>` parameter to specify which bits will be enabled. The specified decimal value corresponds to the binary-weighted sum of the bits you wish to enable in the register. For example, to enable bit 2 (decimal value = 4), bit 3 (decimal value = 8), and bit 7 (decimal value = 128), the corresponding decimal value would be 140 (4 + 8 + 128).
- The clear status (`*CLS`) command will not clear the enable register but it clears all bits in the event register.
- The `*RST` or `SYSTem:PRESet` command does not affect the settings enabled by this command. However, cycling the U8903A power will reset this register to `0`.

Examples

The following command enables bit 4 (decimal value = 16) in the enable register. If an Execution Error occurs, this condition will be reported to the Status Byte register (bit 5 will be set to high).

```
*ESE 16
```

The following query returns the bits set in the register.

```
*ESE?
```

Typical Response: `16`

*ESR?

Syntax

```
*ESR?
```

Description

Reads the event register of the Standard Event register group and returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

Remarks

- The bit definitions for the Standard Event register are listed in "Standard Event register" on page 23.
- Once a bit is set, it remains set until cleared by a clear status (`*CLS`) command or queried by this command.

Example

The following query reads the event register (bits 3 and 4 are set).

```
*ESR?
```

Typical Response: `24`

*IDN?

Syntax

```
*IDN?
```

Description

Reads the U8903A identification string which contains four comma-separated fields. The first field is the manufacturer's name, the second field is the instrument model number, the third field is the serial number, and the fourth field is the firmware revision. This query returns an ASCII string with the following format.

<Manufacturer's name>,<model number>,<serial number>, <firmware revision>

Agilent Technologies	Manufacturer
U8903A	Instrument model number
MYxxxxxxxx	Instrument serial number if available, or 0
x.x.x.x	Firmware revision levels

Example

The following query returns the U8903A identification string.

```
*IDN?
```

Typical Response:

```
AGILENT TECHNOLOGIES,U8903A,MY00123456,1.0.0.0
```

*OPC

Syntax

```
*OPC

*OPC?
```

Description

Sets the “Operation Complete” bit (bit 0) in the Standard Event register when all pending operations have completed. This query sends `1` to the output buffer when all pending operations have completed.

Remark

This command is used to synchronize your application with the U8903A.

Examples

The following command sets the "Operation Complete" bit.

```
*OPC
```

The following query waits until the completion of the current command and then sends `1` to the output buffer.

```
*OPC?
```

Typical Response: `1`

*RST

Syntax

```
*RST
```

Description

Resets the U8903A to its factory default settings.

Remarks

- This command does not affect any user-defined files in the U8903A memory.
- The time taken to reset all settings for all channels of the U8903A is approximately 1 s. If you only wish to reset the settings for a particular channel, use the `SYSTem:RESet:CHANnel` command instead. To reset the settings for only a particular mode such as the generator mode, use the `SYSTem:RESet[:MODE]` command.

Example

The following command resets all settings for all channels of the U8903A to its factory default settings.

```
*RST
```

*SRE

Syntax

```
*SRE <value>
```

```
*SRE?
```

Description

Enables the bits in the Status Byte enable register. The selected enabled bits are summarized in the “Master Summary” bit (bit 6) of the Status Byte register. If any of the selected bit condition changes from 0 to 1, a Service Request is generated. The query reads the enable register and returns a decimal value that corresponds to the binary-weighted sum of all bits set in the register.

Parameter

Item	Type	Range of values	Default value
value	Numeric	A decimal value which corresponds to the binary-weighted sum of the bits in the register	0

Remarks

- The bit definitions for the Status Byte register are listed in "Status Byte register" on page 24.
- Use the `<value>` parameter to specify which bits to enable. The specified decimal value corresponds to the binary-weighted sum of the bits you wish to enable in the register. For example, to enable bit 2 (decimal value = 4) and bit 5 (decimal value = 32), the corresponding decimal value would be 36 (4 + 32).
- The `STATus:PRESet`, `SYSTem:PRESet`, `*CLS`, or `*RST` command does not clear the bits in the Status Byte enable register.
- Cycling the U8903A power will reset it to `0`.

Examples

The following command enables bit 4 (decimal value = 16) in the enable register.

```
*SRE 16
```

The following query returns which bits are enabled in the register.

```
*SRE?
```

Typical Response: `16`

*STB?

Syntax

```
*STB?
```

Description

Queries the condition register for the Status Byte register and returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register. This query is similar to a Serial Poll but it is processed like any other instrument command. This is a read-only register and the bits are not cleared when you read the register.

Remarks

- The bit definitions for the Status Byte register are listed in "Status Byte register" on page 24.
- This query returns the same results as a Serial Poll but the "Master Summary" bit (bit 6) is not cleared if a Serial Poll has occurred.
- A power-on cycle will clear all bits in the condition register.

Example

The following query reads the condition register (bits 2 and 5 are set).

```
*STB?
```

Typical Response: `36`

*TRG

Syntax

```
*TRG
```

Description

This command is used in conjunction with the `TRIGger:ANALyzer:SOURce` or `TRIGger:GRAPh:SOURce` command to trigger the U8903A from the remote interface.

Remarks

- Use the `TRIGger:ANALyzer:SOURce` command in the analyzer mode or `TRIGger:GRAPh:SOURce` command in the graph mode to select the BUS trigger source.
- After setting the trigger source, the U8903A must be set in the "wait-for-trigger" state using the `INITiate[:IMMediate]:ANALyzer` command in the analyzer mode. For the graph mode, use the `INITiate[:IMMediate]:GRAPh` command. The `*TRG` command will not be accepted unless the U8903A is in the "wait-for-trigger" state.

Example

The following command sequence is used to trigger the U8903A in the analyzer mode.

```
TRIG:ANAL:SOUR BUS

INIT:ANAL

*TRG
```

*TST?

Syntax

```
*TST?
```

Description

Initiates an internal self-test of the U8903A and returns a pass or fail indication. The self-test runs a series of tests and will take approximately 30 s to complete.

Remarks

- If one or more tests fail, `1` is returned and the errors are stored in the error queue. For a complete listing of the error messages related to self-test failures, refer to Chapter 3, "Error Messages" on page 230. Use the `SYSTem:ERRor?` command to read the error queue.
- If all tests pass, `0` is returned.

NOTE Do not operate the U8903A while the self-test is in progress as doing so might cause unexpected results.

Example

The following query performs a self-test and returns a pass or fail indication.

```
*TST?
```

Typical Response: `0`

*WAI

Syntax

```
*WAI
```

Description

The Wait-to-Continue (WAI) command causes the U8903A to wait until all pending operations have completed, before executing any other command.

Example

The following command waits until all pending operations have completed.

```
*WAI
```

SCPI Status System

This section describes the structure of the SCPI status system used by the U8903A. Each register group is made up of several low-level registers called Condition register, Event register, and Enable register which control the action of specific bits within the register group.

Condition register

A condition register continuously monitors the state of the U8903A. The bits in the condition register are updated in realtime and the bits are not latched or buffered. This is a read-only register and the bits are not cleared when you read the register. A query of the condition register returns a decimal value which corresponds to the binary-weighted sum of all bits set in that register.

Event register

An event register latches the various events from the changes in the condition register. There is no buffering in this register; while an event bit is set, subsequent events corresponding to that bit are ignored. This is a read-only register. Once a bit is set, it remains set until cleared by a query or clear status (`*CLS`) command. A query of this register returns a decimal value which corresponds to the binary-weighted sum of all bits set in that register.

Enable register

An enable register defines which bits in the event register will be reported to the Status Byte register group. You can write to or read from an enable register. A clear status (`*CLS`) command will not clear the enable register but it clears all bits in the event register. To enable bits in the enable register to be reported to the Status Byte register, you must write a decimal value which corresponds to the binary-weighted sum of the corresponding bits.

Status system diagram

The U8903A uses the Operation, Questionable, Standard Event, and Status Byte register groups to record a variety of instrument conditions. The relationship between various registers in the U8903A SCPI status system is shown below.

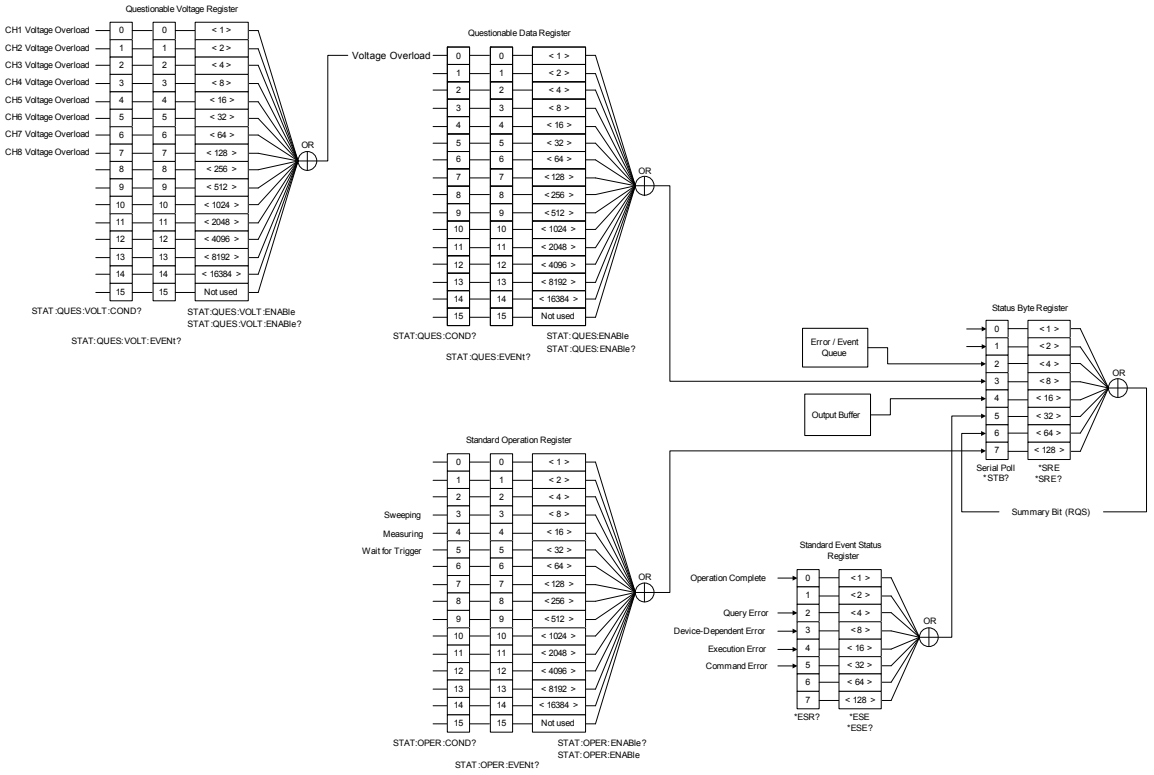


Figure 1-1 Status system diagram

Standard Operation register

This register record signals that occur during normal operation. The outputs of the Standard Operation register are logically-ORed into the OPERation summary bit (7) of the Status Byte register.

Bit definitions: Standard Operation register

Bit number	Decimal value	Definition
0 to 2 Not Used	Not Used	0 is returned
3 Sweeping in Progress	8	The U8903A is performing sweep
4 Measuring in Progress	16	The U8903A is initiated, and is making, or about to make a measurement
5 Waiting for Trigger	32	The U8903A is waiting for an external or bus trigger
6 to 15 Not Used	Not Used	0 is returned

The `STATus:PRESet` command will clear all bits in the NTR and enable registers.

Questionable Status registers

These registers record signals that indicate abnormal operation. The Questionable Data and Questionable Voltage registers are used for the U8903A. The outputs of the Questionable Voltage register are logically-ORed into the Voltage Overload bit (0) of the Questionable Data register. The outputs of the Questionable Data register are logically-ORed into the QUEStionable summary bit (3) of the Status Byte register.

Bit definitions: Questionable Data register

Bit number	Decimal value	Definition
0 Voltage Overload	1	The voltage of one of the input signals is over the limit
1 to 15 Not Used	Not Used	0 is returned

Bit definitions: Questionable Voltage register

Bit number	Decimal value	Definition
0 Channel 1 Voltage Overload	1	The voltage of channel 1 is over the limit
1 Channel 2 Voltage Overload	2	The voltage of channel 2 is over the limit
2 Channel 3 Voltage Overload	4	The voltage of channel 3 is over the limit
3 Channel 4 Voltage Overload	8	The voltage of channel 4 is over the limit
4 Channel 5 Voltage Overload	16	The voltage of channel 5 is over the limit
5 Channel 6 Voltage Overload	32	The voltage of channel 6 is over the limit
6 Channel 7 Voltage Overload	64	The voltage of channel 7 is over the limit
7 Channel 8 Voltage Overload	128	The voltage of channel 8 is over the limit
8 to 15 Not Used	Not Used	0 is returned

Standard Event register

The Standard Event register reports the following types of instrument events: command syntax errors, command execution errors, device errors (self-test or calibration), query errors, or when an *OPC command is executed. All of these conditions can be reported in the Standard Event summary bit through the enable register.

Bit definitions: Standard Event register

Bit number	Decimal value	Definition
0 Operation Complete	1	All commands prior to and including *OPC have been executed
1 Not Used	Not Used	0 is returned
2 Query Error	4	The U8903A tried to read the output buffer but it was empty. Or, a new command line was received before a previous query has been read. Or, both the input and output buffers are full.
3 Device-Dependent Error	8	A self-test, calibration, or other device-specific error has occurred
4 Execution Error	16	A command execution error occurred
5 Command Error	32	A command syntax error occurred
6 to 7 Not Used	Not Used	0 is returned

The event register in the Standard Event is cleared when:

- you execute the clear status (*CLS) command
- querying the event register using the event status register (*ESR?) command

The Standard Event enable register is cleared when you execute the *ESE 0 command.

Status Byte register

The Status Byte register reports the conditions from the other status registers. Clearing an event register from one of the other registers will clear the corresponding bits in the Status Byte condition register.

Data that is waiting in the U8903A output buffer is immediately reported on the "Message Available" bit (bit 4).

Bit definitions: Status Byte register

Bit number	Decimal value	Definition
0 to 1 Not Used	Not Used	0 is returned
2 Error Queue	4	There is at least one error message in the error queue
3 Questionable Data Summary	8	One or more bits are set in the Questionable Data register (bits must be enabled in the enable register)
4 Message Available	16	Data is available in the U8903A output buffer
5 Event Status Byte Summary	32	One or more bits are set in the Standard Event register (bits must be enabled in the enable register)
6 Master Status Summary (Request for Service)	64	One or more bits are set in the Status Byte register (bits must be enabled in the enable register). Also used to indicate a request for service.
7 Standard Operation Summary	128	One or more bits are set in the Standard Operation register (bits must be enabled in the enable register)

The Status Byte condition register will be cleared when:

- you execute the clear status (`*CLS`) command
- you read the event register from one of the other registers, only the corresponding bits are cleared in the condition register

The Status Byte enable register is cleared when you execute the `*SRE 0` command.

System Subsystem

This subsystem is used to set up the system configuration.

SYSTem:ERRor[:NEXT]?

Syntax

```
SYSTem:ERRor[:NEXT]?
```

Description

Returns the error number and its corresponding message string from the U8903A error queue. A record of up to 30 errors can be stored in the U8903A error queue. The format of the response is:

```
<error number>,"<error string>"
```

where the error number is defined in Chapter 3, "Error Messages" on page 230.

Remarks

- Errors are retrieved in the first-in, first-out (FIFO) order where the first error returned is the first error that has been stored.
- If more than 30 errors have occurred, the last error stored in the queue (the most recent error) is replaced with **`-350,"Queue overflow"`**. No additional errors are stored until you remove errors from the queue. If no error occur when you read the error queue, the U8903A responds with **`0,"No error"`**.
- The error queues are cleared by the clear status (`*CLS`) command and when power is cycled. The errors are also cleared when you read the error queue. The error queue is not cleared by a factory reset (`*RST`) or `SYSTem:PRESet` command.

- The command reads and clears one error string from the error queue. The error string may contain up to 255 characters and consists of an error number and an error string enclosed in double quotes. For example:

  `-113,"Undefined header"`

Example

The following query reads and clears one error.

```
SYST:ERR?
```

Typical Response: `-101,"Invalid character"`

SYSTem:DATE

Syntax

```
SYSTem:DATE <yyyy>, <mm>, <dd>
```

```
SYSTem:DATE?
```

Description

Sets the date of the realtime clock in year (yyyy), month (mm), and day (dd) format. The query returns comma-separated values that correspond to the year, month, and day.

Parameters

Item	Type	Range of values	Default value
yyyy	Numeric	A 4-digit integer representing the year. The value is within the range of 2000 to 2099.	Required parameter
mm	Numeric	An integer from 1 to 12	Required parameter
dd	Numeric	An integer from 1 to 31	Required parameter

Examples

The following command sets the date (April 1, 2008).

```
SYST:DATE 2008, 4, 1
```

The following query returns the date.

```
SYST:DATE?
```

Typical Response: `2008,4,1`

SYSTem:TIME

Syntax

```
SYSTem:TIME <hh>, <mm>, <ss>
```

```
SYSTem:TIME?
```

Description

Sets the realtime clock in hours (hh), minutes (mm), and seconds (ss). The query returns comma-separated values that correspond to the hour, minute, and seconds.

Parameters

Item	Type	Range of values	Default value
hh	Numeric	An integer from 0 to 23	Required parameter
mm	Numeric	An integer from 0 to 59	Required parameter
ss	Numeric	An integer from 0 to 59	Required parameter

Examples

The following command sets the time.

```
SYST:TIME 13, 30, 10
```

The following query returns the time.

```
SYST:TIME?
```

Typical Response: `13,30,10`

SYSTem:VERSion?

Syntax

```
SYSTem:VERSion?
```

Description

Returns the SCPI standard version with which the U8903A is in compliance. The U8903A complies with the rules and conventions of the indicated SCPI standard version. The response format is in the form of XXXX.Y, where XXXX represents the year of the version and Y represents the version number for that year.

Example

The following query returns the SCPI version.

```
SYST:VERS?
```

Typical Response: `1999.0`

SYSTem:COMMunicate:GPIB[:SELF]:ADDRess

Syntax

```
SYSTem:COMMunicate:GPIB[:SELF]:ADDRess <address>
SYSTem:COMMunicate:GPIB[:SELF]:ADDRess?
```

Description

Assigns the U8903A GPIB (IEEE-488) address. Each device on the GPIB interface must have a unique address.

Parameter

Item	Type	Range of values	Default value
address	Numeric	0 to 30	28

Remarks

- The factory GPIB address setting is 28.
- Your PC GPIB interface card has its own address. Avoid using the PC address for any instrument on the interface bus.
- The GPIB address is stored in nonvolatile memory, and does not change when power has been turned off, after a factory reset (`*RST` command), or after an instrument preset (`SYSTem:PRESet` command).

Examples

The following command sets the GPIB address to 28.

```
SYST:COMM:GPIB:ADDR 28
```

The following query returns the GPIB address.

```
SYST:COMM:GPIB:ADDR?
```

Typical Response: `28`

SYSTem:COMMunicate:LAN:DHCP:ENABled

Syntax

```
SYSTem:COMMunicate:LAN:DHCP:ENABled
```

Description

Enables the Dynamic Host Configuration Protocol (DHCP) for the U8903A. When the DHCP is enabled (factory setting), the U8903A will try to obtain an IP address from a DHCP server. If a DHCP server is found, it will assign a dynamic IP address, subnet mask, and default gateway to the U8903A.

Example

The following command enables the DHCP.

```
SYST:COMM:LAN:DHCP:ENAB
```

SYSTem:COMMunicate:LAN:ADDRess

Syntax

```
SYSTem:COMMunicate:LAN:ADDRess <address>
```

```
SYSTem:COMMunicate:LAN:ADDRess?
```

Description

Assigns a static Internet Protocol (IP) address for the U8903A. The query returns the IP address in the form of `"A.B.C.D"`.

Parameter

Item	Type	Range of values	Default value
address	String	Up to 15 characters formatted as A.B.C.D where A, B, C, and D is within the range of 0 to 255 each (no embedded spaces)	Required parameter

Remarks

- Sending this command will automatically disable the DHCP and switch to static IP.
- The IP address is stored in nonvolatile memory, and does not change when power has been turned off, after a factory reset (`*RST` command), or after an instrument preset (`SYSTem:PRESet` command).

Examples

The following command sets the IP address.

```
SYST:COMM:LAN:ADDR "169.254.149.35"
```

The following query returns the IP address in double quotes.

```
SYST:COMM:LAN:ADDR?
```

Typical Response: `"169.254.149.35"`

SYSTem:COMMunicate:LAN:SMASk

Syntax

```
SYSTem:COMMunicate:LAN:SMASk <subnet mask>
```

```
SYSTem:COMMunicate:LAN:SMASk?
```

Description

Sets the static subnet mask address. The query returns the subnet mask address in the form of `"A.B.C.D"`.

Parameter

Item	Type	Range of values	Default value
subnet mask	String	Up to 15 characters formatted as A.B.C.D where A, B, C, and D is within the range of 0 to 255 each (no embedded spaces)	Required parameter

Remarks

- Sending this command will automatically disable the DHCP and switch to static subnet mask.
- The subnet mask address is stored in nonvolatile memory, and does not change when power has been turned off, after a factory reset (`*RST` command), or after an instrument preset (`SYSTem:PRESet` command).

Examples

The following command sets the subnet mask.

```
SYST:COMM:LAN:SMAS "255.255.20.11"
```

The following query returns the subnet mask address in double quotes.

```
SYST:COMM:LAN:SMAS?
```

Typical Response: `"255.255.20.11"`

SYSTem:COMMunicate:LAN:DGATeway

Syntax

```
SYSTem:COMMunicate:LAN:DGATeway <gateway>
```

```
SYSTem:COMMunicate:LAN:DGATeway?
```

Description

Assigns the static default gateway address. The query returns the default gateway address in the form of `"A.B.C.D"`.

Parameter

Item	Type	Range of values	Default value
gateway	String	Up to 15 characters formatted as A.B.C.D where A, B, C, and D is within the range of 0 to 255 each (no embedded spaces)	Required parameter

Remarks

- Sending this command will automatically disable the DHCP and switch to static default gateway.
- The default gateway address is stored in nonvolatile memory, and does not change when power has been turned off, after a factory reset (`*RST` command), or after an instrument preset (`SYSTem:PRESet` command).

Examples

The following command sets the default gateway.

```
SYST:COMM:LAN:DGAT "255.255.20.11"
```

The following query returns the default gateway address in double quotes.

```
SYST:COMM:LAN:DGAT?
```

Typical Response: `"255.255.20.11"`

SYSTem:COMMunicate:LAN:HNAMe?

Syntax

```
SYSTem:COMMunicate:LAN:HNAMe?
```

Description

Queries the LAN hostname and returns an ASCII string enclosed in double quotes.

Example

The following query returns the hostname of the U8903A in double quotes.

```
SYST:COMM:LAN:HNAM?
```

Typical Response: `"U8903A"`

SYSTem:COMMunicate:LAN:MAC?

Syntax

```
SYSTem:COMMunicate:LAN:MAC?
```

Description

Reads the U8903A Media Access Control (MAC) address, also known as either the link-layer address, Ethernet (station) address, LANIC ID, or hardware address. This is an unchangeable 48-bit address assigned by the manufacturer to each unique Internet device. The query returns an ASCII string enclosed in double quotes. The MAC address is represented as 12 hexadecimal characters.

NOTE Your network administrator may need the MAC address if they are assigning a static IP address for this device.

Remarks

- The U8903A MAC address is set at the factory and cannot be changed.
- The MAC address is stored in nonvolatile memory, and does not change when power has been turned off, after a factory reset (`*RST` command), or after an instrument preset (`SYSTem:PRESet` command).

Example

The following query returns the MAC address in double quotes.

```
SYST:COMM:LAN:MAC?
```

Typical Response: `"0003D3041075"`

SYSTem:CHANnel?

Syntax

```
SYSTem:CHANnel?
```

Description

Queries the available channels in the U8903A to determine if the channel hardware card is available or in good condition. This query returns comma-separated channel numbers of the available channels in the U8903A.

Remark

If a hardware card is available but in bad condition, this query will not return the channel number for that particular channel.

Example

The following query returns the channel numbers of the available channels which are in good condition.

```
SYST:CHAN?
```

Typical response: `1,2`

SYSTem:PRESet

Syntax

```
SYSTem:PRESet
```

Description

Presets the U8903A to its factory default settings and deletes all user-defined files.

Example

The following command presets the U8903A.

```
SYST:PRES
```

SYSTem:RESet[:MODE]

Syntax

```
SYSTem:RESet[:MODE] <system mode>
```

Description

Resets the customized settings of the specified U8903A system mode to the default settings.

Parameter

Item	Type	Range of values	Default value
system mode	Discrete	AANalyzer, AGENerator, SWEep, or GRAPh	Required parameter

Remarks

- This command resets the customized settings of the selected system mode excluding the stored files, I/O configuration, and common system settings.
- For the analyzer mode, the measurement bandwidth, measurement time, and trigger source will also be reset to the default settings.

Example

The following command resets the analyzer mode.

```
SYST:RES AAN
```

SYSTem:RESet:CHANnel

Syntax

```
SYSTem:RESet:CHANnel <system mode>, (@<channel>)
```

Description

Resets the customized settings of the U8903A system mode for the specified channel to the default settings.

Parameters

Item	Type	Range of values	Default value
system mode	Discrete	AANalyzer, AGENerator, or SWEep	Required parameter
channel	Numeric	1 or 2	Required parameter

Remarks

- This command resets the customized settings of the system mode for the selected channel excluding the stored files, I/O configuration, and common system settings.
- For the analyzer mode, the measurement bandwidth, measurement time, and trigger source will not be reset to the default settings.

Example

The following command resets channel 1 of the analyzer mode.

```
SYST:RES:CHAN AAN, (@1)
```

Output Subsystem

The Output subsystem provides the commands to program the U8903A generator output configuration.

OUTPut:TYPE

Syntax

```
OUTPut:TYPE <type>, (@<channel list>)

OUTPut:TYPE? (@<channel list>)
```

Description

Sets the generator output connection for the specified channel(s). The query returns the output connection type of the selected channel(s). Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
type	Discrete	BALanced, UNBalanced, or COMMon	UNBalanced
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remark

You are required to reconfigure the output impedance each time you change the output connection.

Examples

The following commands set the generator outputs for channels 1 and 2 to Unbalanced and Balanced respectively.

```
OUTP:TYPE UNB, (@1)
OUTP:TYPE BAL, (@2)
```

The following query returns the output connection types of channels 1 and 2.

```
OUTP:TYPE? (@1,2)
```

Typical Response: `UNB,BAL`

OUTPut:IMPedance

Syntax

```
OUTPut:IMPedance <impedance>, (@<channel list>)
```

```
OUTPut:IMPedance? (@<channel list>)
```

Description

Sets the generator output impedance for the specified channel(s). The query returns the output impedance of the selected channel(s). Multiple responses are separated by commas.

