Keysight U8480 Series USB Thermocouple Power Sensor

Programming Guide

KEYSIGHT TECHNOLOGIES

Notices

Copyright Notice

Manual Part Number

U8481-90003

Edition

Edition 9, April 15, 2019

Printed in:

Printed in Malaysia

Published by:

Keysight Technologies
Bayan Lepas Free Industrial Zone,
11900 Penang, Malaysia

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.

Declaration of Conformity

Declarations of Conformity for this product and for other Keysight products may be downloaded from the Web. Go to http://www.keysight.com/go/conformity. You can then search by product number to find the latest Declaration of Conformity.

U.S. Government Rights

The Software is “commercial computer software,” as defined by Federal Acquisition Regulation (“FAR”) 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement (“DFARS”) 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at http://www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

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, KEYSIGHT 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. KEYSIGHT 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 KEYSIGHT 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.

Safety Information

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.

Table of Contents

List of Figures

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

List of Tables

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

1 U8480 Series Remote Operation

This chapter describes the parameters that configure the U8480 Series and helps you determine settings to optimize performance.

Introduction

This chapter contains the following sections:

Configuring the USB Interface

The USB interface requires no front panel or remote configuration.

Before connecting the USB cable, make sure that the Keysight IO Libraries software is installed on your PC.

NOTE For further information on connecting and verifying the U8480 Series via USB, refer to the *U8480 Series User's Guide*.

NOTE

- For more information on configuring the USB remote interface connectivity, refer to the *Keysight USB/LAN/GPIB Interfaces Connectivity Guide*.
- If you have installed the IO Libraries Suite, you can access the Connectivity Guide via the IO Libraries Control icon or via the Web at www.keysight.com/find/connectivity.
- If you have installed other I/O software, refer to the documentation that accompanies the software.

An Introduction to the SCPI Language

Standard Commands for Programmable Instruments (SCPI) defines how you communicate with an instrument from a bus controller. The SCPI language uses a hierarchical structure similar to the file systems used by many bus controllers. The command tree is organized with root-level commands (also called subsystems) positioned at the top, with multiple levels below each root-level command. You must specify the complete path to execute the individual lower-level commands.

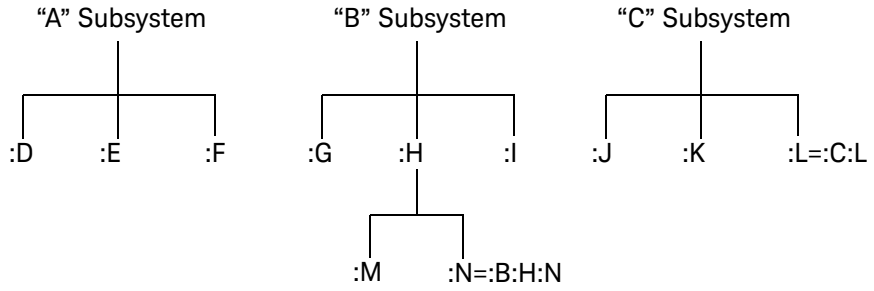


Figure 1-1 Hierarchical structure of SCPI

Mnemonic forms

Each keyword has both a long form and a short form. A standard notation is used to differentiate the short-form keyword from the long-form keyword. The long form of the keyword is shown, with the short form portion shown in upper-case characters, and the rest of the keyword shown in lower-case characters. For example, the short form of `TRIGger` is `TRIG`.

Using a colon (:)

When a colon is the first character of a command keyword, it indicates that the next command mnemonic is a root-level command. When a colon is inserted between two command mnemonics, the colon moves the path down one level in the present path (for the specified root-level command) of the command tree. You *must* separate command mnemonics from each other using a colon. *You can omit the leading colon if the command is the first of a new program line.*

Using a semicolon (;)

Use a semicolon to separate two commands within the same command string. The semicolon does not change the present path specified. For example, the following two statements are equivalent. Note that in the first statement, the first colon is optional but the third is compulsory.

```
SENS:AVER ON;SENS:AVER:COUN 1
SENS:AVER ON;AVER:COUN 1
```

Using a comma (,)

If a command requires more than one parameter, you must separate adjacent parameters using a comma.

Using whitespace

You *must* use whitespace characters, [tab] or [space], to separate a parameter from a command keyword. Whitespace characters are generally ignored *only* in parameter lists.

Using “?” commands

The bus controller may send commands at any time, but a SCPI instrument may only send responses when *specifically* instructed to do so. Only query commands (commands that end with a “?”) instruct the instrument to send a response message. Queries return either measured values or internal instrument settings.

NOTE **If you send two query commands without reading the response from the first, then attempt to read the second response, you may receive some data from the first response followed by the complete second response. To avoid this, do not send a query command without reading the response. When you cannot avoid this situation, send a device clear before sending the second query command.**

Using “*” commands

Commands starting with a “*” are called common commands. They are required to perform the identical function for *all* instruments that are compliant with the IEEE-488.2 interface standard. The “*” commands are used to control reset, self-test, and status operations in the U8480 Series.

Syntax conventions

Throughout this guide, the following conventions are used for the SCPI command syntax.

- Square brackets ([]) indicate optional keywords or parameters.
- Braces ({}) enclose one or more parameters that may be included zero or more times.
- Triangle brackets (<>) indicate that you must substitute a value for the enclosed parameter.
- Bars (|) can be read as “or” and are used to separate alternative parameter options.

Syntax diagram conventions

- Solid lines represent the recommended path.
- Ovals enclose command mnemonics. The command mnemonic must be entered exactly as shown.
- Dotted lines indicate an optional path for bypassing secondary keywords.
- Arrows and curved intersections indicate command path direction.

SCPI data types

The SCPI language defines different data formats for use in program messages and response messages. Instruments are flexible listeners and can accept commands and parameters in various formats. However, SCPI instruments are precise talkers. This means that SCPI instruments *always* respond to a particular query in a predefined, rigid format.

<boolean> definition

Throughout this document, **<boolean>** is used to represent **ON|OFF|<NRf>**. Boolean parameters have a value of 0 or 1 and are unitless. **ON** corresponds to **1** and **OFF** corresponds to **0**.

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

Queries always return a **1** or **0**, never **ON** or **OFF**.

<character_data> definition

Throughout this document, **<character_data>** is used to represent character data, that is, A-Z, a-z, 0-9, and _ (underscore). For example: START and R6_5F. The format is defined as follows:

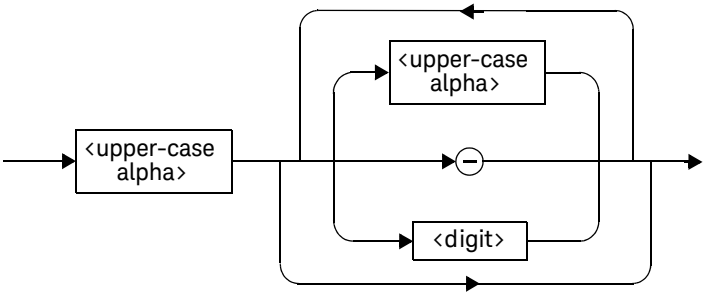


Figure 1-2 Format of <character_data>

<NAN> definition

Not a number (NAN) is represented as 9.91E37. Not a number is defined in IEEE 754.

<non-decimal numeric> definition

Throughout this document, **<non-decimal numeric>** is used to represent numeric information in bases other than ten (that is, hexadecimal, octal, and binary). The following syntax diagram shows the standard for these three data structures. For example: #HA2F, #ha4e, #Q62, #q15, #B01011.