The output impedance selection is described as follows.

`IMP50`	Output impedance is 50 Ω for the Unbalanced output connection
`IMP100`	Output impedance is 100 Ω for the Balanced or Common output connection
`IMP600`	Output impedance is 600 Ω for the UnBalanced, Balanced, or Common output connection

Parameters

Item	Type	Range of values	Default value
impedance	Discrete	• IMP100 or IMP600 (Balanced or Common output connection) • IMP50 or IMP600 (Unbalanced output connection)	IMP600
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remark

You must set the output connection type before configuring the output impedance.

Examples

The following commands set the generator output impedance for channels 1 and 2 to 50 Ω and 100 Ω respectively. Assume that the output connection for channel 1 has been set to Unbalanced, and channel 2 to Balanced.

```
OUTP:IMP IMP50, (@1)
```

```
OUTP:IMP IMP100, (@2)
```

The following query returns the output impedance of channels 1 and 2.

```
OUTP:IMP? (@1,2)
```

Typical Response: `IMP50,IMP100`

OUTPut:STATe

Syntax

```
OUTPut:STATe <state>, (@<channel list>)

OUTPut:STATe? (@<channel list>)
```

Description

Enables or disables the generator output for the specified channel(s). The query returns the output state of the selected channel(s) as `0` if the output state is `OFF`, or `1` if the output state is `ON`. Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
state	Boolean	OFF(0) or ON(1)	OFF
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remark

You must configure the output connection and impedance before setting the output state.

Examples

The following commands enable the channel 1 generator output but disable the output for channel 2.

```
OUTP:STAT ON, (@1)

OUTP:STAT OFF, (@2)
```

The following query returns the output states of channels 1 and 2.

```
OUTP:STAT? (@1,2)
```

Typical Response: `1,0`

Input Subsystem

The Input subsystem provides the commands to program the U8903A analyzer input configuration.

INPut:TYPE

Syntax

```
INPut:TYPE <type>, (@<channel list>)
```

```
INPut:TYPE? (@<channel list>)
```

Description

Sets the input connection for the specified channel(s). The query returns the input connection type of the selected channel(s). Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
type	Discrete	BALanced or UNBalanced	UNBalanced
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Examples

The following commands set the analyzer input connections for channels 1 and 2 to Unbalanced and Balanced respectively.

```
INP:TYPE UNB, (@1)
```

```
INP:TYPE BAL, (@2)
```

The following query returns the input connection types of channels 1 and 2.

```
INP:TYPE? (@1,2)
```

Typical Response: `UNB,BAL`

INPut:COUPling

Syntax

```
INPut:COUPling <coupling>, (@<channel list>)
```

```
INPut:COUPling? (@<channel list>)
```

Description

Sets the analyzer AC or DC coupling for the specified channel(s). The DC coupling allows both AC and DC input signals to pass through to the measurement circuitry. The AC coupling blocks the DC component of the input signal. The query returns the input coupling type of the selected channel(s). Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
coupling	Discrete	AC or DC	AC
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

Ensure that the measurement function is not set to VDC (`SENSe:FUNCtion` command) when you are trying to set the AC coupling. Else, the measurement function will be automatically changed to default.

Examples

The following commands set the input coupling types for channels 1 and 2 to AC and DC respectively.

```
INP:COUP AC, (@1)
INP:COUP DC, (@2)
```

The following query returns the input coupling types of channels 1 and 2.

```
INP:TYPE? (@1,2)
```

Typical Response: `AC,DC`

INPut:BANDwidth

Syntax

```
INPut:BANDwidth <bandwidth>
INPut:BANDwidth?
```

Description

Sets the analyzer measurement bandwidth. The query returns the measurement bandwidth type.

Parameter

Item	Type	Range of values	Default value
bandwidth	Discrete	HIGH or LOW • HIGH: 312.5 kHz • LOW: 78.125 kHz	HIGH

Remark

The selected measurement bandwidth applies to all channels.

Examples

The following command sets the High measurement bandwidth.

```
INP:BAND HIGH
```

The following query returns the measurement bandwidth type.

```
INP:BAND?
```

Typical Response: `HIGH`

Source Subsystem

The Source subsystem provides the commands to select the waveform type and configure the generator parameters.

SOURce:FUNCtion

Syntax

```
SOURce:FUNCtion <waveform type>, (@<channel list>)
SOURce:FUNCtion? (@<channel list>)
```

Description

Sets the generator waveform type for the specified channel(s). The query returns the waveform type of the selected channel(s). Multiple responses are separated by commas.

The waveform types with their corresponding `<waveform type>` parameters are listed as follows.

`SINE`	Sine waveform
`VPHase`	Variable phase waveform
`DUAL`	Dual waveform
`SMPTe11`	SMPTE IMD 1 to 1 waveform
`SMPTe41`	SMPTE IMD 4 to 1 waveform
`SMPTe101`	SMPTE IMD 10 to 1 waveform
`DFDiec118`	DFD IEC 60118 waveform
`DFDiec268`	DFD IEC 60268 waveform
`WGAussian`	Gaussian PDF white noise signal
`WRECtangular`	Rectangular PDF white noise signal
`DC`	DC signal
`MULTitone`	Multitone waveform
`SQUare`	Square waveform
`ARBitrary`	Arbitrary waveform

Parameters

Item	Type	Range of values	Default value
waveform type	Discrete	SINE, VPHase, DUAL, SMPTe11, SMPTe41, SMPTe101, DFDiec118, DFDiec268, WGAussian, WRECtangular, DC, MULTitone, SQUare, or ARBitrary	SINE
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- If you change the waveform type, the particular parameter values of the previous waveform will be set to the default values of the current waveform.
- Selecting the variable phase waveform on a selected channel will change the waveform type for all channels to variable phase.
- Refer to "Appendix C: Waveform Parameters" on page 241 for the configurable parameters of the corresponding waveform types.

Examples

The following commands set the waveform types for channels 1 and 2 to Sine and Square respectively.

```
SOUR:FUNC SINE, (@1)
SOUR:FUNC SQU, (@2)
```

The following query returns the waveform types of channels 1 and 2.

```
SOUR:FUNC? (@1,2)
```

Typical Response: `SINE,SQU`

SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet

Syntax

```
SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet
<voltage>[<unit>], (@<channel list>)

SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet?
(@<channel list>)
```

Description

Sets the signal DC offset level in V for the specified channel(s). The query returns the DC offset of the selected channel(s) in V. Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
voltage	Numeric	±11.3 V	0 V
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- The DC offset is not applicable for the square, DC, and variable phase waveform types, and will be automatically set to 0 when you change the waveform type to either DC, square, or variable phase using the `SOURce:FUNCtion` command.
- The DC offset is dependent on the amplitude of the signal to be generated for a particular channel. When the DC offset and amplitude are added together, it must not exceed the maximum voltage. The relationship between the amplitude in Vp and DC offset is as follows.

  $$Vp \leq Vmax - |Voffset|$$

  where Vmax is the maximum voltage of the output connector. For the Balanced output connection, the maximum voltage is 22.6 Vp, while for Unbalanced and Common, the maximum voltage is 11.3 Vp.

- If the specified DC offset is invalid, the generator will automatically adjust it to the maximum DC offset allowed with the specified amplitude. The **–222,"Data out of range**" error will be generated and the DC offset will be adjusted as described.
- You can also include a multiplier for the unit, for example, mV. The 'm' is the multiplier for the unit V.

Examples

The following commands set the DC offset for channels 1 and 2 to 1 V and 3.1 V respectively.

```
SOUR:VOLT:OFFS 1, (@1)
```

```
SOUR:VOLT:OFFS 3.1, (@2)
```

The following query returns the DC offset values of channels 1 and 2 in V.

```
SOUR:VOLT:OFFS? (@1,2)
```

Typical Response: `1.000000E+00,3.100000E+00`

SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]

Syntax

```
SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]
<voltage>[<unit>], (@<channel list>)
```

```
SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]?
(@<channel list>)
```

Description

Sets the signal amplitude level for the specified channel(s). The query returns the amplitude of the selected channel(s) in Vrms. Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
voltage	Numeric	Refer to "Appendix D: Waveform Amplitude Range" on page 243	0 Vrms
unit	Discrete	• V (for the DC signal) • Vrms, Vpp, Vp, or dBV (for other waveform types)	Vrms
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- The DC offset is dependent on the amplitude of the signal to be generated for a particular channel. When the DC offset and amplitude are added together, it must not exceed the maximum voltage. The relationship between the amplitude in Vp and DC offset is as follows.

  $$Vp \leq Vmax - |Voffset|$$

  where Vmax is the maximum voltage of the output connector. For the Balanced output connection, the maximum voltage is 22.6 Vp, while for Unbalanced and Common, the maximum voltage is 11.3 Vp.
- This command is used to set the amplitude of the composite signal if the dual waveform is selected using the `SOURce:FUNCtion` command. Use the `SOURce:VOLTage:RATio` command to set the amplitude ratio of the second component over the first component.
- The allowable unit for the DC signal is only V. The **–131,"Invalid suffix"** error will be generated if other units have been selected for the DC signal.
- For all waveform types except DC, you can select either Vrms, Vpp, Vp, or dBV. The **–131,"Invalid suffix"** error will be generated if you have selected an invalid unit.
- You can also include a multiplier for the unit, for example, mVrms. The 'm' is the multiplier for the unit Vrms.

Examples

The following commands set the amplitude levels for channels 1 and 2 to 1 Vrms and 5 Vrms respectively.

```
SOUR:VOLT 1Vrms, (@1)
```

```
SOUR:VOLT 5Vrms, (@2)
```

The following query returns the amplitude levels of channels 1 and 2 in Vrms.

```
SOUR:VOLT? (@1,2)
```

Typical Response: `1.000000E+00,5.000000E+00`

SOURce:FREQuency[<j>][:CW]

Syntax

```
SOURce:FREQuency[<j>][:CW] <frequency>[<unit>],
(@<channel list>)
```

```
SOURce:FREQuency[<j>][:CW]? (@<channel list>)
```

Description

Sets the signal frequency for the specified channel(s) in Hz. The query returns the frequency of the selected channel(s) in Hz. Multiple responses are separated by commas.

Parameter

Item	Type	Range of values	Default value
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Refer to "Appendix A: Waveform Frequency Range and Default Values" on page 238 for the `<frequency>` parameter.

Remarks

- The <j> parameter represents 1 or 2.
  - For the SMPTE IMD 1:1, 4:1, and 10:1 waveforms, SOURce:FREQuency1 represents the lower frequency while SOURce:FREQuency2 represents the upper frequency.
  - For the dual waveform, SOURce:FREQuency1 represents the frequency of the first sine component while SOURce:FREQuency2 represents the frequency of the second sine component.
  - For the DFD IEC 60118 waveform, use SOURce:FREQuency2 to set the upper frequency and the SOURce:FREQuency:DIFFerence command to set the frequency difference.
  - For the DFD IEC 60268 waveform, use the SOURce:FREQuency:DIFFerence command to set the frequency difference and the SOURce:FREQuency:CENTer command to set the center frequency.
- The frequency setting is not applicable for the DC, noise, multitone, and arbitrary waveforms.
- You can also include a multiplier for the unit, for example, kHz. The 'k' is the multiplier for the unit Hz.

Examples

The following commands set the sine waveform frequency for channel 1 and square waveform frequency for channel 2 to 1 kHz and 5 kHz respectively.

```
SOUR:FREQ 1000, (@1)

SOUR:FREQ 5000, (@2)
```

The following query returns the frequency values of channels 1 and 2 in Hz.

```
SOUR:FREQ? (@1,2)
```

Typical Response: `1.000000E+03,5.000000E+03`

SOURce:FREQuency:CENTer

Syntax

```
SOURce:FREQuency:CENTer <frequency>[<unit>],
(@<channel list>)
```

```
SOURce:FREQuency:CENTer? (@<channel list>)
```

Description

Sets the center frequency of the DFD IEC 60268 waveform for the specified channel(s) in Hz. The query returns the center frequency value of the selected channel(s) in Hz. Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
frequency	Numeric	3 kHz to 79 kHz	10 kHz
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- This setting is only applicable for the DFD IEC 60268 waveform. Use the `SOURce:FUNCtion` command to select the DFD IEC 60268 waveform type.
- You can also include a multiplier for the unit, for example, kHz. The 'k' is the multiplier for the unit Hz.

Examples

The following commands set the center frequencies for channels 1 and 2 to 1 kHz and 5 kHz respectively.

```
SOUR:FREQ:CENT 1kHz, (@1)
```

```
SOUR:FREQ:CENT 5kHz, (@2)
```

The following query returns the center frequency values of channels 1 and 2 in Hz.

```
SOUR:FREQ:CENT? (@1,2)
```

Typical Response: `1.000000E+03,5.000000E+03`

SOURce:FREQuency:DIFFerence

Syntax

```
SOURce:FREQuency:DIFFerence <frequency>[<unit>],
(@<channel list>)
```

```
SOURce:FREQuency:DIFFerence? (@<channel list>)
```

Description

Sets the frequency difference of the DFD IEC 60268 and DFD IEC 60118 waveforms for the specified channel(s) in Hz. The query returns the frequency difference of the selected channel(s) in Hz. Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
frequency	Numeric	80 Hz to 2 kHz	80 Hz
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- This setting is only applicable for the DFD IEC 60118 and DFD IEC 60268 waveforms. Use the `SOURce:FUNCtion` command to select either one of these two waveform types.
- You can also include a multiplier for the unit, for example, kHz. The 'k' is the multiplier for the unit Hz.

Examples

The following commands set the frequency difference values for channels 1 and 2 to 100 Hz and 80 Hz respectively.

```
SOUR:FREQ:DIFF 100Hz, (@1)
```

```
SOUR:FREQ:DIFF 80Hz, (@2)
```

The following query returns the frequency difference values of channels 1 and 2 in Hz.

```
SOUR:FREQ:DIFF? (@1,2)
```

Typical Response: `1.000000E+02,8.000000E+01`

SOURce:VOLTage:RATio

Syntax

```
SOURce:VOLTage:RATio <ratio>, (@<channel list>)
```

```
SOURce:VOLTage:RATio? (@<channel list>)
```

Description

Sets the amplitude ratio of the second component over the first component of the dual waveform for the specified channel(s) in percentage. The query returns the amplitude ratio of the selected channel(s) in percentage. Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
ratio	Numeric	0 to 100%	0
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

This setting is only applicable for the dual waveform. Use the SOURce:FUNCtion command to select the dual waveform type.

Examples

The following commands set the amplitude ratio values for channels 1 and 2 to 1% and 10% respectively.

```
SOUR:VOLT:RAT 1, (@1)
SOUR:VOLT:RAT 10, (@2)
```

The following query returns the amplitude ratio values of channels 1 and 2 in percentage.

```
SOUR:VOLT:RAT? (@1,2)
```

Typical Response: 1.000E+00,1.000E+01

SOURce:PHASe[:ADJust]

Syntax

```
SOURce:PHASe[:ADJust] <phase>, (@<channel list>)
SOURce:PHASe[:ADJust]? (@<channel list>)
```

Description

Sets the phase of the selected channel with reference to channel 1 in degree. The query returns the phase of the selected channel(s) in degree. Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
phase	Numeric	–180 ° to 179.99 °	0
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

This setting is only applicable for the variable phase waveform. Use the `SOURce:FUNCtion` command to select the variable phase waveform type.

Examples

The following command sets the phase for channel 2 to 100 ° with reference to channel 1.

```
SOUR:PHAS 100, (@2)
```

The following query returns the phase of channel 2 with reference to channel 1.

```
SOUR:PHAS? (@2)
```

Typical Response: `1.000000E+02`

SOURce:MULTitone:COUNt

Syntax

```
SOURce:MULTitone:COUNt <tone count>,
(@<channel list>)
```

```
SOURce:MULTitone:COUNt? (@<channel list>)
```

Description

Sets the tones of the multitone waveform for the specified channel(s). Tones refer to the number of signal frequency components. The query returns the number of tones of the selected channel(s). Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
tone count	Numeric	2 to 60	3
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

This setting is only applicable for the multitone waveform. Use the `SOURce:FUNCtion` command to select the multitone waveform type.

Examples

The following commands set the tones for channels 1 and 2 to 3 and 15 respectively.

```
SOUR:MULT:COUN 3, (@1)

SOUR:MULT:COUN 15, (@2)
```

The following query returns the number of tones of channels 1 and 2.

```
SOUR:MULT:COUN? (@1,2)
```

Typical Response: `3,15`

SOURce:MULTitone:PHASe[:MODE]

Syntax

```
SOURce:MULTitone:PHASe[:MODE] <type>,
(@<channel list>)
```

```
SOURce:MULTitone:PHASe[:MODE]? (@<channel list>)
```

Description

Sets the phase mode of the multitone waveform for the specified channel(s). The phase mode consists of the Zero Phase and Random Phase modes. In the Zero Phase mode, all frequency components have the same phase where there is zero phase difference between any two tones. In Random Phase mode, the phase difference between the tones is randomized. The query returns the phase mode of the selected channel(s) in the form of `ZERO` or `RAND`. Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
type	Discrete	ZERO or RANDom	RANDom
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

This setting is only applicable for the multitone waveform. Use the `SOURce:FUNCtion` command to select the multitone waveform type.

Examples

The following commands set the phase modes for channels 1 and 2 to Zero Phase and Random Phase respectively.

```
SOUR:MULT:PHAS ZERO, (@1)
SOUR:MULT:PHAS RAND, (@2)
```

The following query returns the phase modes of channels 1 and 2.

```
SOUR:MULT:PHAS? (@1,2)
```

Typical Response: `ZERO,RAND`

SOURce:FREQuency:STARt

Syntax

```
SOURce:FREQuency:STARt <start
frequency>[<unit>], (@<channel list>)
```

```
SOURce:FREQuency:STARt? (@<channel list>)
```

Description

Sets the start frequency of the multitone waveform for the specified channel(s) in Hz. The query returns the start frequency of the selected channel(s) in Hz. Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
start frequency	Discrete	10 Hz to 9 kHz	1 kHz
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- This setting is only applicable for the multitone waveform. Use the `SOURce:FUNCtion` command to select the multitone waveform type.
- You can also include a multiplier for the unit, for example, kHz. The 'k' is the multiplier for the unit Hz.

- You must use the frequency multiplier to generate each signal frequency component. Use the SOURce:FREQuency:MULTiplier command to set the multiplier.

Examples

The following commands set the start frequencies for channels 1 and 2 to 100 Hz and 2 kHz respectively.

```
SOUR:FREQ:STAR 100, (@1)
SOUR:FREQ:STAR 2000, (@2)
```

The following query returns the start frequency values of channels 1 and 2 in Hz.

```
SOUR:FREQ:STAR? (@1,2)
```

Typical Response: `1.000000E+02,2.000000E+03`

SOURce:FREQuency:MULTiplier

Syntax

```
SOURce:FREQuency:MULTiplier <multiplier>,
(@<channel list>)
SOURce:FREQuency:MULTiplier? (@<channel list>)
```

Description

Sets the frequency multiplier of the multitone waveform for the specified channel(s). The query returns the multiplier of the selected channel(s). Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
multiplier	Numeric	1 to 7999	19
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- This setting is only applicable for the multitone waveform. Use the `SOURce:FUNCtion` command to select the multitone waveform type.
- The multiplier determines the frequency spacing between the signal frequency components or tones.
  For example, the start frequency of a multitone waveform is 10 Hz. If you set the multiplier to 2, the frequency spacing becomes 20 Hz (2 × 10 Hz). In this case, the second tone will be 30 Hz (10 Hz + 20 Hz), the third tone will be 50 Hz (10 Hz + 20 Hz + 20 Hz), and the same process applies for the rest of the subsequent tones.

Examples

The following commands set the frequency multipliers for channels 1 and 2 to 10 and 500 respectively.

```
SOUR:FREQ:MULT 10, (@1)
SOUR:FREQ:MULT 500, (@2)
```

The following query returns the multipliers of channels 1 and 2.

```
SOUR:FREQ:MULT? (@1,2)
```

Typical Response: `10,500`

Sense Subsystem

The Sense subsystem provides the commands to select the U8903A measurement functions and configure the measurement settings for the analyzer and graph modes.

SENSe:FUNCtion<j>

Syntax

```
SENSe:FUNCtion<j> <function>, (@<channel list>)
SENSe:FUNCtion<j>? (@<channel list>)
```

Description

Sets the analyzer measurement function for the specified channel(s). The query returns the measurement function of the selected channel(s). Multiple responses are separated by commas.

The measurement functions with their corresponding `<function>` parameters are listed as follows.

`FREQuency`	Frequency measurement
`VAC`	AC voltage measurement
`VDC`	DC voltage measurement
`THDRatio`	THD + N Ratio measurement
`THDLevel`	THD + N Level measurement
`SINad`	SINAD measurement
`SNRatio`	Signal-to-noise ratio measurement
`NOISe`	Noise Level measurement
`IMD`	SMPTE IMD measurement
`SDFDiec118`	DFD IEC 60118 2nd order measurement
`TDFDiec118`	DFD IEC 60118 3rd order measurement
`SDFDiec268`	DFD IEC 60268 2nd order measurement
`TDFDiec268`	DFD IEC 60268 3rd order measurement
`PHASe`	Phase measurement
`DCRosstalk`	Crosstalk (channel driven) measurement
`MCRosstalk`	Crosstalk (channel measured) measurement

Parameters

Item	Type	Range of values	Default value
j	Numeric	1 to 2 • SENS:FUNC1 indicates the first measurement function • SENS:FUNC2 indicates the second measurement function	1
function	Discrete	FREQuency, VAC, VDC, THDRatio, THDLevel, SINad, SNRatio, NOISe, IMD, SDFDiec118, TDFDiec118, SDFDiec268, TDFDiec268, PHASe, DCRosstalk, or MCRosstalk	• FREQuency (first measurement function) • VAC (second measurement function)
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- For the first measurement function, there are only three types of selectable functions comprising frequency, Vac, and Vdc. For the second measurement function, you can select any of the measurement functions listed above.
- You need to set the DC coupling (`INPut:COUPling` command) prior to setting the Vdc measurement function.
- If either phase, crosstalk (channel driven), or crosstalk (channel measured) is selected, you must also specify the reference channel using the `SENSe:REFerence:CHANnel` command. You must set the reference channel prior to sending the `SENSe:FUNCtion` command.
- If phase is selected for a particular channel, the measurement function for all channels will automatically change to phase.

- If crosstalk (channel driven) is selected for a particular channel, all channels will change to the crosstalk measurement. The measurement function for all channels except the reference channel will change to crosstalk (channel measured) and the measurement function for the reference channel will change to crosstalk (channel driven).
- If crosstalk (channel measured) is selected for a particular channel, all channels will change to the crosstalk measurement. The measurement function for all channels except the reference channel will change to crosstalk (channel driven) and the measurement function for the reference channel will change to crosstalk (channel measured).
- Refer to "Measuring the crosstalk" on page 204 for the programming example on measuring crosstalk.
- Noise Level is not applicable for the sweep measurement parameter selection.

Examples

To measure the DC voltage on channel 1, you can set Vdc as the first measurement function. Assume that the DC voltage is measured immediately without waiting for any bus or external trigger. The following commands are configured.

```
SENS:FUNC1 VDC, (@1)

TRIG:ANAL:SOUR IMM

INIT:ANAL (@1)

FETC? FUNC1, (@1)
```

NOTE

- When FETCh is queried, the measurement result will be returned in the unit as listed in "Appendix B: Units of the Measurement Function Returned Values" on page 239.
- For crosstalk measurements, a value of 0 dB or 100% will always be returned when FETCh is used to acquire the result of the reference channel.

The following query returns the measurement function of channel 1.

```
SENS:FUNC1? (@1)
```

Typical Response: `VDC`

SENSe:FUNCtion<j>:UNIT

Syntax

```
SENSe:FUNCtion<j>:UNIT <unit>, (@<channel list>)
```

```
SENSe:FUNCtion<j>:UNIT? (@<channel list>)
```

Description

Specifies the unit for the measurement result (which is obtained using the `FETCh` command) of the corresponding function for the selected channel(s). The query returns the unit of the corresponding function for the selected channel(s). Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
j	Numeric	1 to 2 • `SENS:FUNC1` indicates the first measurement function • `SENS:FUNC2` indicates the second measurement function	1
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

For the `<unit>` range of values and formulas, refer to "Appendix B: Units of the Measurement Function Returned Values" on page 239.

Remarks

The unit specified using this command will cause the measurement result to be returned in that unit. For example, changing the unit to dBV for the Vac function will return the measurement result obtained by the `FETCh` command in dBV.

Examples

The following commands set the AC voltage as the second measurement function in the unit dBV for both channels.

```
SENS:FUNC2 VAC, (@1,2)
SENS:FUNC2:UNIT dBV, (@1,2)
```

The following query returns the unit of the second measurement function for both channels.

```
SENS:FUNC2:UNIT? (@1,2)
```

Typical Response: `dBV,dBV`

SENSe:REFerence:CHANnel

Syntax

```
SENSe:REFerence:CHANnel <reference channel>
SENSe:REFerence:CHANnel?
```

Description

Sets the reference channel for the phase or crosstalk measurement functions. The query returns the reference channel.

Parameter

Item	Type	Range of values	Default value
reference channel	Numeric	1 or 2	1

Examples

The following commands provide the sequence to measure crosstalk from channel 2 to 1.

```
SENS:REF:CHAN 2

SENS:FUNC2 DCR, (@1,2)

TRIG:ANAL:SOUR IMM

INIT:ANAL (@1)

FETC? FUNC2, (@1)
```

The following query returns the reference channel.

```
SENS:REF:CHAN?
```

Typical Response: `2`

SENSe:VOLTage:RANGe:AUTO

Syntax

```
SENSe:VOLTage:RANGe:AUTO <mode>, (@<channel list>)
SENSe:VOLTage:RANGe:AUTO? (@<channel list>)
```

Description

Disables or enables autoranging for voltage measurements for the specified channel(s). Autoranging allows the U8903A to automatically select the range for each measurement based on the input signal detected. The query returns the autoranging state of the selected channel(s) as `0` if the autoranging is disabled, or `1` if the autoranging is enabled. Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
mode	Boolean	OFF(0) or ON(1)	ON
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- Selecting a discrete range using the `SENSe:VOLTage:RANGe[:UPPER]` command will disable the autoranging.
- Autoranging is enabled after a factory reset (`*RST`) command or instrument preset (`SYSTem:PRESet`) command.