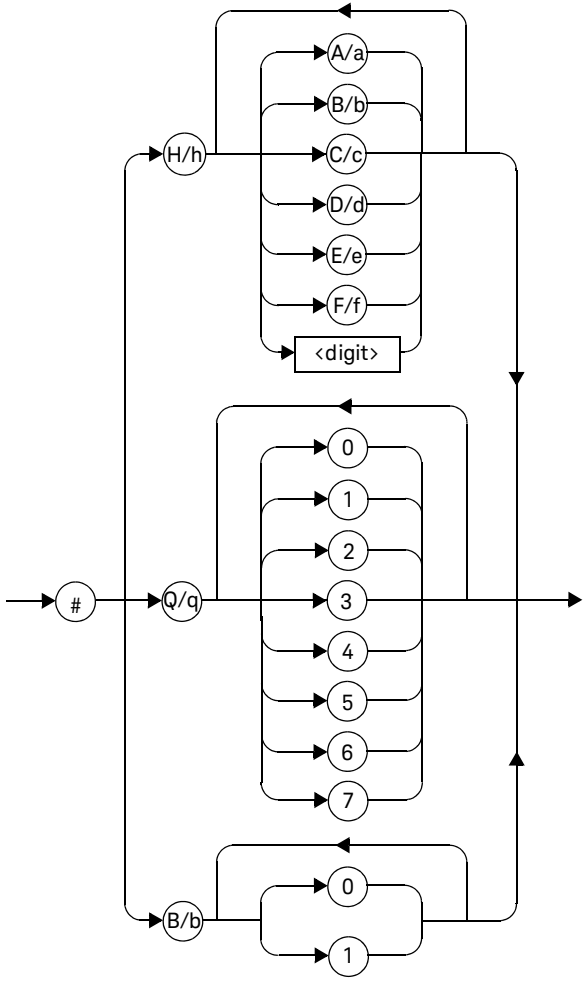


Figure 1-3 Format of <non-decimal numeric>

Refer to section 7.7.4.1 of IEEE 488.2 for further details.

<NRf> definition

Throughout this document, **<NRf>** is used to denote a flexible numeric representation. For example: +200; –56; +9.9E36. Refer to section 7.7.2.1 of IEEE 488.2 for further details.

<NR1> definition

Throughout this document, the ***<NR1>*** numeric response data is defined as:

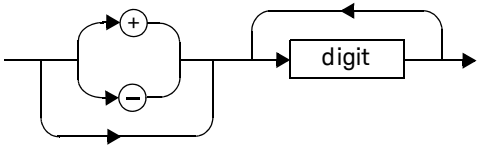


Figure 1-4 Format of <NR1>

For example:

– 146

– +146

– –12345

Refer to section 8.7.2 of IEEE 488.2 for further details.

<NR2> definition

Throughout this document, the **<NR2>** numeric response data is defined as:

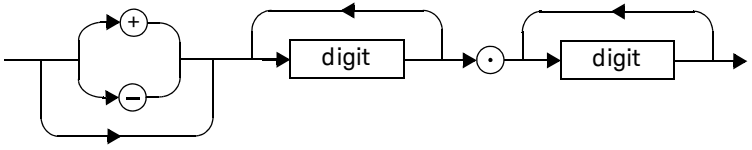


Figure 1-5 Format of <NR2>

For example:

– 12.3

– +1.2345

– –0.123

Refer to section 8.7.3 of IEEE 488.2 for further details.

<NR3> definition

Throughout this document, the **<NR3>** numeric response data is defined as:

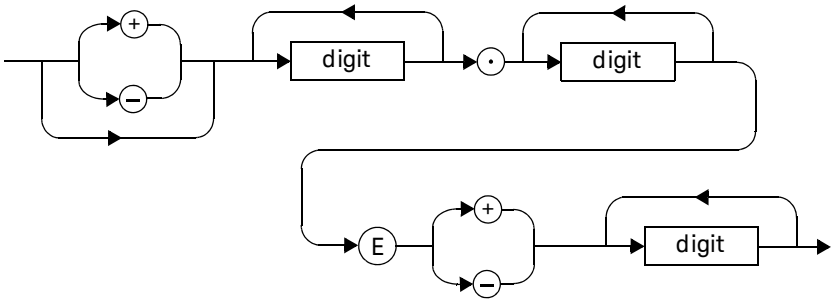


Figure 1-6 Format of <NR3>

For example:

- 1.23E+6
- 123.4E-54
- –1234.567E+90

Refer to section 8.7.4 of IEEE 488.2 for further details.