Examples

The following commands disable autoranging for channel 1 but enable autoranging for channel 2.

```
SENS:VOLT:RANG:AUTO OFF, (@1)
SENS:VOLT:RANG:AUTO ON, (@2)
```

The following query returns the autoranging states for channels 1 and 2.

```
SENS:VOLT:RANG:AUTO? (@1,2)
```

Typical Response: `0,1`

SENSe:VOLTage:RANGe[:UPPer]

Syntax

```
SENSe:VOLTage:RANGe[:UPPer] <range>[<unit>],
(@<channel list>)
SENSe:VOLTage:RANGe[:UPPer]? (@<channel list>)
```

Description

Sets the measurement range for voltage measurements for the specified channel(s) in V. The query returns the voltage range of the selected channel(s) in V. Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
range	Numeric	400 mV, 800 mV, 1.6 V, 3.2 V, 6.4 V, 12.8 V, 25 V, 50 V, 100 V, or 140 V	Required parameter
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- Selecting a discrete range using this command will disable the autoranging on the specified channel(s).
- If the input signal is greater than the selected measurement range, an overload indication of `9.9E+37` will be generated.
- Autoranging is enabled after a factory reset (`*RST`) command or instrument preset (`SYSTem:PRESet`) command.
- If you set a value in between the selected measurement range, for example, 401 mV, it will clip the value to the upper range which is 800 mV.
- You can also include a multiplier for the unit, for example, mV. The 'm' is the multiplier for the unit V.

Examples

The following commands set the measurement range values to 400 mV and 3.2 V for channels 1 and 2 respectively.

```
SENS:VOLT:RANG 400mV, (@1)

SENS:VOLT:RANG 3.2V, (@2)
```

The following query returns the measurement range values for channels 1 and 2.

```
SENS:VOLT:RANG? (@1,2)
```

Typical Response: `4.000000E-01,3.200000E+00`

SENSe:VOLTage:DETector

Syntax

```
SENSe:VOLTage:DETector <detector type>,
(@<channel list>)
SENSe:VOLTage:DETector? (@<channel list>)
```

Description

Sets the analyzer AC level detector for the specified channel(s). The query returns the detector type of the selected channel(s). Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
detector type	Discrete	RMS, QPK, or VPP	RMS
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- This command is only applicable if the selected measurement function is Vac.
- If `VPP` is selected, querying `FETCh[:SCALar]?` will return the AC voltage result in Vpp. If `RMS` is selected, the returned AC voltage result is in Vrms. If `QPK` is selected, the returned AC voltage result is in V.

Examples

The following commands set the detector types to RMS and Vpp for channels 1 and 2 respectively.

```
SENS:VOLT:DET RMS, (@1)
SENS:VOLT:DET VPP, (@2)
```

The following query returns the detector types for channels 1 and 2.

```
SENS:VOLT:DET? (@1,2)
```

Typical Response: `RMS,VPP`

SENSe:MTIMe

Syntax

```
SENSe:MTIMe <measurement time>
SENSe:MTIMe?
```

Description

Sets the analyzer measurement time. The query returns the measurement time.

The measurement time values with their corresponding `<measurement time>` parameters are listed as follows.

`GTRack`	Gen Track
`SP128`	1/128 s
`SP64`	1/64 s
`SP32`	1/32 s
`SP16`	1/16 s
`SP8`	1/8 s
`SP4`	1/4 s
`SP2`	1/2 s
`S1`	1 s

Parameter

Item	Type	Range of values	Default value
measurement time	Discrete	GTRack, SP128, SP64, SP32, SP16, SP8, SP4, SP2, or S1	GTRack

Remark

The selected measurement time applies for all channels.

Examples

The following command sets the measurement time to 1/128 s.

```
SENS:MTIM SP128
```

The following query returns the measurement time.

```
SENS:MTIM?
```

Typical Response: `SP128`

SENSe:FILTer:LPASs

Syntax

```
SENSe:FILTer:LPASs <low pass filter>, (@<channel list>)
SENSe:FILTer:LPASs? (@<channel list>)
```

Description

Sets the low pass filter for the specified channel(s). The query returns the low pass filter type of the selected channel(s). Multiple responses are separated by commas.

The low pass filter types with their corresponding <low pass filter> parameters are listed as follows.

NONE	No low pass filter is applied
LP15	Low pass filter with 15 kHz cutoff frequency
LP20	Low pass filter with 20 kHz cutoff frequency
LP30	Low pass filter with 30 kHz cutoff frequency
CUSTom	User-defined low pass filter

Parameters

Item	Type	Range of values	Default value
low pass filter	Discrete	NONE, LP15, LP20, LP30, or CUSTom	NONE
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

If you switch from CUSTom to either NONE, LP15, LP20, or LP30, your previously downloaded information for the custom filter will be lost. You will need to redownload the filter information into the system. Refer to "Using the user-defined filter data" on page 210 for the information on how to use the custom filter.

Examples

The following commands set the low pass filter types to LP15 and LP30 for channels 1 and 2 respectively.

```
SENS:FILT:LPAS LP15, (@1)

SENS:FILT:LPAS LP30, (@2)
```

The following query returns the low pass filter types for channels 1 and 2.

```
SENS:FILT:LPAS? (@1,2)
```

Typical Response: `LP15,LP30`

SENSe:FILTer:HPASs

Syntax

```
SENSe:FILTer:HPASs <high pass filter>,
(@<channel list>)
SENSe:FILTer:HPASs? (@<channel list>)
```

Description

Sets the high pass filter for the specified channel(s). The query returns the high pass filter type of the selected channel(s). Multiple responses are separated by commas.

The high pass filter types with their corresponding `<high pass filter>` parameters are listed as follows.

`NONE`	No high pass filter is applied
`HP22`	High pass filter with 22 Hz cutoff frequency
`HP100`	High pass filter with 100 Hz cutoff frequency
`HP400`	High pass filter with 400 Hz cutoff frequency
`CUSTom`	User-defined high pass filter

Parameters

Item	Type	Range of values	Default value
high pass filter	Discrete	NONE, HP22, HP100, HP400, or CUSTom	NONE
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

If you switch from CUSTom to either NONE, HP22, HP100, or HP400, your previously downloaded information for the custom filter will be lost. You will need to redownload the filter information into the system. Refer to "Using the user-defined filter data" on page 210 for the information on how to use the custom filter.

Examples

The following commands set the high pass filter types to HP22 and HP100 for channels 1 and 2 respectively.

```
SENS:FILT:HPAS HP22, (@1)
SENS:FILT:HPAS HP100, (@2)
```

The following query returns the high pass filter types for channels 1 and 2.

```
SENS:FILT:HPAS? (@1,2)
```

Typical Response: HP22,HP100

SENSe:FILTer:WEIGhting

Syntax

```
SENSe:FILTer:WEIGhting <weighting filter>,
(@<channel list>)
SENSe:FILTer:WEIGhting? (@<channel list>)
```

Description

Sets the weighting filter for the specified channel(s). The query returns the weighting filter type of the selected channel(s). Multiple responses are separated by commas.

The weighting filter types with their corresponding <weighting filter> parameters are listed as follows.

NONE	No weighting filter is applied
AWEighting	A-Weighting filter
CCIR1k	CCIR-1k weighted
CCIR2k	CCIR-2k weighted
CMESsage	C-Message
CCITt	CCITT
CUSTom	User-defined arbitrary filter type including Bandpass and Bandstop filters

Parameters

Item	Type	Range of values	Default value
weighting filter	Discrete	NONE, AWEighting, CCIR1k, CCIR2k, CMESsage, CCITt, or CUSTom	NONE
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- The custom filter type includes the bandpass and bandstop arbitrary filters.
- If you switch from CUSTom to either NONE, AWE, CCIR1k, CCIR2k, CMES, or CCIT, your previously downloaded information for the custom filter will be lost. You will need to redownload the filter information into the system. Refer to "Using the user-defined filter data" on page 210 for the information on how to use the custom filter.

Examples

The following commands set the weighting filter types to A-Weighting and C-Message for channels 1 and 2 respectively.

```
SENS:FILT:WEIG AWE, (@1)
```

```
SENS:FILT:WEIG CMES, (@2)
```

The following query returns the weighting filter types for channels 1 and 2.

```
SENS:FILT:WEIG? (@1,2)
```

Typical Response: `AWE,CMES`

SENSe:REFerence:LEVel

Syntax

```
SENSe:REFerence:LEVel <level>, (@<channel list>)
SENSe:REFerence:LEVel? (@<channel list>)
```

Description

Sets the reference level for the specified channel(s) in V. The reference level is used for conversion of the measurement result in unit dBr. The query returns the reference level of the selected channel(s). Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
level	Numeric	0 < level < INF (INF is 9.90E+37)	387.3 mV
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remark

The reference level setting is only applicable for the Vac, Vdc, THD + N Level, and Noise Level measurement functions to specify the measurement results in dBr.

Examples

The following commands set the reference levels to 200 mV and 500 mV for channels 1 and 2 respectively.

```
SENS:REF:LEV 0.2, (@1)
```

```
SENS:REF:LEV 0.5, (@2)
```

The following query returns the reference levels for channels 1 and 2.

```
SENS:REF:LEV? (@1,2)
```

Typical Response: `2.000000E-01,5.000000E-01`

SENSe:REFerence:IMPedance

Syntax

```
SENSe:REFerence:IMPedance <impedance>,
(@<channel list>)
SENSe:REFerence:IMPedance? (@<channel list>)
```

Description

Sets the reference impedance for the specified channel(s) in ohms (Ω). The reference impedance is used for conversion of the measurement result in unit W or dBm. The query returns the reference impedance of the selected channel(s). Multiple responses are separated by commas.

Parameters

Item	Type	Range of values	Default value
impedance	Numeric	0 < impedance < INF (INF is 9.90E+37)	600 Ω
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remark

The reference impedance setting is only applicable for the Vac, Vdc, THD + N Level, and Noise Level measurement functions to specify the measurement results in W or dBm.

Examples

The following commands set the reference impedances to 600 Ω and 50 Ω for channels 1 and 2 respectively.

```
SENS:REF:IMP 600, (@1)
SENS:REF:IMP 50, (@2)
```

The following query returns the reference impedances for channels 1 and 2.

```
SENS:REF:IMP? (@1,2)
```

Typical Response: `6.000000E+02,5.000000E+01`

SENSe:AVERaging:SYNC:POINts

Syntax

```
SENSe:AVERaging:SYNC:POINts <number of points>
SENSe:AVERaging:SYNC:POINts?
```

Description

Sets the number of points for the synchronous averaging. Synchronous averaging reduces noise levels by averaging the acquired data in the time domain. The query returns the number of averaging points.

Parameter

Item	Type	Range of values	Default value
number of points	Numeric	1 to 64	1

Remark

This setting is only applicable if you trigger from the channel 1 or channel 2 input.

Examples

The following command sets eight averaging points.

```
SENS:AVER:SYNC:POIN 8
```

The following query returns the number of averaging points.

```
SENS:AVER:SYNC:POIN?
```

Typical Response: `8`

SENSe:WAVeform:POINts

Syntax

```
SENSe:WAVeform:POINts <number of points>
SENSe:WAVeform:POINts?
```

Description

Sets the number of data points to acquire with the `FETCh:ARRAy?` command. The query returns the selected acquisition length.

If you select the frequency domain analysis, the acquisition length represents the FFT size. The acquisition length of the frequency domain analysis doubles the acquisition length that you select using this command.

Parameter

Item	Type	Range of values	Default value
number of points	Numeric	256, 512, 1024, 2048, 4096, 8192, 16384, or 32768	256

Remarks

If the number of points that you enter is not the exact value of the acquisition length, the value is always clipped to its lower value. For instance, if the number of points that you enter is 500, it will be clipped to 256 which is the number lower than 500.

Examples

The following command sets the acquisition length to 512.

```
SENS:WAV:POIN 512
```

The following query returns the acquisition length.

```
SENS:WAV:POIN?
```

Typical Response: `512`

SENSe:FFT:WINDow

Syntax

```
SENSe:FFT:WINDow <type>
SENSe:FFT:WINDow?
```

Description

Sets the window function for frequency domain analysis. The query returns the window function.

The window functions with their corresponding <type> parameters are listed as follows.

HANN	Hann window
RECTangular	Rectangular window
BLACkman	Blackman-Harris window
RIFe1	Rife-Vincent 1 window
RIFe3	Rife-Vincent 3 window
HAMMing	Hamming window
FLATtop	Flattop window

Parameters

Item	Type	Range of values	Default value
type	Discrete	RECTangular, HANN, BLACkman, RIFe1, RIFe3, HAMMing, or FLATtop	RECTangular

Examples

The following command sets the Rectangular window function.

```
SENS:FFT:WIND RECT
```

The following query returns the window function.

```
SENS:FFT:WIND?
```

Typical Response: RECT

Display Subsystem

The Display subsystem provides the commands to select the U8903A graph display and front panel LCD display, as well as configure the axis settings for the graph and sweep modes.

DISPlay:ANALysis:MODE

Syntax

```
DISPlay:ANALysis:MODE <mode>
DISPlay:ANALysis:MODE?
```

Description

Sets the graph display as either time domain, frequency domain (magnitude), or frequency domain (phase). The query returns the graph display mode in the form of `MAGN`, `PHAS`, or `TIME`.

Parameter

Item	Type	Range of values	Default value
mode	Discrete	MAGNitude, PHASe, or TIME	MAGNitude

Examples

The following command sets the graph display as frequency domain (magnitude).

```
DISP:ANAL:MODE MAGN
```

The following query returns the graph display mode.

```
DISP:ANAL:MODE?
```

Typical Response: `MAGN`

DISPlay[:WINDow]:GRAPh:TRACe:X:SPACing

Syntax

```
DISPlay[:WINDow]:GRAPh:TRACe:X:SPACing
<spacing type>
DISPlay[:WINDow]:GRAPh:TRACe:X:SPACing?
```

Description

Sets the X-axis spacing as either linear or log. The query returns the X-axis spacing type in the form of `LIN` or `LOG`.

Parameter

Item	Type	Range of values	Default value
spacing type	Discrete	LINear or LOGarithmic	LINear

Examples

The following command sets the log X-axis spacing.

```
DISP:GRAP:TRAC:X:SPAC LOG
```

The following query returns the X-axis spacing type.

```
DISP:GRAP:TRAC:X:SPAC?
```

Typical Response: `LOG`

DISPlay[:WINDow]:GRAPh:TRACe:X[:SCALe]:AUTO

Syntax

```
DISPlay[:WINDow]:GRAPh:TRACe:X[:SCALe]:AUTO
```

Description

Performs an autoscale on the X-axis to automatically scale the graph display according to the signal each time this command is sent.

Example

The following command performs an autoscale on the X-axis.

```
DISP:GRAP:TRAC:X:AUTO
```

DISPlay[:WINDow]:GRAPh:TRACe:X[:SCALe]:LEFT

Syntax

```
DISPlay[:WINDow]:GRAPh:TRACe:X[:SCALe]:LEFT
<minimum limit>
DISPlay[:WINDow]:GRAPh:TRACe:X[:SCALe]:LEFT?
```

Description

Sets the value represented by the minimum (left) edge of the X-axis. The query returns the left X-axis setting.

Parameter

Item	Type	Range of values	Default value
minimum limit	Numeric	–200000 to 200000	0

Examples

The following command sets the left X-axis setting to 100.

```
DISP:GRAP:TRAC:X:LEFT 100
```

The following query returns the left X-axis setting.

```
DISP:GRAP:TRAC:X:LEFT?
```

Typical Response: `1.000000E+02`

DISPlay[:WINDow]:GRAPh:TRACe:X[:SCALe]:RIGHt

Syntax

```
DISPlay[:WINDow]:GRAPh:TRACe:X[:SCALe]:RIGHt
<maximum limit>
DISPlay[:WINDow]:GRAPh:TRACe:X[:SCALe]:RIGHt?
```

Description

Sets the value represented by the maximum (right) edge of the X-axis. The query returns the right X-axis setting.

Parameter

Item	Type	Range of values	Default value
maximum limit	Numeric	–200000 to 200000	100000

Examples

The following command sets the right X-axis setting to 10000.

```
DISP:GRAP:TRAC:X:RIGH 10000
```

The following query returns the right X-axis setting.

```
DISP:GRAP:TRAC:X:RIGH?
```

Typical Response: `1.000000E+04`

DISPlay[:WINDow]:GRAPh:TRACe:Y:SPACing

Syntax

```
DISPlay[:WINDow]:GRAPh:TRACe:Y:SPACing
<spacing type>
DISPlay[:WINDow]:GRAPh:TRACe:Y:SPACing?
```

Description

Sets the Y-axis spacing as either linear or log. The query returns the Y-axis spacing type in the form of `LIN` or `LOG`.

Parameter

Item	Type	Range of values	Default value
spacing type	Discrete	LINear or LOGarithmic	LINear

Examples

The following command sets the log Y-axis spacing.

```
DISP:GRAP:TRAC:Y:SPAC LOG
```

The following query returns the Y-axis spacing type.

```
DISP:GRAP:TRAC:Y:SPAC?
```

Typical Response: `LOG`

DISPlay[:WINDow]:GRAPh:TRACe:Y[:SCALe]:AUTO

Syntax

```
DISPlay[:WINDow]:GRAPh:TRACe:Y[:SCALe]:AUTO
```

Description

Performs an autoscale on the Y-axis to automatically scale the graph display according to the signal each time this command is sent.

Example

The following command performs an autoscale on the Y-axis.

```
DISP:GRAP:TRAC:Y:AUTO
```

DISPlay[:WINDow]:GRAPh:TRACe:Y[:SCALe]:BOTTom

Syntax

```
DISPlay[:WINDow]:GRAPh:TRACe:Y[:SCALe]:BOTTom
<minimum limit>
DISPlay[:WINDow]:GRAPh:TRACe:Y[:SCALe]:BOTTom?
```

Description

Sets the value represented by the minimum (bottom) edge of the Y-axis. The query returns the bottom Y-axis setting.

Parameter

Item	Type	Range of values	Default value
minimum limit	Numeric	–200000 to 200000	–200

Examples

The following command sets the bottom Y-axis setting to –200.

```
DISP:GRAP:TRAC:Y:BOTT –200
```

The query returns the bottom Y-axis setting.

```
DISP:GRAP:TRAC:Y:BOTT?
```

Typical Response: `–2.000000E+02`

DISPlay[:WINDow]:GRAPh:TRACe:Y[:SCALe]:TOP

Syntax

```
DISPlay[:WINDow]:GRAPh:TRACe:Y[:SCALe]:TOP
<maximum limit>
DISPlay[:WINDow]:GRAPh:TRACe:Y[:SCALe]:TOP?
```

Description

Sets the value represented by the maximum (top) edge of the Y-axis. The query returns the top Y-axis setting.

Parameter

Item	Type	Range of values	Default value
maximum limit	Numeric	–200000 to 200000	0

Examples

The following command sets the top Y-axis setting to 200.

```
DISP:GRAP:TRAC:Y:TOP 200
```

The following query returns the top Y-axis setting.

```
DISP:GRAP:TRAC:Y:TOP?
```

Typical Response: `2.000000E+02`

DISPlay[:WINDow]:GRAPh:TRACe:AUTO

Syntax

```
DISPlay[:WINDow]:GRAPh:TRACe:AUTO
```

Description

Performs an autoscale to automatically scale the graph display according to the signal each time this command is sent.

Example

The following command performs an autoscale on the graph.

```
DISP:GRAP:TRAC:AUTO
```

DISPlay[:WINDow]:SWEep:TRACe:X:SPACing

Syntax

```
DISPlay[:WINDow]:SWEep:TRACe:X:SPACing
<spacing type>
DISPlay[:WINDow]:SWEep:TRACe:X:SPACing?
```

Description

Sets the X-axis spacing as either linear or log for the sweep. The query returns the X-axis spacing in the form of `LIN` or `LOG`.

Parameter

Item	Type	Range of values	Default value
spacing type	Discrete	LINear or LOGarithmic	LINear

Examples

The following command sets the log X-axis spacing.

```
DISP:SWE:TRAC:X:SPAC LOG
```

The following query returns the X-axis spacing type.

```
DISP:SWE:TRAC:X:SPAC?
```

Typical Response: `LOG`

DISPlay[:WINDow]:SWEep:TRACe:X[:SCALe]:AUTO

Syntax

```
DISPlay[:WINDow]:SWEep:TRACe:X[:SCALe]:AUTO
```

Description

Performs an autoscale on the X-axis of the sweep plot to automatically scale the sweep plot according to the signal each time this command is sent.

Example

The following command performs an autoscale on the X-axis.

```
DISP:SWE:TRAC:X:AUTO
```

DISPlay[:WINDow]:SWEep:TRACe:X[:SCALe]:LEFT

Syntax

```
DISPlay[:WINDow]:SWEep:TRACe:X[:SCALe]:LEFT
<minimum limit>
DISPlay[:WINDow]:SWEep:TRACe:X[:SCALe]:LEFT?
```

Description

Sets the value represented by the minimum (left) edge of the X-axis of the sweep plot. The query returns the left X-axis setting.

Parameter

Item	Type	Range of values	Default value
minimum limit	Numeric	–200000 to 200000	5

Examples

The following command sets the left X-axis setting to 5.

```
DISP:SWE:TRAC:X:LEFT 5
```

The following query returns the left X-axis setting.

```
DISP:SWE:TRAC:X:LEFT?
```

Typical Response: `5.000000E+00`

DISPlay[:WINDow]:SWEep:TRACe:X[:SCALe]:RIGHt

Syntax

```
DISPlay[:WINDow]:SWEep:TRACe:X[:SCALe]:RIGHt
<maximum limit>
DISPlay[:WINDow]:SWEep:TRACe:X[:SCALe]:RIGHt?
```

Description

Sets the value represented by the maximum (right) edge of the X-axis of the sweep plot. The query returns the right X-axis setting.

Parameter

Item	Type	Range of values	Default value
maximum limit	Numeric	–200000 to 200000	80000

Examples

The following command sets the right X-axis setting to 10000.

```
DISP:SWE:TRAC:X:RIGH 10000
```

The following query returns the right X-axis setting.

```
DISP:SWE:TRAC:X:RIGH?
```

Typical Response: `1.000000E+04`

DISPlay[:WINDow]:SWEep:TRACe:Y:SPACing

Syntax

```
DISPlay[:WINDow]:SWEep:TRACe:Y:SPACing
<spacing type>
DISPlay[:WINDow]:SWEep:TRACe:Y:SPACing?
```

Description

Sets the Y-axis spacing as either linear or log for the sweep. The query returns the Y-axis spacing in the form of `LIN` or `LOG`.

Parameter

Item	Type	Range of values	Default value
spacing type	Discrete	LINear or LOGarithmic	LINear

Examples

The following command sets the log Y-axis spacing.

```
DISP:SWE:TRAC:Y:SPAC LOG
```

The following query returns the Y-axis spacing type.

```
DISP:SWE:TRAC:Y:SPAC?
```

Typical Response: `LOG`

DISPlay[:WINDow]:SWEep:TRACe:Y[:SCALe]:AUTO

Syntax

```
DISPlay[:WINDow]:SWEep:TRACe:Y[:SCALe]:AUTO
```

Description

Performs an autoscale on the Y-axis of the sweep plot to automatically scale the sweep plot according to the signal each time this command is sent.

Example

The following command performs an autoscale on the Y-axis.

```
DISP:SWE:TRAC:Y:AUTO
```

DISPlay[:WINDow]:SWEep:TRACe:Y[:SCALe]:BOTTom

Syntax

```
DISPlay[:WINDow]:SWEep:TRACe:Y[:SCALe]:BOTTom
<minimum limit>
DISPlay[:WINDow]:SWEep:TRACe:Y[:SCALe]:BOTTom?
```

Description

Sets the value represented by the minimum (bottom) edge of the Y-axis of the sweep plot. The query returns the bottom Y-axis setting.

Parameter

Item	Type	Range of values	Default value
minimum limit	Numeric	–200000 to 200000	–1

Examples

The following command sets the bottom Y-axis setting to 50.

```
DISP:SWE:TRAC:Y:BOTT 50
```

The following query returns the bottom Y-axis setting.

```
DISP:SWE:TRAC:Y:BOTT?
```

Typical Response: `5.000000E+01`

DISPlay[:WINDow]:SWEep:TRACe:Y[:SCALe]:TOP

Syntax

```
DISPlay[:WINDow]:SWEep:TRACe:Y[:SCALe]:TOP
<maximum limit>
DISPlay[:WINDow]:SWEep:TRACe:Y[:SCALe]:TOP?
```

Description

Sets the value represented by the maximum (top) edge of the Y-axis of the sweep plot. The query returns the top Y-axis setting.

Parameter

Item	Type	Range of values	Default value
maximum limit	Numeric	–200000 to 200000	1

Examples

The following command sets the top Y-axis setting to 100.

```
DISP:SWE:TRAC:Y:TOP 100
```

The following query returns the top Y-axis setting.

```
DISP:SWE:TRAC:Y:TOP?
```

Typical Response: `1.000000E+02`

DISPlay[:WINDow]:SWEep:TRACe:AUTO

Syntax

```
DISPlay[:WINDow]:SWEep:TRACe:AUTO
```

Description

Performs an autoscale on the sweep plot to automatically scale the sweep plot according to the signal each time this command is sent.

Example

The following command performs an autoscale on the sweep plot.

```
DISP:SWE:TRAC:AUTO
```

DISPlay[:WINDow]:VIEW

Syntax

```
DISPlay[:WINDow]:VIEW <view>, [<channel>]
DISPlay[:WINDow]:VIEW?
```

Description

Sets the front panel LCD display type for the specified channel. The query returns the current display type of the selected channel. Multiple responses are separated by commas.

The display types with their corresponding `<view>` parameters are listed as follows.

`ANALyzer2`	Analyzer mode 2-channel display
`ANALyzer8`	Analyzer mode 8-channel display
`GENerator2`	Generator mode 2-channel display
`GENerator8`	Generator mode 8-channel display
`ASWeep`	Sweep mode axis settings
`GSWeep`	Sweep mode graph view
`ISWeep`	Sweep mode input settings
`SSWeep`	Sweep mode sweep settings
`TSWeep`	Sweep mode list view
`AGRaph`	Graph mode axis settings
`GGRaph`	Graph mode graph view
`IGRaph`	Graph mode input settings
`MGRaph`	Graph mode monitor settings
`HGRaph`	Graph mode harmonics view
`SYSTem`	System mode

Parameters

Item	Type	Range of values	Default value
view	Discrete	ANALyzer2, ANALyzer8, GENerator2, GENerator8, ASWeep, GSWeep, ISWeep, SSWeep, TSWeep, AGRaph, GGRaph, IGRaph, MGRaph, HGRaph, or SYSTem	ANALyzer2
channel	Discrete	NONE, CHANnel1, or CHANnel2	NONE

Remarks

- The `<channel>` parameter is optional.
- `NONE` indicates no channel selection. When `ASWeep`, `GSWeep`, `ISWeep`, `SSWeep`, `TSWeep`, `AGRaph`, `GGRaph`, `IGRaph`, `MGRaph`, `HGRaph`, or `SYSTem` is the selected display type, the `<channel>` parameter has to be set to `NONE`.