<numeric_value> definition

Throughout this document, the decimal numeric element is abbreviated to **<numeric_value>**. For example: **<NRf>**, **MINimum**, **MAXimum**, **DEFault**, or Not A Number **(NAN)**.

<string> definition

Throughout this document, **<string>** is used to represent 7-bit ASCII characters.

The format is defined as:

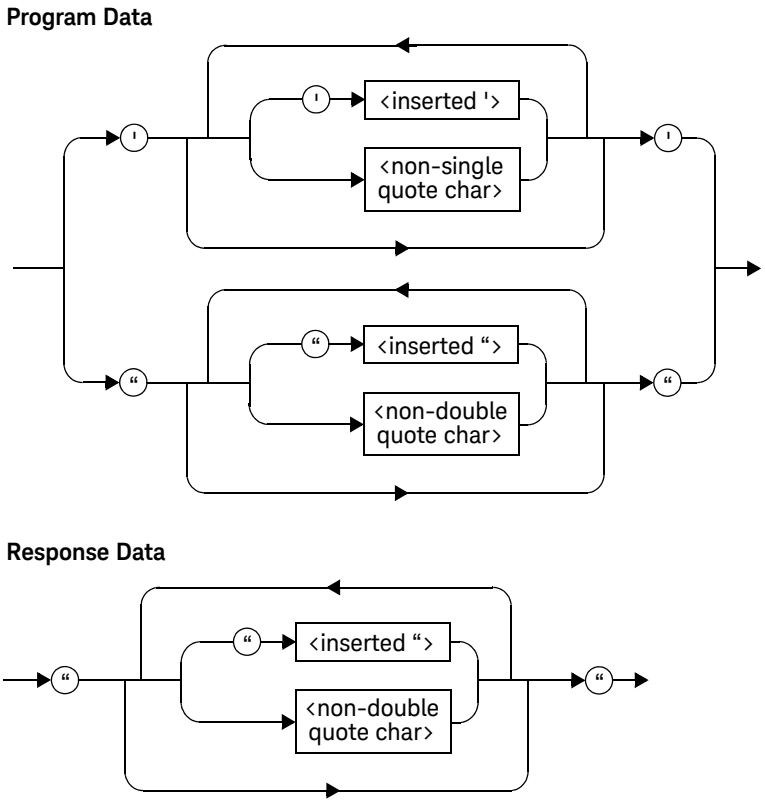


Figure 1-7 Format of <string>

Input message terminators

Program messages sent to a SCPI instrument *must* terminate with a <newline> character. The IEEE.488 EOI (end or identify) signal is interpreted as a <newline> character and may also be used to terminate a message in place of the <newline> character. A <carriage return> followed by a <newline> is also accepted. Many programming languages allow you to specify a message terminator character or EOI state to be automatically sent with each bus transaction. Message termination *always* sets the current path back to the root-level.

Zeroing and Calibrating the U8480 Series

The U8480 Series does not require manual calibration. It is equipped with a highly stable and accurate Internal Reference circuitry so that calibration can be performed without an external 50 MHz 1 mW power reference.

Zeroing must be performed on the U8480 Series without the presence of RF power at the U8480 Series input.

Zeroing

Zeroing adjusts the U8480 Series for a zero power reading. Input power to the U8480 Series must not be present while zeroing is performed.

The `CALibration[1]:ZERO:AUTO ONCE` command causes the U8480 Series to perform its zeroing routine, assuming that there is no power being applied to the U8480 Series.

Zeroing takes approximately 15 seconds to complete.