Examples

The following command sets the 2-channel display of the analyzer mode for channel 1.

```
DISP:VIEW ANAL2, CHAN1
```

The following query returns the display type of channel 1.

```
DISP:VIEW?
```

Typical Response: `ANAL2,CHAN1`

DISPlay[:WINDow]:STATe

Syntax

```
DISPlay[:WINDow]:STATe <state>
DISPlay[:WINDow]:STATe?
```

Description

Enables or disables the front panel LCD backlight. The query returns the LCD backlight state as `0` if the state is `OFF`, or `1` if the state is `ON`.

Parameter

Item	Type	Range of values	Default value
state	Boolean	OFF(0) or ON(1)	ON

Remark

Sending the `SYSTem:PRESet` or `*RST` command, or cycling the U8903A power, will enable the LCD backlight.

Examples

The following command enables the front panel LCD backlight.

```
DISP:STAT ON
```

The following query returns the LCD backlight state.

```
DISP:STAT?
```

Typical Response: `1`

Calculate Subsystem

The Calculate subsystem provides the commands to configure the frequency domain harmonics settings as well as the graph marker functions.

CALCulate:HARMonic:COUNt

Syntax

```
CALCulate:HARMonic:COUNt <count>
CALCulate:HARMonic:COUNt?
```

Description

Sets the number of signal harmonic components in the frequency domain (magnitude) display. The query returns the number of harmonic components.

Parameter

Item	Type	Range of values	Default value
count	Numeric	1 to 20	8

Remarks

- The graph display must be set to frequency domain (magnitude) using the `DISPlay:ANALysis:MODE` command.
- This command is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.

Examples

The following command sets eight signal harmonic components.

```
CALC:HARM:COUN 8
```

The following query returns the number of harmonic components.

```
CALC:HARM:COUN?
```

Typical Response: `8`

CALCulate:HARMonic:FUNDamental?

Syntax

```
CALCulate:HARMonic:FUNDamental? (@<channel>)
```

Description

Returns the signal fundamental frequency in Hz for the specified channel.

Parameter

Item	Type	Range of values	Default value
channel	Numeric	1 or 2	Required parameter

Remarks

- The graph display must be set to frequency domain (magnitude) using the `DISPlay:ANALysis:MODE` command.
- This query is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command at the particular input channel.

Example

The following commands are used to obtain the signal fundamental frequency of channel 2.

```
INIT:GRAP (@2)
```

```
CALC:HARM:FUND? (@2)
```

Typical Response: `1.000000E+03`

CALCulate:HARMonic:VALue?

Syntax

```
CALCulate:HARMonic:VALue? (@<channel>)
```

Description

Returns the harmonic component results of the trace for the specified channel. Multiple responses are separated by commas.

Parameter

Item	Type	Range of values	Default value
channel	Numeric	1 or 2	Required parameter

Remarks

- The graph display must be set to frequency domain (magnitude) using the `DISPlay:ANALysis:MODE` command.
- This query is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command at the particular input channel.
- The number of harmonic component results returned is based on the harmonic count specified in the `CALCulate:HARMonic:COUNt` command.

Example

The following commands are used to obtain the harmonic component results of channel 2.

```
INIT:GRAP (@2)
CALC:HARM:VAL? (@2)
```

Typical Response:

```
-1.440191E+00,-6.487222E+01,-7.282130E+01,
-7.767053E+01,-8.125921E+01,-8.396585E+01,
-8.624970E+01,-8.790641E+01
```

CALCulate:HARMonic:FREQuencies?

Syntax

```
CALCulate:HARMonic:FREQuencies? (@<channel>)
```

Description

Returns the signal harmonic frequency values for the specified channel. Multiple responses are separated by commas.

Parameter

Item	Type	Range of values	Default value
channel	Numeric	1 or 2	Required parameter

Remarks

- The graph display must be set to frequency domain (magnitude) using the `DISPlay:ANALysis:MODE` command.
- This query is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command at the particular input channel.
- The number of harmonic frequency values returned is based on the harmonic count specified in the `CALCulate:HARMonic:COUNt` command.

Example

The following commands are used to obtain the harmonic frequency values of channel 2.

```
INIT:GRAP (@2)
CALC:HARM:FREQ? (@2)
```

Typical Response:

```
9.918210E+02,2.002721E+03,2.994543E+03,
4.005431E+03,4.997250E+03,6.008151E+03,
6.999972E+03,7.991791E+03
```

CALCulate:THDistortion?

Syntax

```
CALCulate:THDistortion? <unit>, (@<channel>)
```

Description

Returns the Total Harmonic Distortion (THD) value of the input signal in the specified unit for the selected channel. The returned value can either be in dB or percentage by setting `<unit>` to `DB` or `PCT` respectively.

Parameters

Item	Type	Range of values	Default value
unit	Discrete	DB or PCT	PCT
channel	Numeric	1 or 2	Required parameter

Remarks

- The graph display must be set to frequency domain (magnitude) using the `DISPlay:ANALysis:MODE` command.
- This query is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command at the particular input channel.

Example

The following commands are used to obtain the distortion value of the input signal at channel 2 in percentage.

```
INIT:GRAP (@2)
```

```
CALC:THD? PCT, (@2)
```

Typical Response: `1.691385E+01`

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:STATe

Syntax

```
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:STATe
<state>
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:STATe?
```

Description

Turns on or off the selected marker on the graph display in the graph mode. The selected marker will become the active marker when it is turned on. The query returns the marker state as `0` if the marker state is `OFF`, or `1` if the marker state is `ON`.

Parameter

Item	Type	Range of values	Default value
state	Boolean	OFF(0) or ON(1)	OFF

Remark

This command is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.

Examples

The following command turns on marker 2 on the graph display.

```
CALC:GRAP:MARK2:STAT ON
```

The following query returns the state for marker 2.

```
CALC:GRAP:MARK2:STAT?
```

Typical Response: `1`

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:TRACe

Syntax

```
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:TRACe
<trace no>
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:TRACe?
```

Description

Assigns the marker to the trace of the specified channel on the graph display in the graph mode. The trace number corresponds with the channel number. For example, trace number 1 represents the trace for channel 1. The selected marker will become the active marker. The query returns the trace number for the specified marker.

Parameter

Item	Type	Range of values	Default value
trace no	Discrete	CHANnel1 or CHANnel2	CHANnel1

Remarks

- This command is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.
- If a marker is activated using other command without assigning a channel to it, the marker will be assigned to channel 1 by default.

Examples

The following command assigns marker 1 to the channel 2 trace on the graph display.

```
CALC:GRAP:MARK1:TRAC CHAN2
```

The following query returns the trace number for marker 1.

```
CALC:GRAP:MARK1:TRAC?
```

Typical Response: `CHAN2`

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:X

Syntax

```
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:X
<x position>
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:X?
```

Description

Sets the marker X-axis value on the graph display in the graph mode. The selected marker will become the active marker. The query returns the marker X-axis value. If the marker state is off, the response is not a number (`NAN`).

Parameter

Item	Type	Range of values	Default value
x position	Numeric	–200000 to 200000	0

Remark

This command is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.

Examples

The following command sets the marker 2 X-axis value to 550 Hz on the graph display. (Assume that the graph is in the frequency domain mode)

```
CALC:GRAP:MARK2:X 550
```

The following query returns the marker 2 X-axis value.

```
CALC:GRAP:MARK2:X?
```

Typical Response: `5.500000E+02`

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:Y?

Syntax

```
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:Y?
```

Description

Returns the marker Y-axis value on the graph display in the graph mode.

Remarks

- This query is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.
- If the graph analysis mode is set to FFT magnitude, the returned value is in dBV.
- If the graph analysis mode is set to FFT phase, the returned value is in degree.
- If the graph analysis mode is set to time domain, the returned value is in Vrms.
- If the marker state is off, the response is not a number (`NAN`).

Example

The following query returns the marker 2 Y-axis value.

```
CALC:GRAP:MARK2:Y?
```

Typical Response: `0.000000E+00`

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:PEAK

Syntax

```
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:PEAK
<direction>
```

Description

Searches for the peak value of the trace data by placing the specified marker at either the left or right peak of the graph display in the graph mode. The specified marker will become the active marker.

Parameter

Item	Type	Range of values	Default value
direction	Discrete	LEFT or RIGHt	RIGHt

Remark

This command is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.

Example

The following command places marker 2 at the left peak of the graph display.

```
CALC:GRAP:MARK2:PEAK LEFT
```

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:MIN

Syntax

```
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:MIN
<direction>
```

Description

Searches for the minimum value of the trace data by placing the specified marker at either the left or right minimum of the graph display in the graph mode. The specified marker will become the active marker.

Parameter

Item	Type	Range of values	Default value
direction	Discrete	LEFT or RIGHt	RIGHt

Remark

This command is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.

Example

The following command places marker 2 at the right minimum of the graph display.

```
CALC:GRAP:MARK2:MIN RIGH
```

CALCulate:GRAPh:MARKer:THReshold[:LEVel]

Syntax

```
CALCulate:GRAPh:MARKer:THReshold[:LEVel]
<threshold level>
```

```
CALCulate:GRAPh:MARKer:THReshold[:LEVel]?
```

Description

Sets the threshold level that the marker can identify as a peak or minimum on the graph display in the graph mode. If the trace is above the threshold level, it will be identified as a peak, whereas the trace below the threshold level will be identified as a minimum. The query returns the specified threshold level.

Parameter

Item	Type	Range of values	Default value
threshold level	Numeric	Within the top edge and bottom edge of the display	–100

Remark

This command is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.

Examples

The following command sets the threshold level to 20.

```
CALC:GRAP:MARK:THR 20
```

The following query returns the threshold level.

```
CALC:GRAP:MARK:THR?
```

Typical Response: `2.000000E+01`

CALCulate:GRAPh:MARKer:THReshold:STATe

Syntax

```
CALCulate:GRAPh:MARKer:THReshold:STATe
<threshold state>
```

```
CALCulate:GRAPh:MARKer:THReshold:STATe?
```

Description

Turns on or off the threshold on the graph display in the graph mode. The query returns the threshold state as `0` if the state is `OFF`, or `1` if the state is `ON`.

Parameter

Item	Type	Range of values	Default value
threshold state	Boolean	OFF(0) or ON(1)	OFF

Remark

This command is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.

Examples

The following command turns on the threshold on the graph display.

```
CALC:GRAP:MARK:THR:STAT ON
```

The following query returns the threshold state.

```
CALC:GRAP:MARK:THR:STAT?
```

Typical Response: `1`

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:REFerence

Syntax

```
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:REFerence
<reference marker no>

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:REFerence
?
```

Description

Sets the reference marker for the selected marker on the graph display in the graph mode. The query returns the reference marker for the specified marker.

Parameter

Item	Type	Range of values	Default value
reference marker no	Discrete	M1, M2, M3, M4, M5, M6, M7, M8, or OFF	OFF

Remarks

- This command is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.
- A marker cannot be referenced to itself.
- Sending this command will turn on the reference marker. Select `OFF` to turn off the reference marker.
- If the selected marker has no reference marker when queried, the **–200,"Execution Error;The marker has no reference marker"** error will appear.

Examples

The following command sets the reference marker as marker 2 for marker 1 on the graph display.

```
CALC:GRAP:MARK1:REF M2
```

The following query returns the reference marker for marker 1.

```
CALC:GRAP:MARK1:REF?
```

Typical Response: `M2`

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:XDELta?

Syntax

```
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:XDELta?
```

Description

Returns the difference in the X-axis value between the selected marker and its reference marker on the graph display in the graph mode.

Remarks

- This query is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.
- If the marker state is off, the response is not a number (`NAN`).
- If the selected marker has no reference marker, the response is also not a number (`NAN`).

Examples

The following query returns the delta X-axis value for marker 2.

```
CALC:GRAP:MARK2:XDEL?
```

Typical Response: `3.500000E+02`

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:YDELta?

Syntax

```
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:YDELta?
```

Description

Returns the difference in the Y-axis value between the selected marker and its reference marker on the graph display in the graph mode.

Remarks

- This query is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.
- If the marker state is off, the response is not a number (`NAN`).
- If the selected marker has no reference marker, the response is also not a number (`NAN`).

Example

The following query returns the delta Y-axis value for marker 2.

```
CALC:GRAP:MARK2:YDEL?
```

Typical Response: `5.000000E+00`

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:MOVement

Syntax

```
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:MOVement
<movement characteristic>

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8:MOVement?
```

Description

Sets the marker movement characteristic of either single or in pair on the graph display in the graph mode. The query returns the marker movement characteristic in the form of either SING or PAIR.

The description for each <movement characteristic> parameter is shown as follows.

SINGle	Move only the selected marker on the graph.
PAIR	Move both the selected and reference markers in the same direction on the graph.

Parameter

Item	Type	Range of values	Default value
movement characteristic	Discrete	SINGle or PAIR	SINGle

Remarks

- This command is only applicable after the graph data has been acquired using the INITiate[:IMMediate]:GRAPh command.
- To move the markers in pairs, you need to specify the reference marker of the selected marker prior to sending this command.

Examples

The following command sequence sets marker 1 and marker 2 as its reference marker to move together on the graph.

```
CALC:GRAP:MARK1:REF M2

CALC:GRAP:MARK1:MOV PAIR
```

The following query returns the movement characteristic of marker 1.

```
CALC:GRAP:MARK1:MOV?
```

Typical Response: PAIR

CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8[:SET]:MODE

Syntax

```
CALCulate:GRAPh:MARKer[1]|2|3|4|5|6|7|8[:SET]:
MODE <marker mode>
```

Description

Positions the marker at either the start, stop, or center points of the graph in the graph mode. You may also expand the area between the selected marker and its reference marker.

The description for each `<marker mode>` parameter is shown as follows.

`STARt`	Position the marker at the graph start point.
`STOP`	Position the marker at the graph stop point.
`CENTer`	Position the marker at the graph center point.
`DSPan`	Expand the area of the graph between the selected marker and its reference marker.

Parameter

Item	Type	Range of values	Default value
marker mode	Discrete	STARt, STOP, CENTer, or DSPan	Required parameter

Remarks

- This command is only applicable after the graph data has been acquired using the `INITiate[:IMMediate]:GRAPh` command.
- The DSPan mode is only applicable for a selected marker which has a reference marker.

Example

The following command positions marker 2 at the graph start point.

```
CALC:GRAP:MARK1:MODE STAR
```

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:STATe

Syntax

```
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:STATe
<state>
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:STATe?
```

Description

Turns on or off the selected marker on the graph display in the sweep mode. The selected marker will become the active marker when it is turned on. The query returns the marker state as `0` if the marker state is `OFF`, or `1` if the marker state is `ON`.

Parameter

Item	Type	Range of values	Default value
state	Boolean	OFF(0) or ON(1)	OFF

Remark

This command is only applicable after the sweep data has been acquired using the `INITiate[:IMMediate]:SWEep` command.

Examples

The following command turns on marker 2 on the graph display.

```
CALC:SWE:MARK2:STAT ON
```

The following query returns the state for marker 2.

```
CALC:SWE:MARK2:STAT?
```

Typical Response: `1`

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:TRACe?

Syntax

```
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:TRACe?
```

Description

Assigns the marker to the trace of the specified channel on the graph display in the sweep mode. The trace number corresponds with the channel number. For example, trace number 1 represents the trace for channel 1. The selected marker will become the active marker. The returned value is the trace number for the specified marker.

Remarks

- This query is only applicable after the sweep data has been acquired using the `INITiate[:IMMediate]:SWEep` command.
- Only the query is provided for this setting. The selected marker will be automatically assigned to the current active channel. The active channel for the sweep can be selected using the `SOURce:SWEep:CHANnel` command.

Example

The following query returns the trace number for marker 1.

```
CALC:SWE:MARK1:TRAC?
```

Typical Response: `CHAN2`

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:X

Syntax

```
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:X
<x position>
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:X?
```

Description

Sets the marker X-axis value on the graph display in the sweep mode. The selected marker will become the active marker. The query returns the marker X-axis value. If the marker state is off, the response is not a number (NAN).

Parameter

Item	Type	Range of values	Default value
x position	Numeric	–200000 to 200000	0

Remark

This command is only applicable after the sweep data has been acquired using the `INITiate[:IMMediate]:SWEep` command.

Examples

The following command sets the marker 2 X-axis value to 550 Hz on the graph display.

```
CALC:SWE:MARK2:X 550
```

The following query returns the marker 2 X-axis value.

```
CALC:SWE:MARK2:X?
```

Typical Response: `5.500000E+02`

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:Y?

Syntax

```
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:Y?
```

Description

Returns the marker Y-axis value on the graph display in the sweep mode.

Remarks

- This query is only applicable after the sweep data has been acquired using the `INITiate[:IMMediate]:SWEep` command.
- If the marker state is off, the response is not a number (`NAN`).

Example

The following query returns the marker 2 Y-axis value.

```
CALC:SWE:MARK2:Y?
```

Typical Response: `0.000000E+00`

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:PEAK

Syntax

```
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:PEAK
<direction>
```

Description

Searches for the peak value of the trace data by placing the specified marker at either the left or right peak of the graph display in the sweep mode. The specified marker will become the active marker.

Parameter

Item	Type	Range of values	Default value
direction	Discrete	LEFT or RIGHt	RIGHt

Remark

This command is only applicable after the sweep data has been acquired using the `INITiate[:IMMediate]:SWEep` command.

Example

The following command places marker 2 at the left peak of the graph display.

```
CALC:SWE:MARK2:PEAK LEFT
```

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:MIN

Syntax

```
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:MIN
<direction>
```

Description

Searches for the minimum value of the trace data by placing the specified marker at either the left or right minimum of the graph display in the sweep mode. The specified marker will become the active marker.

Parameter

Item	Type	Range of values	Default value
direction	Discrete	LEFT or RIGHt	RIGHt

Remark

This command is only applicable after the sweep data has been acquired using the `INITiate[:IMMediate]:SWEep` command.

Example

The following command places marker 2 at the right minimum of the graph display.

```
CALC:SWE:MARK2:MIN RIGH
```

CALCulate:SWEep:MARKer:THReshold[:LEVel]

Syntax

```
CALCulate:SWEep:MARKer:THReshold[:LEVel]
<threshold level>
```

```
CALCulate:SWEep:MARKer:THReshold[:LEVel]?
```

Description

Sets the threshold level that the marker can identify as a peak or minimum on the graph display in the sweep mode. If the trace is above the threshold level, it will be identified as a peak, whereas the trace below the threshold level will be identified as a minimum. The query returns the specified threshold level.

Parameter

Item	Type	Range of values	Default value
threshold level	Numeric	Within the top edge and bottom edge of the display	0

Remark

This command is only applicable after the sweep data has been acquired using the `INITiate[:IMMediate]:SWEep` command.

Examples

The following command sets the threshold level to 20.

```
CALC:SWE:MARK:THR 20
```

The following query returns the threshold level.

```
CALC:SWE:MARK:THR?
```

Typical Response: `2.000000E+01`

CALCulate:SWEep:MARKer:THReshold:STATe

Syntax

```
CALCulate:SWEep:MARKer:THReshold:STATe
<threshold state>
```

```
CALCulate:SWEep:MARKer:THReshold:STATe?
```

Description

Turns on or off the threshold on the graph display in the sweep mode. The query returns the threshold state as `0` if the state is `OFF`, or `1` if the state is `ON`.

Parameter

Item	Type	Range of values	Default value
threshold state	Boolean	OFF(0) or ON(1)	OFF

Remark

This command is only applicable after the sweep data has been acquired using the `INITiate[:IMMediate]:SWEep` command.

Examples

The following command turns on the threshold on the graph display.

```
CALC:SWE:MARK:THR:STAT ON
```

The following query returns the threshold state.

```
CALC:SWE:MARK:THR:STAT?
```

Typical Response: `1`

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:REFerence

Syntax

```
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:REFerence
<reference marker no>

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:REFerence
?
```

Description

Sets the reference marker for the selected marker on the graph display in the sweep mode. The query returns the reference marker for the specified marker.

Parameter

Item	Type	Range of values	Default value
reference marker no	Discrete	M1, M2, M3, M4, M5, M6, M7, M8, or OFF	OFF

Remarks

- This command is only applicable after the sweep data has been acquired using the `INITiate[:IMMediate]:SWEep` command.
- A marker cannot be referenced to itself.
- Sending this command will turn on the reference marker. Select `OFF` to turn off the reference marker.
- If the selected marker has no reference marker when queried, the **–200,"Execution Error;The marker has no reference marker"** error will appear.

Examples

The following command sets the reference marker as marker 2 for marker 1 on the graph display.

```
CALC:SWE:MARK1:REF M2
```

The following query returns the reference marker for marker 1.

```
CALC:SWE:MARK1:REF?
```

Typical Response: `M2`

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:XDELta?

Syntax

```
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:XDELta?
```

Description

Returns the difference in the X-axis value between the selected marker and its reference marker on the graph display in the sweep mode.

Remarks

- This query is only applicable after the sweep data has been acquired using the `INITiate[:IMMediate]:SWEep` command.
- If the marker state is off, the response is not a number (`NAN`).
- If the selected marker has no reference marker, the response is also not a number (`NAN`).

Example

The following query returns the delta X-axis value for marker 2.

```
CALC:SWE:MARK2:XDEL?
```

Typical Response: `3.500000E+02`

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:YDELta?

Syntax

```
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:YDELta?
```

Description

Returns the difference in the Y-axis value between the selected marker and its reference marker on the graph display in the sweep mode.

Remarks

- This query is only applicable after the sweep data has been acquired using the `INITiate[:IMMediate]:SWEep` command.
- If the marker state is off, the response is not a number (`NAN`).
- If the selected marker has no reference marker, the response is also not a number (`NAN`).

Example

The following query returns the delta Y-axis value for marker 2.

```
CALC:SWE:MARK2:YDEL?
```

Typical Response: `5.000000E+00`

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:MOVement

Syntax

```
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:MOVement
<movement characteristic>

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8:MOVement?
```

Description

Sets the marker movement characteristic of either single or in pair on the graph display in the sweep mode. The query returns the marker movement characteristic in the form of either `SING` or `PAIR`.

The description for each `<movement characteristic>` parameter is shown as follows.

`SINGle`	Move only the selected marker on the graph.
`PAIR`	Move both the selected and reference markers in the same direction on the graph.

Parameter

Item	Type	Range of values	Default value
movement characteristic	Discrete	SINGle or PAIR	SINGle

Remarks

- This command is only applicable after the sweep data has been acquired using the `INITiate[:IMMediate]:SWEep` command.
- To move the markers in pairs, you need to specify the reference marker of the selected marker prior to sending this command.

Examples

The following command sequence sets marker 1 and marker 2 as its reference marker to move together on the graph.

```
CALC:SWE:MARK1:REF M2
```

```
CALC:SWE:MARK1:MOV PAIR
```

The following query returns the movement characteristic of marker 1.

```
CALC:SWE:MARK1:MOV?
```

Typical Response: `PAIR`

CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8[:SET]:MODE

Syntax

```
CALCulate:SWEep:MARKer[1]|2|3|4|5|6|7|8[:SET]:
MODE <marker mode>
```

Description

Positions the marker at either the start, stop, or center points of the graph in the sweep mode. You may also expand the area between the selected marker and its reference marker.

The description for each <marker mode> parameter is shown as follows.

STARt	Position the marker at the graph start point.
STOP	Position the marker at the graph stop point.
CENTer	Position the marker at the graph center point.
DSPan	Expand the area of the graph between the selected marker and its reference marker.

Parameter

Item	Type	Range of values	Default value
marker mode	Discrete	STARt, STOP, CENTer, or DSPan	Required parameter

Remarks

- This command is only applicable after the sweep data has been acquired using the INITiate[:IMMediate]:SWEep command.
- The DSPan mode is only applicable for a selected marker which has a reference marker.

Example

The following command positions marker 2 at the graph start point.

```
CALC:SWE:MARK2:MODE STAR
```

Data Subsystem

The Data subsystem provides the commands which enable you to download the user-defined data for sweep, arbitrary waveform, or filter into the U8903A internal memory.

DATA:SWEep

Syntax

```
DATA:SWEep <data>
```

Description

Downloads the 32-bit floating point sweep data into the U8903A internal sweep memory. The `<data>` parameter is in the IEEE-488.2 binary block program data format.

Remarks

- Refer to "Appendix F: Using the IEEE-488.2 Binary Block Format" on page 245 for details on the `<data>` format.
- The maximum number of allowable sweep points is 1024.
- Refer to "Performing sweep" on page 207 for the example of the List sweep mode.
- Sending the `SYSTem:PRESet`, `*RST`, or `SYSTem:RESet[:MODE] SWEep` command, or cycling the U8903A power, will delete the downloaded sweep points.
- If you select amplitude as the sweep parameter, your downloaded amplitude points are assumed in unit Vp.

Example

The following command downloads the sweep data points into the U8903A internal sweep memory.

```
DATA:SWE <data>
```

DATA:WAVeform

Syntax

```
DATA:WAVeform <Vpeak>, <DC Offset>, <data>
```

Description

Downloads the 32-bit floating point arbitrary waveform data into the U8903A internal waveform memory.

You can download from 32 to 32768 (32K) points per waveform. The data value must be the normalized data between –1 to 1. The values of –1 and +1 correspond to the peak values of the waveform (if the offset is 0 V). For example, if you set the Vpeak to 5 Vp (0 V offset), +1 corresponds to +5 Vp. The `<data>` parameter is in the IEEE-488.2 binary block program data format.

Parameters

Item	Type	Range of values	Default value
Vpeak	Numeric	• 0 to 22.6 Vp (Balanced output connection) • 0 to 11.3 Vp (Unbalanced or Common output connection)	Required parameter
DC Offset	Numeric	–11.3 V to 11.3 V	Required parameter

Remarks

- Refer to "Appendix F: Using the IEEE-488.2 Binary Block Format" on page 245 for details on the `<data>` format.
- Refer to "Generating the arbitrary waveform" on page 203 for the arbitrary waveform example.
- The `DATA:WAVeform` command overwrites the previous waveform data in the U8903A volatile memory.
- Sending the `SYSTem:PRESet`, `*RST`, or `SYSTem:RESet[:MODE] AGENerator` command, or cycling the U8903A power, will delete the downloaded waveform data.

Example

The following command downloads the arbitrary waveform data into the U8903A internal waveform memory.

```
DATA:WAV 5, 0, <data>
```

DATA:FILTer

Syntax

```
DATA:FILTer <filter category>, <no. of section>,
<no. of group delay>, <data>
```

Description

Downloads the 32-bit floating point user-defined filter data into the U8903A volatile memory allocated for the user-defined filter coefficients.

NOTE There is only one memory slot allocated for this function.

The <data> parameter represents the filter coefficients in the IEEE-488.2 binary block program data format, where the minimum number of bytes is 16 and maximum number of bytes is 1024. The maximum number of filter coefficients is 256 with 32-bit for each coefficient.

Parameters

Item	Type	Range of values	Default value
filter category	Numeric	Infinite Impulse Response (IIR) or Finite Impulse Response (FIR)	Required parameter
no. of section	Numeric	1 to 36	Required parameter
no. of group delay	Numeric	0 to 65535	Required parameter

Remarks

- Refer to "Appendix F: Using the IEEE-488.2 Binary Block Format" on page 245 for details on the `<data>` format.
- This command must be sent prior to sending the `SENSe:FILTer:LPASs CUSTom`, `SENSe:FILTer:HPASs CUSTom`, or `SENSe:FILTer:WEIGhting CUSTom`.
- The `DATA:FILTer` command overwrites the previous filter data in the U8903A volatile memory.
- The downloaded filter data will remain in the U8903A volatile memory if the CUSTom filter is not changed to any other preset filters for the selected channel.
- Sending the `SYSTem:PRESet`, `*RST`, or `SYSTem:RESet[:MODE] AANalyzer` command, or cycling the U8903A power, will delete the downloaded filter data.
- If the filter type is FIR, the coefficients are arranged in the following manner.

```
Coefficient[0]          //A[0]
Coefficient[1]          //A[1]
Coefficient[2]          //A[2]
Coefficient[3]          //A[3]
Coefficient[4]          //A[4]
Coefficient[5]          //A[5]
Coefficient[6]          //A[6]
   .
   .
   .
```

NOTE

The FIR filter transfer function, $H(z)$, is defined as:

$$H(z) = A[0] + A[1]z^{-1} + A[2]z^{-2} + A[3]z^{-3} + \ldots$$

where z = complex variable

- If the filter type is IIR, the coefficients are arranged in the following manner.

Coefficient[0]	//Section 1: $Gain_1$
Coefficient[1]	//Section 1: $A_1[0]$
Coefficient[2]	//Section 1: $A_1[1]$
Coefficient[3]	//Section 1: $A_1[2]$
Coefficient[4]	//Section 1: $B_1[0]$
Coefficient[5]	//Section 1: $B_1[1]$
Coefficient[6]	//Section 1: $B_1[2]$
Coefficient[0]	//Section 2: $Gain_2$
Coefficient[1]	//Section 2: $A_2[0]$
Coefficient[2]	//Section 2: $A_2[1]$
Coefficient[3]	//Section 2: $A_2[2]$
Coefficient[4]	//Section 2: $B_2[0]$
Coefficient[5]	//Section 2: $B_2[1]$
Coefficient[6]	//Section 2: $B_2[2]$
⋮	

where A_x = Denominator and B_x = Numerator

NOTE

The IIR filter transfer function, $H(z)$, is defined as:

$$H(z) = \prod_{x=1}^{N} Gain_x \left(\frac{B_x[0] + B_x[1]z^{-1} + B_x[2]z^{-2}}{A_x[0] + A_x[1]z^{-1} + A_x[2]z^{-2}} \right)$$

where z = complex variable, N = number of sections, x = section number

- Each section must contain second-order filter coefficients.
- Refer to "Using the user-defined filter data" on page 210 for the user-defined filter example.

Example

The following command downloads the user-defined FIR low pass filter data into the U8903A volatile memory.

```
DATA:FILT FIR, 1, 0, <data>
```

Sweep Subsystem

The Sweep subsystem provides the commands to select the channel to perform sweep and sweep mode, as well as configure the sweep settings.

SOURce:SWEep:CHANnel

Syntax

```
SOURce:SWEep:CHANnel <channel>
SOURce:SWEep:CHANnel?
```

Description

Sets the channel to perform sweep. The query returns the selected sweep channel.

Parameter

Item	Type	Range of values	Default value
channel	Numeric	1 or 2	1

Remarks

- Only one channel can be swept at a time.
- The sweep channel refers to the generator channel to perform sweep.
- Refer to "Performing sweep" on page 207 for the examples on performing sweep.

NOTE

- The analyzer channel number must be the same as the generator channel number to perform sweep.
- You must not select channel 1 as the sweep channel if the generator function is variable phase, as channel 1 is the reference channel for variable phase.

Examples

The following command sets channel 1 to perform sweep.

```
SOUR:SWE:CHAN 1
```

The following query returns the sweep channel.

```
SOUR:SWE:CHAN?
```

Typical Response: `1`

SOURce:SWEep:MODE

Syntax

```
SOURce:SWEep:MODE <mode>, (@<channel>)
SOURce:SWEep:MODE? (@<channel>)
```

Description

Sets the sweep or list mode for the specified channel. The query returns the sweep mode of the selected channel. The description for each `<mode>` parameter is shown as follows.

`ASWeep`	Auto Sweep • Sweep is performed automatically. • Sweep points are based on the Start, Stop, and Step Size sweep parameter settings.
`ALISt`	Auto List • Sweep is performed automatically. • Sweep points are predefined and downloaded, or loaded from a file into the U8903A.
`MSWeep`	Manual Sweep • Sweep is performed manually. • Sweep points are based on the Start, Stop, and Step Size sweep parameter settings.

MLISt	Manual List • Sweep is performed manually. • Sweep points are predefined and downloaded, or loaded from a file into the U8903A.

Parameters

Item	Type	Range of values	Default value
mode	Discrete	ASWeep, ALISt, MSWeep, or MLISt	ASWeep
channel	Numeric	1 or 2	Required parameter

Remarks

- The selected channel refers to the generator channel to perform sweep.
- Refer to "Performing sweep" on page 207 for the examples on performing sweep.

Examples

The following command sets the sweep mode to Auto Sweep for channel 1.

```
SOUR:SWE:MODE ASW, (@1)
```

The following query returns the sweep mode of channel 1.

```
SOUR:SWE:MODE? (@1)
```

Typical Response: `ASW`

SOURce:SWEep:PARameter

Syntax

```
SOURce:SWEep:PARameter <sweep parameter>,
(@<channel>)
SOURce:SWEep:PARameter? (@<channel>)
```

Description

Sets the parameter to sweep for the specified channel. The query returns the sweep parameter of the selected channel. The description for each `<sweep parameter>` is shown as follows.

`FREQuency1`	Frequency values of the sine, dual, SMPTE IMD (Lower frequency), and square waveforms
`FREQuency2`	Frequency values of the dual, SMPTE IMD (Upper frequency), and DFD IEC 60118 (Upper frequency) waveforms
`AMPLitude`	Amplitude values of all waveform types except variable phase waveform
`PHASe`	Phase value of the variable phase waveform
`CENTer`	Center frequency value of the DFD IEC 60268 waveform

Parameters

Item	Type	Range of values	Default value
sweep parameter	Discrete	FREQuency1, FREQuency2, AMPLitude, PHASe, or CENTer	FREQuency1
channel	Numeric	1 or 2	Required parameter

Remarks

- The selected channel refers to the generator channel to perform sweep.

- Refer to "Performing sweep" on page 207 for the examples on performing sweep.

Examples

The following command sets the sweep parameter to the frequency of the sine waveform for channel 1.

```
SOUR:SWE:PAR FREQ1, (@1)
```

The following query returns the sweep parameter of channel 1.

```
SOUR:SWE:PAR? (@1)
```

Typical Response: `FREQ1`

SOURce:SWEep:SPACing

Syntax

```
SOURce:SWEep:SPACing <spacing>, (@<channel>)
SOURce:SWEep:SPACing? (@<channel>)
```

Description

Sets either linear or log interval for the sweep of the specified channel. The query returns the sweep spacing of the selected channel in the form of `LIN` or `LOG`. The description for each `<spacing>` parameter is shown as follows.

`LINear`	The sweep step size will increment or decrement the sweep point until the sweep limit is reached.
`LOGarithmic`	For nonlinear sweeps, the step size is determined by a logarithmic curve fitted between the start and stop frequency. Stepping is determined by the number of sweep points.

Parameters

Item	Type	Range of values	Default value
spacing	Discrete	LINear or LOGarithmic	• LINear (for amplitude and phase sweep) • LOGarithmic (for frequency sweep)
channel	Numeric	1 or 2	Required parameter

Remarks

- The selected channel refers to the generator channel to perform sweep.
- Refer to "Performing sweep" on page 207 for the examples on performing sweep.

Examples

The following command sets the log sweep interval for channel 1.

```
SOUR:SWE:SPAC LOG, (@1)
```

The following query returns the sweep spacing for channel 1.

```
SOUR:SWE:SPAC? (@1)
```

Typical Response: `LOG`

SOURce:SWEep:POINts

Syntax

```
SOURce:SWEep:POINts <points>, (@<channel>)
SOURce:SWEep:POINts?  (@<channel>)
```

Description

Sets the number of sweep points for the specified channel. The query returns the number of sweep points of the selected channel.

The relationship between the number of points and the stop, start, and step size for linear sweep is computed as follows.

$$STEP = (STOP - STARt)/(POINts - 1)$$

The following equation shows the relationship between the number of points and the stop, start, and step size for logarithmic sweep.

$$STOP = (STARt)(STEP)^{POINts - 1}$$

If the number of points changes, the step size will also change, but span will not be affected.

$$(SPAN = STOP - STARt)$$

Parameters

Item	Type	Range of values	Default value
points	Numeric	Minimum: 2 Maximum: 1024	30
channel	Numeric	1 or 2	Required parameter

Remarks

- The number of sweep points configuration is not applicable for the Auto List or Manual List sweep mode.
- The selected channel refers to the generator channel to perform sweep.
- Refer to "Performing sweep" on page 207 for the examples on performing sweep.

Examples

The following command sets the number of sweep points to 20 for channel 1.

```
SOUR:SWE:POIN 20, (@1)
```

The following query returns the number of sweep points for channel 1.

```
SOUR:SWE:POIN? (@1)
```

Typical Response: `20`

SOURce:SWEep:DWELl

Syntax

```
SOURce:SWEep:DWELl <delay>, (@<channel>)
SOURce:SWEep:DWELl? (@<channel>)
```

Description

Sets the sweep dwell time (ms) for the specified channel. The dwell time is the delay between the start of the signal generation and the start of making the measurement. The query returns the dwell time of the selected channel in ms.

Parameters

Item	Type	Range of values	Default value
delay	Numeric	0 to 5000 ms	0
channel	Numeric	1 or 2	Required parameter

Remarks

- The selected channel refers to the generator channel to perform sweep.
- Refer to "Performing sweep" on page 207 for the examples on performing sweep.

Examples

The following command sets the dwell time to 1 s for channel 1.

```
SOUR:SWE:DWEL 1000, (@1)
```

The following query returns the dwell time of channel 1.

```
SOUR:SWE:DWEL? (@1)
```

Typical Response: `1000`

SOURce:SWEep:NEXT

Syntax

```
SOURce:SWEep:NEXT
```

Description

Jumps to the next sweep point in the Manual Sweep or Manual List sweep mode.

SOURce:SWEep:STEP

Syntax

```
SOURce:SWEep:STEP <step>, (@<channel>)
SOURce:SWEep:STEP? (@<channel>)
```

Description

Sets the step size of the linear sweep interval, or multiplier factor of the log sweep interval for the specified channel. The query returns the step size of the selected channel.

Parameter

Item	Type	Range of values	Default value
channel	Numeric	1 or 2	Required parameter

Refer to `SOURce:SWEep:POINts` for the relationship between the step size and the stop, start, and number of points for linear or logarithmic sweep. The start and stop range of values for each waveform type and sweep parameter are listed in "Appendix E: Sweep Start and Stop Range" on page 244.

Remarks

- The multiplier factor for the log interval must not be <0 or equal to 1.
- The unit for the step size of each corresponding sweep parameter is listed as follows. The returned value is also in the unit as listed.

Sweep parameter	Unit for the step size
FREQuency1	Hz
FREQuency2	Hz
AMPLitude	• Vrms • V (for the DC signal)
PHASe	°
CENTer	Hz

- The selected channel refers to the generator channel to perform sweep.
- Refer to "Performing sweep" on page 207 for the examples on performing sweep.

Examples

The following command sets the step size to 100 Hz for channel 2. (Assume that frequency is the sweep parameter)

```
SOUR:SWE:STEP 100, (@2)
```

The following query returns the step size of channel 2.

```
SOUR:SWE:STEP? (@2)
```

Typical Response: `1.000000E+02`

SOURce:SWEep:STARt

Syntax

```
SOURce:SWEep:STARt <start>, (@<channel>)
SOURce:SWEep:STARt? (@<channel>)
```

Description

Sets the sweep start point for the specified channel. The query returns the sweep start point of the specified channel.

Parameter

Item	Type	Range of values	Default value
channel	Numeric	1 or 2	Required parameter

Refer to "Appendix E: Sweep Start and Stop Range" on page 244 for the range of the start values for each waveform type and sweep parameter.

Remarks

- The unit for the start value of each corresponding sweep parameter is listed as follows. The returned value is also in the unit as listed.

Sweep parameter	Unit for the start value
FREQuency1	Hz
FREQuency2	Hz
AMPLitude	• Vrms • V (for the DC signal)
PHASe	°
CENTer	Hz

- The selected channel refers to the generator channel to perform sweep.
- Refer to "Performing sweep" on page 207 for the examples on performing sweep.

Examples

The following command sets the sweep start point to 1 kHz for channel 2. (Assume that frequency is the sweep parameter)

```
SOUR:SWE:STAR 1000, (@2)
```

The following query returns the start point of channel 2.

```
SOUR:SWE:STAR? (@2)
```

Typical Response: `1.000000E+03`

SOURce:SWEep:STOP

Syntax

```
SOURce:SWEep:STOP <stop>, (@<channel>)
SOURce:SWEep:STOP? (@<channel>)
```

Description

Sets the sweep stop point for the specified channel. The query returns the sweep stop point of the specified channel.

Parameter

Item	Type	Range of values	Default value
channel	Numeric	1 or 2	Required parameter

Refer to "Appendix E: Sweep Start and Stop Range" on page 244 for the range of the stop values for each waveform type and sweep parameter.

Remarks

- The unit for the stop value of each corresponding sweep parameter is listed as follows. The returned value is also in the unit as listed.

Sweep parameter	Unit for the stop value
FREQuency1	Hz
FREQuency2	Hz
AMPLitude	• Vrms • V (for the DC signal)
PHASe	°
CENTer	Hz

- The selected channel refers to the generator channel to perform sweep.
- Refer to "Performing sweep" on page 207 for the examples on performing sweep.

Examples

The following command sets the start point to 3 kHz for channel 2. (Assume that frequency is the sweep parameter)

```
SOUR:SWE:STOP 3000, (@2)
```

The following query returns the stop point of channel 2.

```
SOUR:SWE:STOP? (@2)
```

Typical Response: `3.000000E+03`

SOURce:SWEep:VALues?

Syntax

```
SOURce:SWEep:VALues? (@<channel>)
```

Description

Returns the values of the sweep points for the specified channel. Multiple responses are separated by commas

Parameter

Item	Type	Range of values	Default value
channel	Numeric	1 or 2	Required parameter

Remarks

- You may query the values of the sweep points after sending the `INITiate[:IMMediate]:SWEep` command to trigger the sweep.
- Refer to "Performing sweep" on page 207 for the examples on performing sweep.

Example

The following query returns the values of the sweep points of channel 2. (Assume that the sweep start point is 100 Hz, stop point is 1000 Hz, and step size is 100 Hz)

```
SOUR:SWE:VAL? (@2)
```

Typical Response:

```
1.000000E+02,2.000000E+02,3.000000E+02,
4.000000E+02,5.000000E+02,6.000000E+02,
7.000000E+02,8.000000E+02,9.000000E+02,
1.000000E+03
```

Trigger Subsystem

The Trigger subsystem provides the commands to configure the trigger source for the analyzer or graph mode, as well as the graph trigger edge.

TRIGger:ANALyzer:SOURce

Syntax

```
TRIGger:ANALyzer:SOURce <trigger source>
TRIGger:ANALyzer:SOURce?
```

Description

Sets the analyzer trigger source for the input signals. The query returns the trigger source in the form of `IMM`, `BUS`, or `EXT`.

The description for each `<trigger source>` parameter is listed as follows.

`IMMediate`	Triggers a measurement automatically without waiting for any event to occur.
`BUS`	Triggers a measurement when the `*TRG` command is received.
`EXTernal`	Triggers a measurement when the external signal source connected to the Trigger In connector provides a low-true signal to the U8903A.

Parameter

Item	Type	Range of values	Default value
trigger source	Discrete	IMMediate, BUS, or EXTernal	IMMediate

Examples

The following command sets the analyzer trigger source to External.

```
TRIG:ANAL:SOUR EXT
```

The following query returns the trigger source.

```
TRIG:ANAL:SOUR?
```

Typical Response: `EXT`

TRIGger:GRAPh:SOURce

Syntax

```
TRIGger:GRAPh:SOURce <trigger source>
TRIGger:GRAPh:SOURce?
```

Description

Sets the graph trigger source for the input signals. The query returns the trigger source in the form of either `IMM`, `EXT`, `BUS`, `CH1`, or `CH2`.

The description for each `<trigger source>` parameter is listed as follows.

`IMMediate`	Free Run
`EXTernal`	Triggers from an external source
`BUS`	Triggers from the internal bus
`CH1`	Triggers from the channel 1 input
`CH2`	Triggers from the channel 2 input

Parameter

Item	Type	Range of values	Default value
trigger source	Discrete	IMMediate, EXTernal, BUS, CH1, or CH2	IMMediate

Examples

The following command sets the graph trigger source to External.

```
TRIG:GRAP:SOUR EXT
```

The following query returns the trigger source.

```
TRIG:GRAP:SOUR?
```

Typical Response: `EXT`

TRIGger:GRAPh:SLOPe

Syntax

```
TRIGger:GRAPh:SLOPe <edge>
TRIGger:GRAPh:SLOPe?
```

Description

Sets the rising or falling edge of the signal to be triggered. The query returns the trigger edge in the form of `POS` or `NEG`.

Parameter

Item	Type	Range of values	Default value
edge	Discrete	POSitive or NEGative	POSitive

Remark

The trigger edge is only applicable for the graph trigger source of `CH1` and `CH2`, else this setting will be ignored.

Examples

The following command sets the rising edge of the signal.

```
TRIG:GRAP:SLOP POS
```

The following query returns the trigger edge.

```
TRIG:GRAP:SLOP?
```

Typical Response: `POS`

Fetch Subsystem

The Fetch subsystem provides the commands to acquire the measurement results for the analyzer, graph, or sweep mode.

FETCh[:SCALar]?

Syntax

```
FETCh[:SCALar]? <function>, (@<channel list>)
```

Description

Retrieves the measurement result for the specified measurement function and channel(s). Multiple responses are separated by commas.

The description for each `<function>` parameter is listed as follows.

`FUNC1`	Measurement result of the first measurement function
`FUNC2`	Measurement result of the second measurement function
`ALL`	Measurement results of the first and second measurement functions

Parameters

Item	Type	Range of values	Default value
function	Discrete	FUNC1, FUNC2, or ALL	FUNC1
channel list	Numeric	One or more analyzer channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- The data returned by the `FETCh?` query is the result of the last acquisition trigger. The data is valid until the next `INITiate[:IMMediate]:ANALyzer` command is sent.
- The `FETCh? ALL` query returns a sequential data format. For example, the `FETCh? ALL, (@1,2)` query returns the result of the first measurement function of channel 1, the result of the second measurement function of channel 1, the result of the first measurement function of channel 2, and the result of the second measurement function of channel 2.
- If no measurement has been taken or there is an error occured when the measurement is being taken for a particular channel, `NAN (9.91E+37)` will be returned. If there is voltage overload, `INF (9.9E+37)` will be returned.

NOTE When FETCh is queried, the measurement result will be returned in the unit as listed in "Appendix B: Units of the Measurement Function Returned Values" on page 239.

Example

The following command sequence is used to measure VDC and VAC at channel 1.

```
SENS:FUNC1 VDC, (@1)

SENS:FUNC2 VAC, (@1)

TRIG:ANAL:SOUR IMM

INIT:ANAL (@1)

FETC? FUNC1, (@1)
```

Typical Response: `8.116441E-02`

```
FETC? FUNC2, (@1)
```

Typical Response: `9.807300E-01`

FETCh:ARRay?

Syntax

```
FETCh:ARRay? (@<channel>)
```

Description

Returns an array of measurement data of the selected channel(s). The returned data is the result of the last acquisition trigger, and in the IEEE-488.2 binary block format. The data is valid until the next `INITiate[:IMMediate]:GRAPh` command is sent.

Parameter

Item	Type	Range of values	Default value
channel	Numeric	1 or 2	Required parameter

Remarks

- The channel(s) that you have selected to acquire the array of data is based on the channel(s) specified in the `INITiate[:IMMediate]:GRAPh` command.
- To plot a graph with the array of data, the X-axis points can be calculated using the following equations.
  - If time domain is the graph analysis mode, the X-axis point can be computed as follows.

$$Point\ x = x \times \left(\frac{1}{Measurement\ bandwidth}\right)$$

where x = 0, 1, 2, ...

  - If frequency domain is the graph analysis mode, the X-axis point can be computed as follows.

$$Point\ x = \left(\frac{x \times Measurement\ bandwidth}{2 \times [PointCount - 1]}\right)$$

where x = 0, 1, 2, ...

- The measurement data is returned in the unit dBV if the graph analysis mode is frequency domain (magnitude). The data is returned in the unit radian if the analysis mode is frequency domain (phase). The data is returned in the unit V if the analysis mode is time domain.

Example

The following command sequence is used to acquire an array of measurement data for channel 1 in the graph mode.

```
TRIG:GRAP:SOUR IMM
INIT:GRAP (@1)
FETC:ARR? (@1)
```

FETCh:SWEep?

Syntax

```
FETCh:SWEep?
```

Description

Returns the sweep result for the channel specified in the `SOURce:SWEep:CHANnel` command. Multiple responses are separated by commas.

Remarks

- The `INITiate[:IMMediate]:SWEep` command must be used to initiate the sweep prior to sending the `FETCh:SWEep?` query.
- The sweep result for the Vac or THD + N Level measurement function is returned in the unit dBV.
- The sweep result for the THD + N Ratio, SINAD, SNR, SMPTE IMD, DFD IEC 60118/60268, or crosstalk measurement function is returned in the unit dB.
- The sweep result for the Vdc measurement function is returned in the unit V.

- The sweep result for the phase measurement function is returned in degrees.
- Refer to "Performing sweep" on page 207 for the examples on performing sweep.

Example

The following command sequence is used to obtain the sweep result for channel 1.

```
SOUR:SWE:CHAN 1

INIT:SWE

FETC:SWE?
```

Typical Response:

```
7.800041E+04,7.800030E+04,7.377602E+04,
6.919201E+04,6.850725E+04,6.282951E+04,
6.018090E+04,5.758000E+04,5.519361E+04,…
```

Initiate Subsystem

The Initiate subsystem provides the commands to initiate the sweep as well as the analyzer measurement and graph trigger systems.

INITiate[:IMMediate]:ANALyzer

Syntax

```
INITiate[:IMMediate]:ANALyzer (@<channel list>)
```

Description

Initiates the analyzer measurement trigger system for the specified channel(s). When a measurement trigger system is initiated, an event on a selected trigger source causes the specified triggering action to occur. If the trigger system is not initiated, all triggers are ignored.

Parameter

Item	Type	Range of values	Default value
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- This is an overlapped command.
- It takes a few milliseconds for the U8903A to be ready to acquire a trigger signal after receiving this command.

- If the analyzer trigger source is set to Immediate, sending this command will cause the U8903A to take the measurement directly. If the trigger source is set to External, sending this command will cause the U8903A to start taking the measurement when the external signal is received. If the trigger source is set to Bus, sending this command will put the U8903A in the 'waiting for trigger' state until the `*TRG` command is sent. The U8903A will only start to take the measurement when the `*TRG` command is received.
- You can verify whether a measurement has completed by polling the status register value via the `STATus:OPERation:CONDition?` query. While a measurement is in progress, bit 4 of the condition register of the Standard Operation register group will be set. After the measurement has completed, bit 4 will be cleared to 0.

Example

The following command initiates the measurement trigger system on channel 1.

```
INIT:ANAL (@1)
```

INITiate[:IMMediate]:GRAPh

Syntax

```
INITiate[:IMMediate]:GRAPh (@<channel list>)
```

Description

Initiates the graph trigger system for an array of data for the specified channel(s). When a graph trigger system is initiated, an event on a selected trigger source causes the specified triggering action to occur. If the trigger system is not initiated, all triggers are ignored.

Parameter

Item	Type	Range of values	Default value
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter

Remarks

- This is an overlapped command.
- It takes a few milliseconds for the U8903A to be ready to acquire a trigger signal after receiving this command.
- You can verify whether a measurement has completed by polling the status register value via the `STATus:OPERation:CONDition?` query. While a measurement is in progress, bit 4 of the condition register of the Standard Operation register group will be set. After the measurement has completed, bit 4 will be cleared to 0.

Example

The following command initiates the graph trigger system on channel 1 to acquire an array of graph points.

```
INIT:GRAP (@1)
```

INITiate[:IMMediate]:SWEep

Syntax

```
INITiate[:IMMediate]:SWEep
```

Description

Initiates the sweep for the channel specified in the `SOURce:SWEep:CHANnel` command.

NOTE Do not perform other operations while sweep is in progress as doing so might cause unexpected results.

Remarks

- This is an overlapped command.
- The Sweep bit at the condition register of the Standard Operation register group will be set if the sweep mode is Auto Sweep or Auto List to indicate the automatic sweep is in progress, when sweep is initiated using this command. The Sweep bit is cleared when the automatic sweep has completed.

Example

The following command initiates the sweep.

```
INIT:SWE
```

Abort Subsystem

The Abort subsystem is used to cancel any triggered actions.

ABORt:ANALyzer

Syntax

```
ABORt:ANALyzer
```

Description

Cancels any initiated analyzer measurement trigger and returns the trigger state to Idle. This command can also be used to abort sweep in progress. Once the sweep is aborted, it will return to the start point.

Remark

This command will abort all the initiated triggered channels.

Example

The following command aborts the initiated analyzer measurement trigger.

```
ABOR:ANAL
```

ABORt:GRAPh

Syntax

```
ABORt:GRAPh
```

Description

Cancels any initiated measurement trigger for an array of graph points and returns the trigger state to Idle.

Remark

This command will abort all the initiated triggered channels.

Example

The following command aborts the initiated measurement trigger for the graph points.

```
ABOR:GRAP
```

Mass Memory Subsystem

The Mass Memory subsystem provides the commands to perform file maintenance and also set the U8903A power-up state.

MMEMory:LOAD

Syntax

```
MMEMory:LOAD <label>, <filename>
```

Description

Loads the 32-bit floating point data from a file into the U8903A. The `<filename>` parameter is a quoted string and the `<label>` parameter refers to an identifier for the data type to be recalled.