Zeroing of the U8480 Series is recommended:

- upon power up.
- when a 5 °C change in temperature occurs.
- every 24 hours.
- prior to measuring low-level signals (for example, lowest 10 dB of the dynamic range).
- when switching from or to the fast measurement mode (`SENSe:MRATe FAST`).

Calibration

The `CALibration:AUTO ONCE` command is used to calibrate the U8480 Series.

The U8480 Series performs an internal or external calibration:

- Internal calibration (`CALibration:TYPE INT`) utilizes the Internal Reference Circuitry to perform calibration, and it does not require a 50 MHz 1 mW power reference. Internal calibration is not impacted by the input power to the U8480 Series.

– External calibration (**CALibration:TYPE EXT**) enables the U8480 Series to perform calibration with a 50 MHz 1 mW power reference or a suitable power reference.

Internal calibration is the default calibration type upon power up.

Internal calibration of the U8480 Series occurs automatically:

– upon power up.

– when a 10 ^{o}C change in temperature has occurred since the last calibration.

The **CALibration:AUTO [ON|OFF|1|0]** command controls the automatic setting of the internal calibration.

Internal calibration takes approximately 1.5 s to complete, while external calibration takes approximately 15 s to complete.

Calibration sequence

You can perform a complete calibration sequence in a single query:

```
CALibration[1][:ALL]?
```

This query is only applicable for the internal calibration as the U8480 Series does not have control of the power reference in the external calibration. The calibration sequence consists of:

1 Zeroing the U8480 Series (**CALibration:ZERO:AUTO ONCE**) and

2 Calibrating the U8480 Series (**CALibration:AUTO ONCE**).

This query enters a number into the output buffer when the sequence has completed. If the result is 0, the sequence is successful. If the result is 1, the sequence has failed.

Refer to "CALCulate Command Subsystem" on page 94 for further information.

Making Measurements

The **MEASure?** query and **CONFigure** command provide a straightforward method to program the U8480 Series for measurements. You can select the measurement expected power level and resolution in one command. The U8480 Series automatically presets other measurement parameters to default values as shown in Table 1-1 below.

Table 1-1 MEASure? and CONFigure preset states

Command	MEASure? and CONFigure settings
Trigger source (**TRIGger[1]:SOURce**)	Immediate
Filter (**[SENSe[1]:]AVERage:COUNt:AUTO**)	On
Filter state (**[SENSe[1]:]AVERage[:STATe]**)	On
Trigger cycle (**INITiate[1]:CONTinuous**)	Off
Trigger delay (**TRIGger[1]:DELay:AUTO**)	On

An alternative method to program the U8480 Series is to use the lower-level commands. The advantage of using the lower-level commands over the **MEASure?** query and **CONFigure** command is that they give you more precise control of the U8480 Series. As shown in Table 1-1, the **CONFigure** command presets various states in the U8480 Series. It may be likely that you do not want to preset these states.

Using MEASure?

The simplest way to program the U8480 Series for measurements is by using the **MEASure?** query. However, this query does not offer much flexibility. When you execute the query, the U8480 Series selects the best settings for the requested configuration and immediately performs the measurement. You cannot change any setting (other than the expected power value and resolution) before the measurement is taken. This means you cannot finetune the measurement; for

example, you cannot change the filter length. To make more flexible and accurate measurements, use the **CONFigure** command. **MEASure?** is a compound command which is equivalent to an **ABORt**, followed by a **CONFigure** and a **READ?**

MEASure? examples

The following examples describe how to use the **MEASure?** query to make a measurement. These examples configure the U8480 Series for a measurement (as described in each individual example), automatically place the U8480 Series in the "wait-for-trigger" state, trigger the U8480 Series to take one reading, and then send the reading to the output buffer.

For further information on the **MEASure?** query, refer to the "Measurement Commands" on page 72.