Parameters

Item	Type	Range of values	Default value
label	Discrete	FILTer, WAVeform, or SWEep	Required parameter
filename	String	Full file path in quoted string. For example: "\Storage 1\filter1.juf"	Required parameter

Remarks

- The file extension type for each `<label>` parameter is listed below.

```
FILTer      .juf
WAVeform    .arb
SWEep       .csv
```

- The folders in the U8903A internal storage to store the data are listed as follows.

Data	Folder
Filter data	Filter
Arbitrary waveform data	Waveform
Sweep list values	Sweep

- For the USB external flash storage, the file path must begin with `"\Storage 1\"`.

Examples

The following command loads the filter data into the U8903A from the "filter1.juf" file in the U8903A internal storage.

```
MMEM:LOAD FILT, "\Filter\filter1.juf"
```

The following command loads the arbitrary waveform data into the U8903A from the "waveform1.arb" file in your USB external flash storage.

```
MMEM:LOAD WAV, "\Storage 1\waveform1.arb"
```

The following command loads the sweep list values into the U8903A from the "mySweep.csv" file in the U8903A internal storage.

```
MMEM:LOAD SWE, "\Sweep\mySweep.csv"
```

MMEMory:STORe

Syntax

```
MMEMory:STORe <label>, <filename>
```

Description

Stores the 32-bit floating point data to a file in either the U8903A internal storage or a USB external flash storage. The `<filename>` parameter is a quoted string and the `<label>` parameter refers to an identifier for the data type to be saved.

Parameters

Item	Type	Range of values	Default value
label	Discrete	FILTer, WAVeform, or SWEep	Required parameter
filename	String	Full file path in quoted string. For example: "\Storage 1\filter1.juf"	Required parameter

Remarks

- The file extension type for each `<label>` parameter is listed below.

```
FILTer      .juf
WAVeform    .arb
SWEep       .csv
```

- The folders in the U8903A internal storage to store the data are listed as follows.

Data	Folder
Filter data	Filter
Arbitrary waveform data	Waveform
Sweep list values	Sweep

- For the USB external flash storage, the file path must begin with `"\Storage 1\"`.

Examples

The following command stores the filter data in the "filter1.juf" file into the U8903A internal storage.

```
MMEM:STOR FILT, "\Filter\filter1.juf"
```

The following command stores the arbitrary waveform data in the "waveform1.arb" file into your USB external flash storage.

```
MMEM:STOR WAV, "\Storage 1\waveform1.arb"
```

The following command stores the sweep list values in the "mySweep.csv" file into the U8903A internal storage.

```
MMEM:STOR SWE, "\Sweep\mySweep.csv"
```

MMEMory:CATalog?

Syntax

```
MMEMory:CATalog? <location>, <directory>
```

Description

Returns the memory usage information (total amount of storage currently used and free space available) in bytes and a list of files and directories in a specified parent directory. The specified parent directory can reside in the U8903A internal memory or a USB external flash storage. Multiple responses are separated by commas.

The response is in the following format:

<used_bytes_in_this_directory>,<free_bytes_on_this_disk>,

"<file_name>,<file_type>,<filesize_in_bytes>",

"<file_name>,<file_type>,<filesize_in_bytes>", ...

Parameters

Item	Type	Range of values	Default value
location	Discrete	INTernal or EXTernal	Required parameter
directory	String	The desired parent directory	Required parameter

Remark

`INTernal` indicates the U8903A internal memory and `EXTernal` indicates a USB external flash storage.

Examples

To query the existing files in the 'Filter' folder in the U8903A internal memory.

```
MMEM:CAT? INT, "\Filter"
```

To query the existing files in a USB external flash storage.

```
MMEM:CAT? EXT, "\Storage 1"
```

MMEMory:DELete

Syntax

```
MMEMory:DELete <location>, <directory>,
<filename>
```

Description

Deletes the specified file in the selected directory.

Parameters

Item	Type	Range of values	Default value
location	Discrete	INTernal or EXTernal	Required parameter
directory	String	The directory of the desired folder.	Required parameter
filename	String	Can be any letters (A to Z), numbers (0 to 9) or underscore character ("_"). Blank spaces are not allowed.	Required parameter

Remarks

- `INTernal` indicates the U8903A internal memory and `EXTernal` indicates a USB external flash storage.
- The specified file must reside in the selected folder, otherwise an error will be generated. You can verify whether the file is available in the 'Filter' folder in the U8903A internal memory using the `MMEMory:CATalog? INTernal, "\Filter"` command.

Examples

The following command deletes a file named 'MyFilter.juf' in the '\Filter' directory of the U8903A internal memory.

```
MMEM:DEL INT, "\Filter", "MyFilter.juf"
```

The following command deletes a file named 'MyFilter.juf' in the '\Storage 1' directory of a USB external flash storage.

```
MMEM:DEL EXT, "\Storage 1", "MyFilter.juf"
```

MMEMory:LOAD:STATe:PUP

Syntax

```
MMEMory:LOAD:STATe:PUP <power-up state>
MMEMory:LOAD:STATe:PUP?
```

Description

Sets the power-up state of the U8903A. Select `LAST` to load the last settings of the U8903A, which are the settings before the U8903A was turned off. Select `DEFault` to load the default settings of the U8903A. The query returns the U8903A power-up state in the form of `LAST` or `DEF`.

Parameter

Item	Type	Range of values	Default value
power-up state	Discrete	DEFault or LAST	DEFault

Examples

The following command loads the U8903A last settings upon power up.

```
MMEM:LOAD:STAT:PUP LAST
```

The following query returns the power-up state of the U8903A.

```
MMEM:LOAD:STAT:PUP?
```

Typical Response: `LAST`

MMEMory:LOAD:STATe[:MODE]

Syntax

```
MMEMory:LOAD:STATe[:MODE] <location>,
<system mode>, <filename>
```

Description

Loads the specified state file to the selected U8903A mode. The `<filename>` parameter is a quoted string and the `<location>` parameter refers to the storage location of the state file. The `<system mode>` parameter refers to the U8903A mode of either analyzer, generator, sweep, or graph.

Parameters

Item	Type	Range of values	Default value
location	Discrete	INTernal or EXTernal	Required parameter
system mode	Discrete	AANalyzer, AGENerator, SWEep, or GRAPh	Required parameter
filename	String	Full file path in quoted string for the external storage. For example: "\Storage 1\GenState1.gen" For the internal storage, only the file name and extension are required. For example: "GenState1.gen"	Required parameter

Remarks

- `INTernal` indicates the U8903A internal memory and `EXTernal` indicates a USB external flash storage.
- If the specified state file is located in the internal storage, only the file name and extension are required. However, if the specified state file is located in the USB external flash storage, the file directory must be stated in full, else an error will be generated.
- An error will be generated if the state file to be loaded is not a `<system mode>` parameter.

Examples

The following command loads a generator mode state file named ‘GenState1.gen’ from the U8903A internal memory to the generator.

```
MMEM:LOAD:STAT INT, AGEN, "GenState1.gen"
```

The following command loads the analyzer mode state file named ‘AnaState2.ana’ in the ‘\Storage 1’ directory from the USB external flash storage to the analyzer.

```
MMEM:LOAD:STAT EXT, AAN, "\Storage 1\
AnaState2.ana"
```

MMEMory:LOAD:STATe:CHANnel

Syntax

```
MMEMory:LOAD:STATe:CHANnel <location>,
<system mode>, (@<channel list>), <filename>
```

Description

Loads the specified single channel state file to the selected U8903A mode channel. The `<filename>` parameter is a quoted string and the `<location>` parameter refers to the storage location of the state file. The `<system mode>` parameter refers to the U8903A mode of either analyzer, generator, or sweep.

Parameters

Item	Type	Range of values	Default value
location	Discrete	INTernal or EXTernal	Required parameter
system mode	Discrete	AANalyzer, AGENerator, or SWEep	Required parameter
channel list	Numeric	One or more channels. • (@1) or (@2) for single channel • (@1,2) for channels 1 and 2	Required parameter
filename	String	Full file path in quoted string for the external storage. For example: "\Storage 1\GenCh1State.gen" For the internal storage, only the file name and extension are required. For example: "GenCh1State.gen"	Required parameter

Remarks

- `INTernal` indicates the U8903A internal memory and `EXTernal` indicates a USB external flash storage.
- If the specified state file is located in the internal storage, only the file name and extension are required. However, if the specified state file is located in a USB external flash storage, the file directory must be stated in full, else an error will be generated.
- The file to be loaded must be a single channel state file and included in the `<system mode>` parameter list.
- This command is not applicable for the graph mode.

Examples

The following command loads a single channel state file named 'GenCh1State.gen' from the U8903A internal memory to the generator channel 1.

```
MMEM:LOAD:STAT:CHAN INT, AGEN, (@1),
"GenCh1State.gen"
```

The following command loads a single channel state file named 'AnaCh1State.ana' in the '\Storage 1' directory from the USB external flash storage to the analyzer channel 1 and 2.

```
MMEM:LOAD:STAT:CHAN EXT, AAN, (@1,2),
"\Storage 1\AnaCh1State.ana"
```

MMEMory:STORe:STATe[:MODE]

Syntax

```
MMEMory:STORe:STATe[:MODE] <location>,
<system mode>, <filename>
```

Description

Stores the current U8903A state to a file in either the internal storage or a USB external flash storage. The `<filename>` parameter is a quoted string and the `<location>` parameter refers to the storage location of the state file. The `<system mode>` parameter refers to the U8903A mode of either analyzer, generator, sweep, or graph.

Parameters

Item	Type	Range of values	Default value
location	Discrete	INTernal or EXTernal	Required parameter
system mode	Discrete	AANalyzer, AGENerator, SWEep, or GRAPh	Required parameter
filename	String	Full file path in quoted string for the external storage. For example: "\Storage 1\GenState1.gen" For the internal storage, only the file name and extension are required. For example: "GenState1.gen"	Required parameter

Remarks

- `INTernal` indicates the U8903A internal memory and `EXTernal` indicates a USB external flash storage.
- The file extension type must follow the system mode. The file extension for the analyzer state is '.ana'. For the generator state, the file extension is '.gen', for the sweep state, the file extension is '.swe', and for the graph state, the file extension is '.gra'. An error will be generated if the file extension does not match the system mode. However, if you do not enter the file extension, the corresponding extension will be automatically applied to the file name once the file is saved.

Examples

The following command stores the current generator state to a file named 'GenState1.gen' in the U8903A internal memory.

```
MMEM:STOR:STAT INT, AGEN, "GenState1.gen"
```

The following command stores the current analyzer state to a file named 'AnaState2.ana' in the '\Storage 1' directory of a USB external flash storage.

```
MMEM:STOR:STAT EXT, AAN, "\Storage 1\
AnaState2.ana"
```

MMEMory:STORe:STATe:CHANnel

Syntax

```
MMEMory:STORe:STATe:CHANnel <location>,
<system mode>, (@<channel>), <filename>
```

Description

Stores the current U8903A single channel state to a file in either the internal storage or a USB external flash storage. The `<filename>` parameter is a quoted string and the `<location>` parameter refers to the storage location of the state file. The `<system mode>` parameter refers to the U8903A mode of either analyzer, generator, or sweep.

Parameters

Item	Type	Range of values	Default value
location	Discrete	INTernal or EXTernal	Required parameter
system mode	Discrete	AANalyzer, AGENerator, or SWEep	Required parameter
channel	Numeric	1 or 2	Required parameter
filename	String	Full file path in quoted string for the external storage. For example: "\Storage 1\GenCh1State.gen" For the internal storage, only the file name and extension are required. For example: "GenCh1State.gen"	Required parameter

Remarks

- `INTernal` indicates the U8903A internal memory and `EXTernal` indicates a USB external flash storage.
- The file extension type must follow the system mode. The file extension for the analyzer state is '.ana'. For the generator state, the file extension is '.gen', and for the sweep state, the file extension is '.swe'. An error will be generated if the file extension does not match the system mode. However, if you do not enter the file extension, the corresponding extension will be automatically applied to the file name once the file is saved.
- The stored channel state file may be loaded to any other channel but must be within the same system mode. For example, if you have stored the analyzer channel 1 state to a file named 'AnaCh1State.ana', then you may load the 'AnaCh1State.ana' file to channel 2 within the analyzer mode.
- This command is not applicable for the graph mode.

Examples

The following command stores the generator channel 1 state to a file named 'GenCh1State.gen' in the U8903A internal memory.

```
MMEM:STOR:STAT:CHAN INT, AGEN, (@1),
"GenCh1State.gen"
```

The following command stores the analyzer channel 2 state to a file named 'AnaCh2State.ana' in the '\Storage 1' directory of a USB external flash storage.

```
MMEM:STOR:STAT:CHAN EXT, AAN, (@2),
"\Storage 1\AnaCh2State.ana"
```

Status Subsystem

The Status reporting commands allow you to determine the operating condition of the U8903A at any time. Refer to "SCPI Status System" on page 18 for more information on the status registers.

STATus:PRESet

Syntax

```
STATus:PRESet
```

Description

Sets all defined bits in the status system PTR registers and clears all bits in the NTR and enable registers.

Operation register	Preset setting
STATus:OPERation:ENABle	0 – all bits disabled
STATus:OPERation:NTR	0 – all bits disabled
STATus:OPERation:PTR	32767 – all defined bits enabled
STATus:QUEStionable:ENABle	0 – all bits disabled
STATus:QUEStionable:NTR	0 – all bits disabled
STATus:QUEStionable:PTR	32767 – all defined bits enabled

Example

The following command presets the Operation enable register.

```
STAT:PRES
```

STATus:OPERation:CONDition?

Syntax

```
STATus:OPERation:CONDition?
```

Description

Queries the condition register for the Standard Operation register group and returns the binary-weighted sum of all bits set in the register. This is a read-only register and the bits are not cleared when you read the register.

Remarks

For more information on the Operation condition register, refer to "Status system diagram" on page 20. The bit definitions for the Standard Operation register are listed in "Standard Operation register" on page 21.

Example

The following query reads the condition register (bit 3 is set).

```
STAT:OPER:COND?
```

Typical Response: `8`

STATus:OPERation:ENABle

Syntax

```
STATus:OPERation:ENABle <enable value>
STATus:OPERation:ENABle?
```

Description

Enables the bits in the enable register for the Standard Operation register group. The selected bits are then reported to the Status Byte register. The query returns the binary-weighted sum of all bits set in the register.

Parameter

Item	Type	Range of values	Default value
enable value	Numeric	A decimal value which corresponds to the binary-weighted sum of the bits in the register	0

Remarks

- For more information on the Operation enable register, refer to "Status system diagram" on page 20. The bit definitions for the Standard Operation register are listed in "Standard Operation register" on page 21.
- Use the `<enable value>` parameter to specify which bits will be reported to the Status Byte register. The specified decimal value corresponds to the binary-weighted sum of the bits you wish to enable in the register. For example, to enable bit 3 (decimal value = 8) and bit 4 (decimal value = 16), the corresponding decimal value would be 24 (8 + 16).
- The clear status (`*CLS`) command will not clear the enable register but it clears all bits in the event register.
- The `STATus:PRESet` command will clear all bits in the enable register.
- The `*RST` and instrument preset (`SYSTem:PRESet`) commands have no effect on this register.

Examples

The following command enables bit 3 (decimal value = 8) in the enable register.

```
STAT:OPER:ENAB 8
```

The following query returns the bits enabled in the register.

```
STAT:OPER:ENAB?
```

Typical Response: `8`

STATus:OPERation[:EVENt]?

Syntax

```
STATus:OPERation[:EVENt]?
```

Description

Queries the event register for the Standard Operation register group and returns the binary-weighted sum of all bits set in the register. This is a read-only register and the bits are cleared when you read the register.

Remarks

- For more information on the Operation event register, refer to "Status system diagram" on page 20. The bit definitions for the Standard Operation register are listed in "Standard Operation register" on page 21.
- Once a bit is set, it remains set until cleared by reading the event register or the clear status (`*CLS`) command.
- The `*RST`, instrument preset (`SYSTem:PRESet`), and `STATus:PRESet` commands have no effect on this register.

Example

The following query reads the event register (bit 3 is set).

```
STAT:OPER?
```

Typical Response: `8`

STATus:OPERation:NTRansition

Syntax

```
STATus:OPERation:NTRansition <value>
STATus:OPERation:NTRansition?
```

Description

Sets and reads the value of the Operation Negative-Transition (NTR) register. This register serves as a polarity filter between the Operation condition and Operation event registers. When a bit in the Operation NTR register is set to 1, then a 1-to-0 transition of the corresponding bit in the Operation condition register causes that bit in the Operation event register to be set. The query returns the binary-weighted sum of all bits set in the register.

Parameter

Item	Type	Range of values	Default value
value	Numeric	A decimal value which corresponds to the binary-weighted sum of the bits in the register	Preset = 0

Remarks

- The bit definitions for the Standard Operation register are listed in "Standard Operation register" on page 21.
- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Operation condition register sets the corresponding bit in the Operation event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Operation condition register can set the corresponding bit in the Operation event register.
- The `STATus:PRESet` command will set all bits in the NTR register to 0.
- The `*RST` and instrument preset (`SYSTem:PRESet`) commands have no effect on this register.

Examples

The following command enables bits 3 and 4 (decimal value = 24) in the NTR register.

```
STAT:OPER:NTR 24
```

The following query returns the bits enabled in the register.

```
STAT:OPER:NTR?
```

Typical Response: `24`

STATus:OPERation:PTRansition

Syntax

```
STATus:OPERation:PTRansition <value>
STATus:OPERation:PTRansition?
```

Description

Sets and reads the value of the Operation Positive-Transition (PTR) register. This register serves as a polarity filter between the Operation condition and Operation event registers. When a bit in the Operation PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Operation condition register causes that bit in the Operation event register to be set. The query returns the binary-weighted sum of all bits set in the register.

Parameter

Item	Type	Range of values	Default value
value	Numeric	A decimal value which corresponds to the binary-weighted sum of the bits in the register	Preset = 32767

Remarks

- The bit definitions for the Standard Operation register are listed in "Standard Operation register" on page 21.
- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Operation condition register sets the corresponding bit in the Operation event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Operation condition register can set the corresponding bit in the Operation Event register.
- The `STATus:PRESet` command will set all bits in the PTR register to 1.
- The `*RST` and instrument preset (`SYSTem:PRESet`) commands have no effect on this register.

Examples

The following command enables bits 3 and 4 (decimal value = 24) in the PTR register.

```
STAT:OPER:PTR 24
```

The following query returns the bits enabled in the register.

```
STAT:OPER:PTR?
```

Typical Response: `24`

STATus:QUEStionable:CONDition?

Syntax

```
STATus:QUEStionable:CONDition?
```

Description

Queries the condition register for the Questionable Data register group and returns the binary-weighted sum of all bits set in the register. This is a read-only register and the bits are not cleared when you read the register.

Remarks

For more information on the Questionable condition register, refer to "Status system diagram" on page 20. The bit definitions for the Questionable Data register are listed in "Questionable Status registers" on page 21.

Example

The following query reads the condition register (bit 0 is set).

```
STAT:QUES:COND?
```

Typical Response: `1`

STATus:QUEStionable:ENABle

Syntax

```
STATus:QUEStionable:ENABle <enable value>
STATus:QUEStionable:ENABle?
```

Description

Enables the bits in the enable register for the Questionable Data register group. The selected bits are then reported to the Status Byte register. The query returns the binary-weighted sum of all bits set in the register.

Parameter

Item	Type	Range of values	Default value
enable value	Numeric	A decimal value which corresponds to the binary-weighted sum of the bits in the register	0

Remarks

- For more information on the Questionable enable register, refer to "Status system diagram" on page 20. The bit definitions for the Questionable Data register are listed in "Questionable Status registers" on page 21.
- Use the `<enable value>` parameter to specify which bits will be reported to the Status Byte register. The specified decimal value corresponds to the binary-weighted sum of the bits you wish to enable in the register.
- The clear status (`*CLS`) command will not clear the enable register but it clears all bits in the event register.
- The `STATus:PRESet` command will clear all bits in the enable register.
- The `*RST` and instrument preset (`SYSTem:PRESet`) commands have no effect on this register.

Examples

The following command enables bit 0 (decimal value = 1) in the enable register.

```
STAT:QUES:ENAB 1
```

The following query returns the bit enabled in the register.

```
STAT:QUES:ENAB?
```

Typical Response: `1`

STATus:QUEStionable[:EVENt]?

Syntax

```
STATus:QUEStionable[:EVENt]?
```

Description

Queries the event register for the Questionable Data register group and returns the binary-weighted sum of all bits set in the register. This is a read-only register and the bits are cleared when you read the register.

Remarks

- For more information on the Questionable event register, refer to "Status system diagram" on page 20. The bit definitions for the Questionable Data register are listed in "Questionable Status registers" on page 21.
- Once a bit is set, it remains set until cleared by reading the event register or the clear status (`*CLS`) command.
- The `*RST`, instrument preset (`SYSTem:PRESet`), and `STATus:PRESet` commands have no effect on this register.

Example

The following query reads the event register (bit 0 is set).

```
STAT:QUES?
```

Typical Response: `1`

STATus:QUEStionable:NTRansition

Syntax

```
STATus:QUEStionable:NTRansition <value>
STATus:QUEStionable:NTRansition?
```

Description

Sets and reads the value of the Questionable Negative-Transition (NTR) register. This register serves as a polarity filter between the Questionable condition and Questionable event registers. When a bit in the Questionable NTR register is set to 1, then a 1-to-0 transition of the corresponding bit in the Questionable condition register causes that bit in the Questionable event register to be set. The query returns the binary-weighted sum of all bits set in the register.

Parameter

Item	Type	Range of values	Default value
value	Numeric	A decimal value which corresponds to the binary-weighted sum of the bits in the register	Preset = 0

Remarks

- The bit definitions for the Questionable Data register is listed in "Questionable Status registers" on page 21.
- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Questionable condition register sets the corresponding bit in the Questionable event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Questionable condition register can set the corresponding bit in the Questionable event register.
- The `STATus:PRESet` command will set all bits in the NTR register to 0.
- The `*RST` and instrument preset (`SYSTem:PRESet`) commands have no effect on this register.

Examples

The following command enables bit 0 (decimal value = 1) in the NTR register.

```
STAT:QUES:NTR 1
```

The following query returns the bit enabled in the register.

```
STAT:QUES:NTR?
```

Typical Response: `1`

STATus:QUEStionable:PTRansition

Syntax

```
STATus:QUEStionable:PTRansition <value>
STATus:QUEStionable:PTRansition?
```

Description

Sets and reads the value of the Questionable Positive-Transition (PTR) register. This register serves as a polarity filter between the Questionable condition and Questionable event registers. When a bit in the Questionable PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Questionable condition register causes that bit in the Questionable event register to be set. The query returns the binary-weighted sum of all bits set in the register.

Parameter

Item	Type	Range of values	Default value
value	Numeric	A decimal value which corresponds to the binary-weighted sum of the bits in the register	Preset = 32767

Remarks

- The bit definitions for the Questionable Data register is listed in "Questionable Status registers" on page 21.
- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Questionable condition register sets the corresponding bit in the Questionable event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Questionable condition register can set the corresponding bit in the Questionable event register.
- The `STATus:PRESet` command will set all bits in the PTR register to 1.
- The `*RST` and instrument preset (`SYSTem:PRESet`) commands have no effect on this register.

Examples

The following command enables bit 0 (decimal value = 1) in the PTR register.

```
STAT:QUES:PTR 1
```

The following query returns the bit enabled in the register.

```
STAT:QUES:PTR?
```

Typical Response: `1`

STATus:QUEStionable:VOLTage:CONDition?

Syntax

```
STATus:QUEStionable:VOLTage:CONDition?
```

Description

Queries the condition register for the Questionable Voltage register group and returns the binary-weighted sum of all bits set in the register. This is a read-only register and the bits are not cleared when you read the register.

Remarks

For more information on the Questionable condition register, refer to "Status system diagram" on page 20. The bit definitions for the Questionable Voltage register are listed in "Questionable Status registers" on page 21.

Example

The following query reads the condition register (bit 1 is set).

```
STAT:QUES:VOLT:COND?
```

Typical Response: `2`

STATus:QUEStionable:VOLTage:ENABle

Syntax

```
STATus:QUEStionable:VOLTage:ENABle <enable value>
STATus:QUEStionable:VOLTage:ENABle?
```

Description

Enables the bits in the enable register for the Questionable Voltage register group. The selected bits are then reported to the Questionable Data register. The query returns the binary-weighted sum of all bits set in the register.

Parameter

Item	Type	Range of values	Default value
enable value	Numeric	A decimal value which corresponds to the binary-weighted sum of the bits in the register	0

Remarks

- For more information on the Questionable enable register, refer to "Status system diagram" on page 20. The bit definitions for the Questionable Voltage register are listed in "Questionable Status registers" on page 21.
- Use the `<enable value>` parameter to specify which bits will be reported to the Questionable Data register. The specified decimal value corresponds to the binary-weighted sum of the bits you wish to enable in the register. For example, to enable bit 0 (decimal value = 1) and bit 1 (decimal value = 2), the corresponding decimal value would be 3 (1 + 2).
- The clear status (`*CLS`) command will not clear the enable register but it clears all bits in the event register.

- The *RST and instrument preset (SYSTem:PRESet) commands have no effect on this register.

Examples

The following command enables bit 1 (decimal value = 2) in the enable register.

```
STAT:QUES:VOLT:ENAB 2
```

The following query returns the bit enabled in the register.

```
STAT:QUES:VOLT:ENAB?
```

Typical Response: 2

STATus:QUEStionable:VOLTage[:EVENt]?

Syntax

```
STATus:QUEStionable:VOLTage[:EVENt]?
```

Description

Queries the event register for the Questionable Voltage register group and returns the binary-weighted sum of all bits set in the register. This is a read-only register and the bits are cleared when you read the register.

Remarks

- For more information on the Questionable event register, refer to "Status system diagram" on page 20. The bit definitions for the Questionable Voltage register are listed in "Questionable Status registers" on page 21.
- Once a bit is set, it remains set until cleared by reading the event register or the clear status (*CLS) command.
- The *RST, instrument preset (SYSTem:PRESet), and STATus:PRESet commands have no effect on this register.

Example

The following query reads the event register (bit 1 is set).

```
STAT:QUES:VOLT?
```

Typical Response: 2

STATus:QUEStionable:VOLTage:NTRansition

Syntax

```
STATus:QUEStionable:VOLTage:NTRansition <value>
STATus:QUEStionable:VOLTage:NTRansition?
```

Description

Sets and reads the value of the Questionable Voltage Negative-Transition (NTR) register. This register serves as a polarity filter between the Questionable Voltage condition and Questionable Voltage event registers. When a bit in the Questionable Voltage NTR register is set to 1, then a 1-to-0 transition of the corresponding bit in the Questionable Voltage condition register causes that bit in the Questionable Voltage event register to be set. The query returns the binary-weighted sum of all bits set in the register.

Parameter

Item	Type	Range of values	Default value
value	Numeric	A decimal value which corresponds to the binary-weighted sum of the bits in the register	Preset = 0

Remarks

- The bit definitions for the Questionable Voltage register are listed in "Questionable Status registers" on page 21.
- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Questionable Voltage condition register sets the corresponding bit in the Questionable Voltage event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Questionable Voltage condition register can set the corresponding bit in the Questionable Voltage event register.
- The `STATus:PRESet` command will set all bits in the NTR register to 0.
- The `*RST` and instrument preset (`SYSTem:PRESet`) commands have no effect on this register.

Examples

The following command enables bits 0 and 1 (decimal value = 3) in the NTR register.

```
STAT:QUES:VOLT:NTR 3
```

The following query returns the bits enabled in the register.

```
STAT:QUES:VOLT:NTR?
```

Typical Response: `3`

STATus:QUEStionable:VOLTage:PTRansition

Syntax

```
STATus:QUEStionable:VOLTage:PTRansition <value>
STATus:QUEStionable:VOLTage:PTRansition?
```

Description

Sets and reads the value of the Questionable Voltage Positive-Transition (PTR) register. This register serves as a polarity filter between the Questionable Voltage condition and Questionable Voltage event registers. When a bit in the Questionable Voltage PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Questionable Voltage condition register causes that bit in the Questionable Voltage event register to be set. The query returns the binary-weighted sum of all bits set in the register.

Parameter

Item	Type	Range of values	Default value
value	Numeric	A decimal value which corresponds to the binary-weighted sum of the bits in the register	Preset = 32767

Remarks

- The bit definitions for the Questionable Voltage register are listed in "Questionable Status registers" on page 21.
- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Questionable Voltage condition register sets the corresponding bit in the Questionable Voltage event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Questionable Voltage condition register can set the corresponding bit in the Questionable Voltage event register.
- The `STATus:PRESet` command will set all bits in the PTR register to 1.
- The `*RST` and instrument preset (`SYSTem:PRESet`) commands have no effect on this register.

Examples

The following command enables bits 0 and 1 (decimal value = 3) in the PTR register.

```
STAT:QUES:VOLT:PTR 3
```

The following query returns the bits enabled in the register.

```
STAT:QUES:VOLT:PTR?
```

Typical Response: 3

Programming Examples

Generating the normal sine waveform

The following command sequence provides an example on how to generate a normal 3 kHz, 2 Vrms sine waveform on the generator channel 1.

```
SOUR:FUNC SINE, (@1)          //Sets the waveform type to sine.
SOUR:VOLT 2Vrms, (@1)         //Sets the amplitude to 2 Vrms.
SOUR:FREQ 3kHz, (@1)          //Sets the frequency to 3 kHz.
OUTP:STAT ON, (@1)            //Turns on the output.
```

Generating the multitone waveform

The following command sequence gives an example on how to generate a multitone waveform on the generator channel 1.

```
SOUR:FUNC MULT, (@1)              //Sets the waveform type to multitone.
SOUR:MULT:COUN 20, (@1)           //Sets the number of tones to 20.
SOUR:FREQ:STAR 500Hz, (@1)        //Sets the start frequency to 500 Hz.
SOUR:FREQ:MULT 50, (@1)           //Sets the multiplier to 50.
SOUR:MULT:PHAS:MODE RAND, (@1)    //Sets the phase mode to Random.
SOUR:VOLT 3Vrms, (@1)             //Sets the amplitude to 3 Vrms.
OUTP:STAT ON, (@1)                //Turns on the output.
```

Generating the arbitrary waveform

The following examples show how to generate a 3 Vp, 0 V offset arbitrary waveform on the generator channel 1.

NOTE The arbitrary data must be uploaded before setting the waveform type to arbitrary.

Example 1

The following command sequence provides an example to generate an arbitrary waveform by uploading the data stream from the PC to the U8903A.

```
DATA:WAV 3, 0, <data>          //Uploads the arbitrary data into the U8903A. The <data> parameter is in the IEEE-488.2 binary block program data format.
SOUR:FUNC ARB, (@1)            //Sets the waveform type to arbitrary on channel 1.
OUTP:STAT ON, (@1)             //Turns on the output.
```

Example 2

The following command sequence provides an example to generate an arbitrary waveform by loading the arbitrary waveform data into the U8903A from the "waveform1.arb" file in your USB external flash storage.

```
MMEM:LOAD WAV, "\Storage 1\
waveform1.arb"                 //Loads the arbitrary waveform data into the U8903A from the "waveform1.arb" file in your USB external flash storage.
SOUR:FUNC ARB, (@1)            //Sets the waveform type to arbitrary on channel 1.
OUTP:STAT ON, (@1)             //Turns on the output.
```

After generating the arbitrary waveform, if you wish to change the amplitude of the waveform from 3 Vp to 5 Vp, just send the `SOUR:VOLT 5Vp, (@1)` command to change the amplitude to 5 Vp without the need to reupload the arbitrary data. This also applies to changing the value of the DC offset.

Making basic measurements

The following command sequence provides an example on how to measure the frequency and amplitude using the analyzer.

```
SENS:FUNC1 FREQ, (@1)          //Sets the first measurement function to
                                 frequency on channel 1.
SENS:FUNC2 VAC, (@1)           //Sets the second measurement function to
                                 amplitude on channel 1.
INIT:ANAL (@1)                 //Initiates the frequency and amplitude
                                 measurements on channel 1.
FETC? FUNC1, (@1)              //Acquires the frequency measurement result.
FETC? FUNC2, (@1)              //Acquires the amplitude measurement result.
```

Measuring the crosstalk

There are two modes of crosstalk measurement comprising channel driven (DCRosstalk) and channel measured (MCRosstalk).

In the channel driven mode, the designated reference channel will be injected with the stimulus. The presence of this signal in the other channel will be measured. The crosstalk result of the channel indicates the crosstalk from the reference channel to that channel. In the channel measured mode, the designated reference channel is used to measure the crosstalk from the other channel to this channel. The crosstalk result of the channel indicates the crosstalk from the other channel to the reference channel.

To measure the crosstalk from channel 2 to channel 1, send the following command sequence.

```
SENS:REF:CHAN 2                 //Sets the reference channel to channel 2
                                  (channel driven).
SENS:FUNC2 DCR,(@1,2)           //Sets the analyzer second measurement
                                  function to crosstalk for all channels in the
                                  U8903A.
INIT:ANAL (@1)                  //Initiates the crosstalk measurement.
FETC? FUNC2 (@1)                //Acquires the measurement result.
```

Optimizing the measurement speed

This section describes the method to optimize the measurement speed or reduce the total test time of your DUT.

You may obtain the measurement results via remote programming instead of viewing the results on the LCD display on the front panel. To do this, switch to the System page before setting up your measurements. This will speed up the measurement time of your DUT. The following command sequence provides an example on how to measure the frequency and amplitude with optimized measurement speed.

```
DISP:VIEW SYST, NONE            //Sets the display to the System page.
SENS:FUNC1 FREQ, (@1)           //Sets the first measurement function to
                                  frequency on channel 1.
SENS:FUNC2 VAC, (@1)            //Sets the second measurement function to
                                  amplitude on channel 1.
INIT:ANAL (@1)                  //Initiates the frequency and amplitude
                                  measurements on channel 1.
FETC:SCAL? FUNC1, (@1)          //Acquires the frequency measurement result.
FETC:SCAL? FUNC2, (@1)          //Acquires the amplitude measurement result.
```

Measuring the FFT magnitude

The following command sequence gives you an example on how to perform the FFT magnitude measurement of the input signals.

```
DISP:ANAL:MODE MAGN          //Sets the analysis mode to FFT magnitude.
SENS:WAV:POIN 256            //Sets the acquisition length to 256.
TRIG:GRAP:SOUR IMM           //Sets the graph trigger source to Immediate.
INIT:GRAP (@1)               //Initiates the measurements on channel 1.
STAT:OPER:COND?              //Polls the status register to check if the
                               measuring operation has completed. The
                               condition register will return 0 if the
                               operation has completed.
FETC:ARR? (@1)               //Acquires the array of measurement data for
                               channel 1.
```

Measuring the FFT phase

The following command sequence gives you an example on how to perform the FFT phase measurement of the input signals.

```
DISP:ANAL:MODE PHAS          //Sets the analysis mode to FFT phase.
SENS:WAV:POIN 256            //Sets the acquisition length to 256.
TRIG:GRAP:SOUR IMM           //Sets the graph trigger source to Immediate.
INIT:GRAP (@1,2)             //Initiates the measurements on all channels.
                               For the FFT phase measurement, you need to
                               include all the U8903A channels.
STAT:OPER:COND?              //Polls the status register to check if the
                               measuring operation has completed. The
                               condition register will return 0 if the
                               operation has completed.
FETC:ARR? (@2)               //Acquires the array of measurement data for
                               channel 2.
```

Performing sweep

Example 1

To perform a frequency response analysis of your DUT, you can connect your DUT to any generator channel and the corresponding analyzer channel. In this example, channel 2 is used to perform sweep, therefore, the DUT must be connected to the generator channel 2 and analyzer channel 2.

NOTE The analyzer channel number must be the same as the generator channel number to perform sweep.

The following command sequence provides an example to perform an automatic linear sweep on a 5 Vp sine waveform on channel 2, from 100 Hz to 1000 Hz with a step size of 225 Hz and 1 s dwell time. The DUT signal amplitude is measured.

```
SOUR:FUNC SINE, (@2)          //Sets the generator waveform type to sine on channel 2.
SOUR:VOLT 5Vp, (@2)           //Sets the amplitude of the sine waveform to 5 Vp.
SOUR:SWE:CHAN 2               //Sets channel 2 to perform sweep.
SOUR:SWE:MODE ASW, (@2)       //Sets the sweep mode to Auto.
SOUR:SWE:PAR FREQ1, (@2)      //Sets the sweep parameter to frequency.
SOUR:SWE:SPAC LIN, (@2)       //Sets the spacing type to linear.
SOUR:SWE:DWEL 1000, (@2)      //Sets the dwell time to 1 s (1000 ms).
SENS:MTIM GTR                 //Sets the measurement time to Gen Track.
SOUR:SWE:STAR 100, (@2)       //Sets the sweep start value to 100 Hz.
SOUR:SWE:STOP 1000, (@2)      //Sets the sweep stop value to 1000 Hz.
SOUR:SWE:STEP 225, (@2)       //Sets the sweep step size to 225 Hz.
```

```
SENS:FUNC2 VAC, (@2)          //Sets the measurement function to VAC. For
                                sweep, the measurement functions are based
                                on the analyzer Function 2 selections.
INIT:SWE                      //Initiates the sweep.
SOUR:SWE:VAL? (@2)            //Acquires the X-axis sweep points values.
FETC:SWE?                     //Acquires the sweep result.
```

Example 2

The following command sequence provides an example to perform a manual log sweep on a 5 Vp sine waveform on channel 2, from 100 Hz to 10 kHz with a 10 ms dwell time. The signal amplitude is measured at the analyzer channel 2.

```
SOUR:FUNC SINE, (@2)          //Sets the generator waveform type to sine on
                                channel 2.
SOUR:VOLT 5Vp, (@2)           //Sets the amplitude of the sine waveform to
                                5 Vp.
SOUR:SWE:CHAN 2               //Sets channel 2 to perform sweep.
SOUR:SWE:MODE MSW, (@2)       //Sets the sweep mode to Manual.
SOUR:SWE:PAR FREQ1, (@2)      //Sets the sweep parameter to frequency.
SOUR:SWE:SPAC LOG, (@2)       //Sets the spacing type to logarithmic.
SOUR:SWE:DWEL 10, (@2)        //Sets the dwell time to 10 ms.
SENS:MTIM GTR                 //Sets the measurement time to Gen Track.
SOUR:SWE:STAR 100, (@2)       //Sets the sweep start value to 100 Hz.
SOUR:SWE:STOP 10000, (@2)     //Sets the sweep stop value to 10 kHz.
SOUR:SWE:POIN 20              //Sets the sweep points to 20.
SENS:FUNC2 VAC, (@2)          //Sets the measurement function to VAC. For
                                sweep, the measurement functions are based
                                on the analyzer Function 2 selections.
INIT:SWE                      //Initiates the sweep.
FETC:SWE?                     //Acquires the sweep result. For Manual sweep,
                                only a single result will be returned each time
                                this query is sent.
```

```
SOUR:SWE:NEXT                    //Jumps to the next sweep point.
FETC:SWE?                        //Acquires the sweep result for the current
                                   point.
```

Use the `SOUR:SWE:NEXT` command and `FETC:SWE?` query to obtain the sweep results for the rest of the 20 sweep points.

Example 3

The following command sequence provides an example on how to use the List sweep function. Assume that you wish to sweep a series of predefined frequency points with no fixed step size and to measure the amplitude of your DUT. You may perform an automatic sweep on a 5 Vp sine waveform on channel 2 with a 0 ms dwell time, using the predefined list of frequency points.

```
SOUR:FUNC SINE, (@2)             //Sets the generator waveform type to sine on
                                   channel 2.
SOUR:VOLT 5Vp, (@2)              //Sets the amplitude of the sine waveform to
                                   5 Vp.
SOUR:SWE:CHAN 2                  //Sets channel 2 to perform sweep.
SOUR:SWE:PAR FREQ1, (@2)         //Sets the sweep parameter to frequency.
DATA:SWE <data>                  //Downloads the predefined list of frequency
                                   points into the U8903A. Refer to
```

"Appendix F: Using the IEEE-488.2 Binary Block Format" on page 245 for the `<data>` parameter.

You need to select the sweep channel prior to sending the `DATA:SWEep` command.

```
SOUR:SWE:MODE ALIS, (@2)         //Sets the sweep mode to Auto List.
SOUR:SWE:DWEL 0, (@2)            //Sets the dwell time to 0 ms.
SENS:FUNC2 VAC, (@2)             //Sets the measurement function to VAC. For
                                   sweep, the measurement functions are based
                                   on the analyzer Function 2 selections.
```

```
SOUR:SWE:POIN? (@2)          //Queries the number of sweep points in the list.
SOUR:SWE:VAL? (@2)           //Acquires the X-axis sweep points values.
INIT:SWE                     //Initiates the sweep.
FETC:SWE?                    //Acquires the sweep result.
```

Using the user-defined filter data

You need to configure your custom filter data using an external software and download the filter data into the U8903A. The following command sequence provides an example on how to load an IIR low pass filter (two sections and three group delays) to the U8903A at channel 1, as well as a FIR low pass filter (one section and ten group delays) at channel 2.

```
DATA:FILT IIR, 2, 3, <data>  //Sends the IIR custom filter data to the U8903A volatile memory allocated for the user-defined filter data. Refer to "Appendix F: Using the IEEE-488.2 Binary Block Format" on page 245 for the <data> parameter.
SENS:FILT:LPAS CUST, (@1)    //Sets the low pass filter to Custom at channel 1.
DATA:FILT FIR, 1, 10, <data> //Downloads the FIR custom filter data into the U8903A.
SENS:FILT:LPAS CUST, (@2)    //Sets the low pass filter to Custom at channel 2.
```

The custom filters for channel 1 and 2 are now ready to be used in the analyzer mode.

2 SCPI Command Summary

The summary of the SCPI commands is listed in this chapter.

SCPI Command Summary

IEEE-488.2 common commands

*CLS	Clears the event registers in all register groups and also clears the error queue.
*ESE <value>	Sets the bits in the Standard Event enable register.
*ESE?	Returns the value of the Standard Event enable register.
*ESR?	Returns the value of the event register of the Standard Event group.
*IDN?	Reads the U8903A identification string which contains four comma-separated fields.
*OPC	Sets the “Operation Complete” bit (bit 0) in the Standard Event register when all pending operations have completed.
*OPC?	Sends 1 to the output buffer when all pending operations have completed.
*RST	Resets the U8903A to its factory default settings.
*SRE <value>	Enables the bits in the Status Byte enable register.
*SRE?	Returns the value of the Status Byte enable register.
*STB?	Reads the summary (condition) of the Status Byte register.
*TRG	Triggers the U8903A from the remote interface.
*TST?	Initiates an internal self-test of the U8903A and returns a pass or fail indication.
*WAI	Sets the U8903A to wait for the completion of all pending operations before executing any other command.

System commands

SYSTem:ERRor[:NEXT]?	Returns the error number and its corresponding message string from the U8903A error queue.
SYSTem:DATE <yyyy>, <mm>, <dd>	Sets the date of the realtime clock in year (yyyy), month (mm), and day (dd) format.
SYSTem:TIME <hh>, <mm>, <ss>	Sets the realtime clock in hours (hh), minutes (mm), and seconds (ss).
SYSTem:VERSion?	Returns the SCPI standard version with which the U8903A is in compliance.
SYSTem:COMMunicate:GPIB [:SELF]:ADDRess <address>	Assigns the U8903A GPIB (IEEE-488) address.
SYSTem:COMMunicate:LAN: DHCP:ENABled	Enables the Dynamic Host Configuration Protocol (DHCP) for the U8903A.
SYSTem:COMMunicate:LAN: ADDRess <address>	Assigns a static Internet Protocol (IP) address for the U8903A.
SYSTem:COMMunicate:LAN: SMASk <subnet mask>	Sets the static subnet mask address.
SYSTem:COMMunicate:LAN: DGATeway <gateway>	Assigns the static default gateway address.
SYSTem:COMMunicate:LAN: HNAMe?	Queries the LAN hostname and returns an ASCII string enclosed in double quotes.
SYSTem:COMMunicate:LAN:MAC?	Reads the U8903A Media Access Control (MAC) address and returns an ASCII string enclosed in double quotes.
SYSTem:CHANnel?	Queries the available channels in the U8903A to determine if the channel hardware card is available or in good condition.
SYSTem:PRESet	Presets the U8903A to its factory default settings and deletes all user-defined files.

`SYSTem:RESet[:MODE] <system mode>`	Resets the customized settings of the specified U8903A system mode to the default settings.
`SYSTem:RESet:CHANnel <system mode>, (@<channel>)`	Resets the customized settings of the U8903A system mode for the specified channel to the default settings.

Output commands

`OUTPut:TYPE <type>, (@<channel list>)`	Sets the generator output connection for the specified channel(s).
`OUTPut:IMPedance <impedance>, (@<channel list>)`	Sets the generator output impedance for the specified channel(s).
`OUTPut:STATe <state>, (@<channel list>)`	Enables or disables the generator output for the specified channel(s).

Input commands

`INPut:TYPE <type>, (@<channel list>)`	Sets the input connection for the specified channel(s).
`INPut:COUPling <coupling>, (@<channel list>)`	Sets the analyzer AC or DC coupling for the specified channel(s).
`INPut:BANDwidth <bandwidth>`	Sets the analyzer measurement bandwidth.

Source commands

`SOURce:FUNCtion <waveform type>, (@<channel list>)`	Sets the generator waveform type for the specified channel(s).
`SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet <voltage>[<unit>], (@<channel list>)`	Sets the signal DC offset level in V for the specified channel(s).
`SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude] <voltage>[<unit>], (@<channel list>)`	Sets the signal amplitude level for the specified channel(s).
`SOURce:FREQuency[<j>][:CW] <frequency>[<unit>], (@<channel list>)`	Sets the signal frequency for the specified channel(s) in Hz.
`SOURce:FREQuency:CENTer <frequency>[<unit>], (@<channel list>)`	Sets the center frequency of the DFD IEC 60268 waveform for the specified channel(s) in Hz.
`SOURce:FREQuency:DIFFerence <frequency>[<unit>], (@<channel list>)`	Sets the frequency difference of the DFD IEC 60268 and DFD IEC 60118 waveforms for the specified channel(s) in Hz.
`SOURce:VOLTage:RATio <ratio>, (@<channel list>)`	Sets the amplitude ratio of the second component over the first component of the dual waveform for the specified channel(s) in percentage.
`SOURce:PHASe[:ADJust] <phase>, (@<channel list>)`	Sets the phase of the selected channel with reference to channel 1 in degree.
`SOURce:MULTitone:COUNt <tone count>, (@<channel list>)`	Sets the tones of the multitone waveform for the specified channel(s).
`SOURce:MULTitone:PHASe[:MODE] <type>, (@<channel list>)`	Sets the phase mode of the multitone waveform for the specified channel(s).

`SOURce:FREQuency:STARt <start frequency>[<unit>], (@<channel list>)`	Sets the start frequency of the multitone waveform for the specified channel(s) in Hz.
`SOURce:FREQuency:MULTiplier <multiplier>, (@<channel list>)`	Sets the frequency multiplier of the multitone waveform for the specified channel(s).

Sense commands

`SENSe:FUNCtion<j> <function>, (@<channel list>)`	Sets the analyzer measurement function for the specified channel(s).
`SENSe:FUNCtion<j>:UNIT <unit>, (@<channel list>)`	Specifies the unit for the measurement result (which is obtained using the `FETCh` command) of the corresponding function for the selected channel(s).
`SENSe:REFerence:CHANnel <reference channel>`	Sets the reference channel for the phase or crosstalk measurement functions.
`SENSe:VOLTage:RANGe:AUTO <mode>, (@<channel list>)`	Disables or enables autoranging for voltage measurements for the specified channel(s).
`SENSe:VOLTage:RANGe[:UPPer] <range>[<unit>], (@<channel list>)`	Sets the measurement range for voltage measurements for the specified channel(s) in V.
`SENSe:VOLTage:DETector <detector type>, (@<channel list>)`	Sets the analyzer AC level detector for the specified channel(s).
`SENSe:MTIMe <measurement time>`	Sets the analyzer measurement time. The query returns the measurement time.
`SENSe:FILTer:LPASs <low pass filter>, (@<channel list>)`	Sets the low pass filter for the specified channel(s).

`SENSe:FILTer:HPASs <high pass filter>, (@<channel list>)`	Sets the high pass filter for the specified channel(s).
`SENSe:FILTer:WEIGhting <weighting filter>, (@<channel list>)`	Sets the weighting filter for the specified channel(s).
`SENSe:REFerence:LEVel <level>, (@<channel list>)`	Sets the reference level for the specified channel(s) in V.
`SENSe:REFerence:IMPedance <impedance>, (@<channel list>)`	Sets the reference impedance for the specified channel(s) in ohms (Ω).
`SENSe:AVERaging:SYNC:POINts <number of points>`	Sets the number of points for the synchronous averaging.
`SENSe:WAVeform:POINts <number of points>`	Sets the number of data points to acquire with the `FETCh:ARRAy?` command.
`SENSe:FFT:WINDow <type>`	Sets the window function for frequency domain analysis.

Display commands

`DISPlay:ANALysis:MODE <mode>`	Sets the graph display as either time domain, frequency domain (magnitude), or frequency domain (phase).
`DISPlay[:WINDow]:GRAPh:TRACe:X:SPACing <spacing type>`	Sets the X-axis spacing as either linear or log.
`DISPlay[:WINDow]:GRAPh:TRACe:X[:SCALe]:AUTO`	Performs an autoscale on the X-axis to automatically scale the graph display according to the signal each time this command is sent.

`DISPlay[:WINDow]:GRAPh:TRACe:X[:SCALe]:LEFT <minimum limit>`	Sets the value represented by the minimum (left) edge of the X-axis.
`DISPlay[:WINDow]:GRAPh:TRACe:X[:SCALe]:RIGHt <maximum limit>`	Sets the value represented by the maximum (right) edge of the X-axis.
`DISPlay[:WINDow]:GRAPh:TRACe:Y:SPACing <spacing type>`	Sets the Y-axis spacing as either linear or log.
`DISPlay[:WINDow]:GRAPh:TRACe:Y[:SCALe]:AUTO`	Performs an autoscale on the Y-axis to automatically scale the graph display according to the signal each time this command is sent.
`DISPlay[:WINDow]:GRAPh:TRACe:Y[:SCALe]:BOTTom <minimum limit>`	Sets the value represented by the minimum (bottom) edge of the Y-axis.
`DISPlay[:WINDow]:GRAPh:TRACe:Y[:SCALe]:TOP <maximum limit>`	Sets the value represented by the maximum (top) edge of the Y-axis.
`DISPlay[:WINDow]:GRAPh:TRACe:AUTO`	Performs an autoscale to automatically scale the graph display according to the signal each time this command is sent.
`DISPlay[:WINDow]:SWEep:TRACe:X:SPACing <spacing type>`	Sets the X-axis spacing as either linear or log for the sweep.
`DISPlay[:WINDow]:SWEep:TRACe:X[:SCALe]:AUTO`	Performs an autoscale on the X-axis of the sweep plot to automatically scale the sweep plot according to the signal each time this command is sent.
`DISPlay[:WINDow]:SWEep:TRACe:X[:SCALe]:LEFT <minimum limit>`	Sets the value represented by the minimum (left) edge of the X-axis of the sweep plot.
`DISPlay[:WINDow]:SWEep:TRACe:X[:SCALe]:RIGHt <maximum limit>`	Sets the value represented by the maximum (right) edge of the X-axis of the sweep plot.

`DISPlay[:WINDow]:SWEep:TRACe:Y:SPACing <spacing type>`	Sets the Y-axis spacing as either linear or log for the sweep.
`DISPlay[:WINDow]:SWEep:TRACe:Y[:SCALe]:AUTO`	Performs an autoscale on the Y-axis of the sweep plot to automatically scale the sweep plot according to the signal each time this command is sent.
`DISPlay[:WINDow]:SWEep:TRACe:Y[:SCALe]:BOTTom <minimum limit>`	Sets the value represented by the minimum (bottom) edge of the Y-axis of the sweep plot.
`DISPlay[:WINDow]:SWEep:TRACe:Y[:SCALe]:TOP <maximum limit>`	Sets the value represented by the maximum (top) edge of the Y-axis of the sweep plot.
`DISPlay[:WINDow]:SWEep:TRACe:AUTO`	Performs an autoscale on the sweep plot to automatically scale the sweep plot according to the signal each time this command is sent.
`DISPlay[:WINDow]:VIEW <view>, [<channel>]`	Sets the front panel LCD display type for the specified channel.
`DISPlay[:WINDow]:STATe <state>`	Enables or disables the front panel LCD backlight.

Calculate commands

`CALCulate:HARMonic:COUNt <count>`	Sets the number of signal harmonic components in the frequency domain (magnitude) display.
`CALCulate:HARMonic:FUNDamental? (@<channel>)`	Returns the signal fundamental frequency in Hz for the specified channel.
`CALCulate:HARMonic:VALue? (@<channel>)`	Returns the harmonic component results of the trace for the specified channel.
`CALCulate:HARMonic:FREQuencies? (@<channel>)`	Returns the signal harmonic frequency values for the specified channel.

`CALCulate:THDistortion? <unit>, (@<channel>)`	Returns the Total Harmonic Distortion (THD) value of the input signal in the specified unit for the selected channel.
`CALCulate:GRAPh:MARKer [1]|2|3|4|5|6|7|8:STATe <state>`	Turns on or off the selected marker on the graph display in the graph mode.
`CALCulate:GRAPh:MARKer [1]|2|3|4|5|6|7|8:TRACe <trace no>`	Assigns the marker to the trace of the specified channel on the graph display in the graph mode.
`CALCulate:GRAPh:MARKer [1]|2|3|4|5|6|7|8:X <x position>`	Sets the marker X-axis value on the graph display in the graph mode.
`CALCulate:GRAPh:MARKer [1]|2|3|4|5|6|7|8:Y?`	Returns the marker Y-axis value on the graph display in the graph mode.
`CALCulate:GRAPh:MARKer [1]|2|3|4|5|6|7|8:PEAK <direction>`	Searches for the peak value of the trace data by placing the specified marker at either the left or right peak of the graph display in the graph mode.
`CALCulate:GRAPh:MARKer [1]|2|3|4|5|6|7|8:MIN <direction>`	Searches for the minimum value of the trace data by placing the specified marker at either the left or right minimum of the graph display in the graph mode.
`CALCulate:GRAPh:MARKer:THReshold[:LEVel] <threshold level>`	Sets the threshold level that the marker can identify as a peak or minimum on the graph display in the graph mode.
`CALCulate:GRAPh:MARKer:THReshold:STATe <threshold state>`	Turns on or off the threshold on the graph display in the graph mode.
`CALCulate:GRAPh:MARKer [1]|2|3|4|5|6|7|8:REFerence <reference marker no>`	Sets the reference marker for the selected marker on the graph display in the graph mode.
`CALCulate:GRAPh:MARKer [1]|2|3|4|5|6|7|8:XDELta?`	Returns the difference in the X-axis value between the selected marker and its reference marker on the graph display in the graph mode.

CALCulate:GRAPh:MARKer [1]|2|3|4|5|6|7|8:YDELta?	Returns the difference in the Y-axis value between the selected marker and its reference marker on the graph display in the graph mode.
CALCulate:GRAPh:MARKer [1]|2|3|4|5|6|7|8:MOVement <movement characteristic>	Sets the marker movement characteristic of either single or in pair on the graph display in the graph mode.
CALCulate:GRAPh:MARKer [1]|2|3|4|5|6|7|8[:SET]: MODE <marker mode>	Positions the marker at either the start, stop, center points of the graph, or expands the area between the selected marker and its reference marker in the graph mode.
CALCulate:SWEep:MARKer [1]|2|3|4|5|6|7|8:STATe <state>	Turns on or off the selected marker on the graph display in the sweep mode.
CALCulate:SWEep:MARKer [1]|2|3|4|5|6|7|8:TRACe?	Assigns the marker to the trace of the specified channel on the graph display in the sweep mode.
CALCulate:SWEep:MARKer [1]|2|3|4|5|6|7|8:X <x position>	Sets the marker X-axis value on the graph display in the sweep mode.
CALCulate:SWEep:MARKer [1]|2|3|4|5|6|7|8:Y?	Returns the marker Y-axis value on the graph display in the sweep mode.
CALCulate:SWEep:MARKer [1]|2|3|4|5|6|7|8:PEAK <direction>	Searches for the peak value of the trace data by placing the specified marker at either the left or right peak of the graph display in the sweep mode.
CALCulate:SWEep:MARKer [1]|2|3|4|5|6|7|8:MIN <direction>	Searches for the minimum value of the trace data by placing the specified marker at either the left or right minimum of the graph display in the sweep mode.
CALCulate:SWEep:MARKer:THReshold[:LEVel] <threshold level>	Sets the threshold level that the marker can identify as a peak or minimum on the graph display in the sweep mode.
CALCulate:SWEep:MARKer:THReshold:STATe <threshold state>	Turns on or off the threshold on the graph display in the sweep mode.

`CALCulate:SWEep:MARKer [1]|2|3|4|5|6|7|8:REFerence <reference marker no>`	Sets the reference marker for the selected marker on the graph display in the sweep mode.
`CALCulate:SWEep:MARKer [1]|2|3|4|5|6|7|8:XDELta?`	Returns the difference in the X-axis value between the selected marker and its reference marker on the graph display in the sweep mode.
`CALCulate:SWEep:MARKer [1]|2|3|4|5|6|7|8:YDELta?`	Returns the difference in the Y-axis value between the selected marker and its reference marker on the graph display in the sweep mode.
`CALCulate:SWEep:MARKer [1]|2|3|4|5|6|7|8:MOVement <movement characteristic>`	Sets the marker movement characteristic of either single or in pair on the graph display in the sweep mode.
`CALCulate:SWEep:MARKer [1]|2|3|4|5|6|7|8[:SET]: MODE <marker mode>`	Positions the marker at either the start, stop, center points of the graph, or expands the area between the selected marker and its reference marker in the sweep mode.

Data commands

`DATA:SWEep <data>`	Downloads the 32-bit floating point sweep data into the U8903A internal sweep memory.
`DATA:WAVeform <Vpeak>, <DC Offset>, <data>`	Downloads the 32-bit floating point arbitrary waveform data into the U8903A internal waveform memory.
`DATA:FILTer <filter category>, <no. of section>, <no. of group delay>, <data>`	Downloads the 32-bit floating point user-defined filter data into the U8903A volatile memory allocated for the user-defined filter coefficients.

Sweep commands

SOURce:SWEep:CHANnel <channel>	Sets the channel to perform sweep.
SOURce:SWEep:MODE <mode>, (@<channel>)	Sets the sweep or list mode for the specified channel.
SOURce:SWEep:PARameter <sweep parameter>, (@<channel>)	Sets the parameter to sweep for the specified channel.
SOURce:SWEep:SPACing <spacing>, (@<channel>)	Sets either linear or log interval for the sweep of the specified channel.
SOURce:SWEep:POINts <points>, (@<channel>)	Sets the number of sweep points for the specified channel.
SOURce:SWEep:DWELl <delay>, (@<channel>)	Sets the sweep dwell time (ms) for the specified channel.
SOURce:SWEep:NEXT	Jumps to the next sweep point in the Manual Sweep or Manual List sweep mode.
SOURce:SWEep:STEP <step>, (@<channel>)	Sets the step size of the linear sweep interval, or multiplier factor of the log sweep interval for the specified channel.
SOURce:SWEep:STARt <start>, (@<channel>)	Sets the sweep start point for the specified channel.
SOURce:SWEep:STOP <stop>, (@<channel>)	Sets the sweep stop point for the specified channel.
SENSe:SWEep:VALues? (@<channel>)	Returns the values of the sweep points for the specified channel.

Trigger commands

`TRIGger:ANALyzer:SOURce <trigger source>`	Sets the analyzer trigger source for the input signals.
`TRIGger:GRAPh:SOURce <trigger source>`	Sets the graph trigger source for the input signals.
`TRIGger:GRAPh:SLOPe <edge>`	Sets the rising or falling edge of the signal to be triggered.

Fetch commands

`FETCh[:SCALar]? <function>, (@<channel list>)`	Retrieves the measurement result for the specified measurement function and channel(s).
`FETCh:ARRay? (@<channel>)`	Returns an array of measurement data of the selected channel(s).
`FETCh:SWEep?`	Returns the sweep result for the channel specified in the `SOURce:SWEep:CHANnel` command.

Initiate commands

`INITiate[:IMMediate]: ANALyzer (@<channel list>)`	Initiates the analyzer measurement trigger system for the specified channel(s).
`INITiate[:IMMediate]:GRAPh (@<channel list>)`	Initiates the graph trigger system for an array of data for the specified channel(s).
`INITiate[:IMMediate]:SWEep`	Initiates the sweep for the channel specified in the `SOURce:SWEep:CHANnel` command.

Abort commands

`ABORt:ANALyzer`	Cancels any initiated analyzer measurement trigger and returns the trigger state to Idle.
`ABORt:GRAPh`	Cancels any initiated measurement trigger for an array of graph points and returns the trigger state to Idle.

Mass Memory commands

`MMEMory:LOAD <label>, <filename>`	Loads the 32-bit floating point data from a file into the U8903A.
`MMEMory:STORe <label>, <filename>`	Stores the 32-bit floating point data to a file in either the U8903A internal storage or a USB external flash storage.
`MMEMory:CATalog? <location>, <directory>`	Returns the memory usage information (total amount of storage currently used and free space available) in bytes and a list of files and directories in a specified parent directory.
`MMEMory:DELete <location>, <directory>, <filename>`	Deletes the specified file in the selected directory.
`MMEMory:LOAD:STATe:PUP <power-up state>`	Sets the power-up state of the U8903A.
`MMEMory:LOAD:STATe[:MODE] <location>, <system mode>, <filename>`	Loads the specified state file to the selected U8903A mode.
`MMEMory:LOAD:STATe:CHANnel <location>, <system mode>, (@<channel list>), <filename>`	Loads the specified single channel state file to the selected U8903A mode channel.

`MMEMory:STORe:STATe[:MODE] <location>, <system mode>, <filename>`	Stores the current U8903A state to a file in either the internal storage or a USB external flash storage.
`MMEMory:STORe:STATe:CHANnel <location>, <system mode>, (@<channel>), <filename>`	Stores the current U8903A single channel state to a file in either the internal storage or a USB external flash storage.

Status commands

`STATus:PRESet`	Sets all defined bits in the status system PTR registers and clears all bits in the NTR and enable registers.
`STATus:OPERation:CONDition?`	Queries the condition register for the Standard Operation register group and returns the binary-weighted sum of all bits set in the register.
`STATus:OPERation:ENABle <enable value>`	Enables the bits in the enable register for the Standard Operation register group.
`STATus:OPERation[:EVENt]?`	Queries the event register for the Standard Operation register group and returns the binary-weighted sum of all bits set in the register.
`STATus:OPERation: NTRansition <value>`	Sets and reads the value of the Operation Negative-Transition (NTR) register.
`STATus:OPERation: PTRansition <value>`	Sets and reads the value of the Operation Positive-Transition (PTR) register.
`STATus:QUEStionable: CONDition?`	Queries the condition register for the Questionable Data register group and returns the binary-weighted sum of all bits set in the register.
`STATus:QUEStionable:ENABle <enable value>`	Enables the bits in the enable register for the Questionable Data register group.

`STATus:QUEStionable[:EVENt]?`	Queries the event register for the Questionable Data register group and returns the binary-weighted sum of all bits set in the register.
`STATus:QUEStionable:NTRansition <value>`	Sets and reads the value of the Questionable Negative-Transition (NTR) register.
`STATus:QUEStionable:PTRansition <value>`	Sets and reads the value of the Questionable Positive-Transition (PTR) register.
`STATus:QUEStionable:VOLTage:CONDition?`	Queries the condition register for the Questionable Voltage register group and returns the binary-weighted sum of all bits set in the register.
`STATus:QUEStionable:VOLTage:ENABle <enable value>`	Enables the bits in the enable register for the Questionable Voltage register group.
`STATus:QUEStionable:VOLTage[:EVENt]?`	Queries the event register for the Questionable Voltage register group and returns the binary-weighted sum of all bits set in the register.
`STATus:QUEStionable:VOLTage:NTRansition <value>`	Sets and reads the value of the Questionable Voltage Negative-Transition (NTR) register.
`STATus:QUEStionable:VOLTage:PTRansition <value>`	Sets and reads the value of the Questionable Voltage Positive-Transition (PTR) register.

3 Error Messages

The U8903A SCPI command errors are summarized in this chapter.

Error Messages

Error messages are created once a command error or an erroneous condition has been detected.

- Errors are retrieved in the first-in-first-out (FIFO) order. The first error returned is the first error that has been stored.
- If more than 30 errors have occurred, the last error stored in the queue (the most recent error) is replaced with **–350,"Error queue overflow"**. No additional errors are stored until you remove all the errors from the queue. If no error occurs when you read the error queue, the U8903A responds with **0,"No error"**.
- The error queue is cleared by the clear status (`*CLS`) command or when power is cycled. The errors are also cleared when you read the queue. The error queue is not cleared by a factory reset (`*RST`) command or an instrument preset (`SYSTem:PRESet`) command.
- The error string may contain up to 255 characters and consists of an error number and an error string enclosed in double quotes.

  For example: **–113,"Undefined header"**

Error list

Table 3-1 No error

Error		Description
0	No error	The queue is completely empty. Every error or event in the queue has been read or the queue has been purposely cleared by power-on, `*CLS`, and so forth.

The following table contains the list of command errors.

Table 3-2 Command error

Error		Description
–100	Command error	Generic syntax error
–101	Invalid character	An invalid character was found in the command string
–102	Syntax error	An invalid syntax was found in the command string. Check for blank spaces.
–103	Invalid separator	An invalid separator was found in the command string. Check for proper use of , ; :
–104	Data type error	A different data type than the one allowed was found in the command string
–105	GET not allowed	A Group Execute Trigger (GET) is not allowed within a command string
–108	Parameter not allowed	More parameters were received than expected
–109	Missing parameter	Fewer parameters were received than expected
–110	Command header error	An error was detected in the header
–111	Header separator error	A character that was not a valid header separator was found in the command string
–112	Program mnemonic too long	The header contains more than 12 characters
–113	Undefined header	An invalid command was received
–114	Header suffix out of range	The value of the numeric suffix is invalid
–120	Numeric data error	Generic numeric data error
–121	Invalid character in number	An invalid character for the data type was found in the command string
–123	Exponent too large	The magnitude of the exponent was larger than 32000
–124	Too many digits	The mantissa of a numeric parameter contained more than 255 digits, excluding leading zeros
–128	Numeric data not allowed	A numeric parameter was received but a character string was expected
–130	Suffix error	Generic suffix error
–131	Invalid suffix	A suffix was incorrectly specified for a numeric parameter
–134	Suffix too long	The suffix contains more than 12 characters
–138	Suffix not allowed	A suffix is not supported for this command

Table 3-2 Command error

Error		Description
–140	Character data error	Generic character data error
–141	Invalid character data	Either the character data element contains an invalid character, or the element is not valid
–144	Character data too long	The character data element contains more than 12 characters
–148	Character data not allowed	A discrete parameter was received, but a string or numeric parameter was expected
–150	String data error	Generic string data error
–151	Invalid string data	An invalid character string was received. Ensure that the string is enclosed in quotation marks.
–158	String data not allowed	A character string was received, but is not allowed for this command
–160	Block data error	Generic block data error
–161	Invalid block data	The number of data bytes sent does not match the number of bytes specified in the header
–168	Block data not allowed	Data was sent in the arbitrary block format but is not allowed for this command
–170	Expression error	Generic expression error
–171	Invalid expression data	The expression data element was invalid
–178	Expression data not allowed	Expression data element was sent but is not allowed for this command

The execution errors are listed in the table below.

Table 3-3 Execution error

Error		Description
–200	Execution error	Generic syntax error
–210	Trigger error	An error occurred during triggering
–211	Trigger Ignored	A *TRG or triggering signal was received but the trigger was ignored
–220	Parameter error	A data element related error occurred
–221	Settings conflict	A data element could not be executed due to the present U8903A state

Table 3-3 Execution error

Error		Description
–222	Data out of range	A data element could not be executed because the value was out of range
–223	Too much data	A data element was received that contains more data than the U8903A can handle
–224	Illegal parameter value	An exact value was expected but not received
–225	Out of memory	The U8903A has insufficient memory to perform the requested operation
–231	Data questionable	The measurement accuracy is questionable
–232	Invalid format	The data format or structure is inappropriate
–233	Invalid version	The version of the data format is incorrect
–240	Hardware error	The command could not be executed due to a hardware problem
–241	Hardware missing	The command could not be executed due to missing hardware, for example, an option was not installed.
–250	Mass Storage Error	Generic error relating to mass storage
–251	Missing Mass Storage	The mass storage is not available
–255	Directory Full	The specified directory is full
–256	File name not found	The selected file was not found
–257	File name error	The file name is invalid
–260	Expression execution error	An expression program data element related error occurred
–291	Out of Memory Error	The memory is not sufficient to implement the command

The list of query errors is shown in the following table.

Table 3-4 Query error

Error		Description
–400	Query error	Generic error query
–410	Query INTERRUPTED	A condition causing an interrupted query error occurred
–420	Query UNTERMINATED	A condition causing an unterminated query error occurred

Table 3-4 Query error

Error		Description
–430	Query DEADLOCKED	A condition causing a deadlocked query error occurred
–440	Query UNTERMINATED after indefinite response	A query was received in the same program message after a query indicating an indefinite response was executed

The table below contains the list of device-specific errors.

Table 3-5 Device-specific error

Error		Description
–300	Device Specific Error	This is the generic device-dependent error for devices that cannot detect more specific errors. This code indicates that only a Device-Dependent Error as defined in the IEEE-488.2, 11.5.1.1.6 has occurred.
–310	System error	The U8903A operation has not completed properly, possibly due to an abnormal hardware or firmware condition
–311	Memory error	An error was detected in the U8903A memory
–330	Self-test failed	The U8903A self-test has failed
–340	Calibration failed	The U8903A calibration has failed
–350	Error Queue Overflow	The error queue is full because more than 30 errors have occurred. No additional errors are stored until you remove errors from the queue.

The list of self-test errors are listed in the following table.

Table 3-6 Self-test error

Error		Description
601	LAN test failed	Possible cause: The LAN adapter is not found or spoiled
602	LAN device faulty	Possible cause: The LAN adapter is found but has malfunctioned
650	Display clock test failed	Possible cause: The display clock signal is not detected

Table 3-6 Self-test error

Error		Description
651	ADC and range test failed at <channel> (Range = <input range>, Freq = <frequency>, <waveform type>)	Possible cause: The test result is out of the expected range
652	Noise level test failed	Possible cause: The test result is out of the expected range
698	Self Test Time Out	Possible cause: No response within the estimated period
699	Unknown Error	Possible cause: Unknown error

Appendixes

Appendix A: Waveform Frequency Range and Default Values

Waveform	Frequency range	Default
Sine	Frequency: 5 Hz to 80 kHz	1 kHz
Dual	Frequency 1: 5 Hz to 80 kHz Frequency 2: 5 Hz to 80 kHz	1 kHz 2 kHz
Variable phase	Frequency: 5 Hz to 80 kHz	1 kHz
SMPTE IMD 1:1/ 4:1/ 10:1	Frequency 1 (Lower frequency): 40 Hz to 500 Hz Frequency 2 (Upper frequency): 2 kHz to 60 kHz	60 Hz 7 kHz
DFD IEC 60118	Difference frequency: 80 Hz to 2 kHz Frequency 2 (Upper frequency): 3 kHz to 80 kHz	80 Hz 10 kHz
DFD IEC 60268	Difference frequency: 80 Hz to 2 kHz Center frequency: 3 kHz to 79 kHz	80 Hz 10 kHz
Gaussian	–	–
Rectangular	–	–
DC	–	–
Multitone	Start frequency: 10 Hz to 9 kHz	1 kHz
Square	Frequency: 5 Hz to 30 kHz	1 kHz
Arbitrary	–	–

Appendix B: Units of the Measurement Function Returned Values

Measurement function	<unit>	Default unit
Frequency	`Hz`	`Hz`

AC voltage
DC voltage
THD + N Level
Noise level
	`V`
	`dBu`
	`dBV`
	`dBm`
	`W`
	`dBr`
	`dBg`
		`V`

THD + N Ratio SINAD SNR SMPTE IMD DFD IEC 60118 2nd order DFD IEC 60118 3rd order DFD IEC 60268 2nd order DFD IEC 60268 3rd order Crosstalk (channel driven) Crosstalk (channel measured)	`dB` `PCT` (%)	`dB`
Phase	`deg` (°)	`deg` (°)

The units can be computed using the following formulas:

<unit>	Formula
`dB`	$20 \times \log_{10}(\text{ratio})$
`dBu`	$20 \times \log_{10}\left(\frac{V}{\sqrt{0.6}}\right)$

<unit>	Formula
dBV	$20 \times \log_{10}(V)$
dBm	$10 \times \log_{10}\left(\frac{1000\,V^2}{Z_{ref}}\right)$, where Z_{ref} = reference impedance
W	$\frac{V^2}{Z_{ref}}$, where Z_{ref} = reference impedance
dBr	$20 \times \log_{10}\left(\frac{V}{V_{ref}}\right)$, where V_{ref} = reference level
dBg	$20 \times \log_{10}\left(\frac{V}{V_{gen}}\right)$, where V_{gen} = amplitude of the generator signal for a corresponding channel
PCT (%)	$100 \times$ (ratio)

Appendix C: Waveform Parameters

Waveform	Parameter	SCPI command
Sine	Frequency	SOURce:FREQuency1
	Amplitude	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]
	DC Offset	SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet
Dual	Frequency 1	SOURce:FREQuency1
	Frequency 2	SOURce:FREQuency2
	Amplitude	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]
	Ratio	SOURce:VOLTage:RATio
	DC Offset	SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet
Variable phase	Frequency	SOURce:FREQuency1
	Amplitude	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]
	Phase → 1	SOURce:PHASe[:ADJust]
SMPTE IMD 1:1/ 4:1/ 10:1	Lower Frequency	SOURce:FREQuency1
	Upper Frequency	SOURce:FREQuency2
	Amplitude	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]
	DC Offset	SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet
DFD IEC 60118	Difference Frequency	SOURce:FREQuency:DIFFerence
	Upper Frequency	SOURce:FREQuency2
	Amplitude	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]
	DC Offset	SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet
DFD IEC 60268	Difference Frequency	SOURce:FREQuency:DIFFerence
	Center Frequency	SOURce:FREQuency:CENTer
	Amplitude	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]
	DC Offset	SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet
Gaussian / Rectangular	Amplitude	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]
	DC Offset	SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet
DC	Amplitude	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]

Waveform	Parameter	SCPI command
Multitone	Start Frequency	SOURce:FREQuency:STARt
	Amplitude	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]
	Tones	SOURce:MULTitone:COUNt
	Multiplier	SOURce:FREQuency:MULTiplier
	DC Offset	SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet
	Phase Mode	SOURce:MULTitone:PHASe[:MODE]
Square	Frequency	SOURce:FREQuency1
	Amplitude	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]
Arbitrary	Amplitude	SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]
	DC Offset	SOURce:VOLTage[:LEVel][:IMMediate]:OFFSet

Appendix D: Waveform Amplitude Range

Waveform	Amplitude range	
	Unbalanced/Common output	Balanced output
Sine Dual Variable phase SMPTE IMD 1:1/ 4:1/ 10:1 DFD IEC 60118/ 60268	0 to 8 Vrms (0 to 11.3 Vp)	0 to 16 Vrms (0 to 22.6 Vp)
Gaussian	0 to 3.6 Vrms (0 to 11.3 Vp)	0 to 7.2 Vrms (0 to 22.6 Vp)
Rectangular	0 to 6.58 Vrms (0 to 11.3 Vp)	0 to 13.16 Vrms (0 to 22.6 Vp)
DC	–11.3 V to 11.3 V	–22.6 V to 22.6 V
Multitone	0 to 11.3 Vp	0 to 22.6 Vp
Square	0 to 11.3 Vrms (0 to 11.3 Vp)	0 to 22.6 Vrms (0 to 22.6 Vp)
Arbitrary	0 to 11.3 Vp	0 to 22.6 Vp

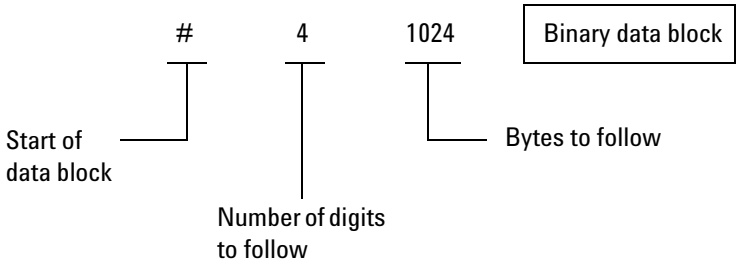
Appendix E: Sweep Start and Stop Range

<sweep parameter>	Waveform	Sweep start range	Sweep stop range
FREQuency1	Sine Dual	5 Hz to 80 kHz	5 Hz to 80 kHz
	Square	5 Hz to 30 kHz	5 Hz to 30 kHz
	SMPTE IMD 1:1/ 4:1/ 10:1	40 Hz to 500 Hz	40 Hz to 500 Hz
FREQuency2	Dual	5 Hz to 80 kHz	5 Hz to 80 kHz
	SMPTE IMD 1:1/ 4:1/ 10:1	2 kHz to 60 kHz	2 kHz to 60 kHz
	DFD IEC 60118	3 kHz to 80 kHz	3 kHz to 80 kHz
AMPLitude	Sine Dual Square Gaussian Rectangular SMPTE IMD 1:1/ 4:1/ 10:1 DFD IEC 60118/ 60268	• 0 to 22.6 Vp (Balanced output) • 0 to 11.3 Vp (Unbalanced or Common output)	• 0 to 22.6 Vp (Balanced output) • 0 to 11.3 Vp (Unbalanced or Common output)
	DC	• –22.6 V to 22.6 V (Balanced output) • –11.3 V to 11.3 V (Unbalanced or Common output)	• –22.6 V to 22.6 V (Balanced output) • –11.3 V to 11.3 V (Unbalanced or Common output)
PHASe	Variable Phase	–180 ° to 179.99 °	–180 ° to 179.99 °
CENTer	DFD IEC 60268	3 kHz to 79 kHz	3 kHz to 79 kHz

Appendix F: Using the IEEE-488.2 Binary Block Format

In the binary block format, a block header precedes the user data.

The block header has the following format:



The U8903A represents binary data as 32-bit floating points, which are sent as four bytes. Therefore, the total number of bytes is always four times the number of data points in the user data (and must always be an even number).

www.agilent.com

Contact us

To obtain service, warranty, or technical assistance, contact us at the following phone or fax numbers:

United States:
(tel) 800 829 4444 (fax) 800 829 4433
Canada:
(tel) 877 894 4414 (fax) 800 746 4866
China:
(tel) 800 810 0189 (fax) 800 820 2816
Europe:
(tel) 31 20 547 2111
Japan:
(tel) (81) 426 56 7832 (fax) (81) 426 56 7840
Korea:
(tel) (080) 769 0800 (fax) (080) 769 0900
Latin America:
(tel) (305) 269 7500
Taiwan:
(tel) 0800 047 866 (fax) 0800 286 331
Other Asia Pacific Countries:
(tel) (65) 6375 8100 (fax) (65) 6755 0042

Or visit Agilent World Wide Web at:
www.agilent.com/find/assist

Product specifications and descriptions in this document are subject to change without notice. Always refer to the Agilent web site for the latest revision.

First Edition, June 5, 2009

U8903-90027

Agilent Technologies