Example 1 - The simplest method

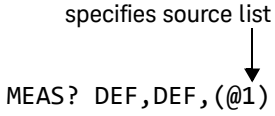
The following shows the simplest method of making measurements using **MEAS?**

```
MEAS?
```

Example 2 - Specifying the source list parameter

The **MEASure?** query has three optional parameters: an expected power value, a resolution, and a source list. These parameters must be entered in the specified order. Parameters may be defaulted from the right by omitting them, or anywhere by substituting the keyword **DEFault**. The parameter **DEFault** is used as a placeholder.

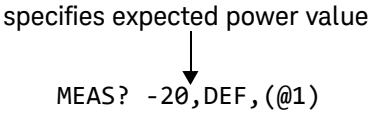
The source list parameter is used to specify a measurement channel. The U8480 Series supports only one channel. Therefore, the only valid value is (@1). The expected power and resolution parameters are set to their default values, leaving them at their current settings.



Example 3 - Specifying the expected power parameter

The previous example details the three optional parameters which can be used with the **MEASure?** query. The first optional parameter is used to enter an expected power value.

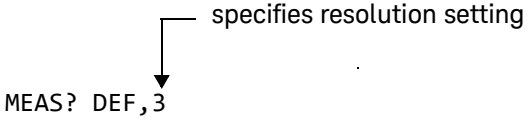
The following example uses the expected value parameter to specify a value of -20 dBm. The resolution parameter is set to its default value, leaving it at its current setting.



Example 4 - Specifying the resolution parameter

The previous examples detail the use of the expected value and source list parameters. The resolution parameter is used to set the resolution. This parameter does not affect the resolution of the data; however it does affect the auto-averaging setting (refer to "Auto-averaging mode" on page 43).

The following example uses the resolution parameter to specify a resolution setting of 3. This setting represents three significant digits if the measurement unit is W, and 0.01 dB if the unit is dBm. Refer to Chapter 2, "MEASurement Commands" on page 71 for further details on the resolution parameter. The expected power and source list parameters are set to their default values in the example. The expected power value remains unchanged at its current setting. Note that as the source list parameter is the last specified parameter, you do not have to specify **DEF**.



Using the CONFigure command

When you execute this command, the U8480 Series presets the optimum settings for the requested configuration (like the `MEASure?` query). However, the measurement is not automatically started, and you can change the measurement parameters before making measurements. This allows you to change the U8480 Series configuration from the preset conditions. The U8480 Series offers a variety of low-level commands in the `SENSe`, `CALCulate`, and `TRIGger` command subsystems. For example, if you want to change the measurement filter length, use the `[SENSe[1]:]AVERage:COUNt` command.

Use the INITiate command or the READ? query to initiate the measurement.

Using READ?

`CONFigure` does not take the measurement. One method of obtaining a result is to use the `READ?` query. The `READ?` query takes the measurement using the parameters set by the `CONFigure` command and then sends the reading to the output buffer. New data is obtained using the `READ?` query.

Using INITiate and FETCh?

`CONFigure` does not take the measurement. One method of obtaining the result is to use the `INITiate` command and `FETCh?` query. The `INITiate` command causes the measurement to be taken. The `FETCh?` query retrieves a reading when the measurement is complete and sends the reading to the output buffer. `FETCh?` can be used to retrieve the measurement results in a number of different formats without taking fresh data for each measurement.

CONFigure examples

The following examples describe how to use the `CONFigure` commands together with the `INITiate, READ?`, and `FETCh?` commands to make measurements.

For further information on the `CONFigure` commands, refer to Chapter 2, MEASurement Commands.

Example 1 - The simplest method

This example shows the simplest method of querying the measurement results.

Using READ?

`*RST`	*Resets the U8480 Series.*
`CONF`	*Configures the measurement -sets to a single measurement by default.*
`READ?`	*Initiates and retrieves the measurement.*

Using INITiate and FETCh?

`*RST`	*Resets the U8480 Series.*
`CONF`	*Configures the measurement -sets to a single measurement by default.*
`INIT`	*Sets it to wait for a trigger state.*
`FETC?`	*Triggers a measurement, and then retrieves the measurement reading.*

Example 2 - Specifying the source list parameter

The `CONFigure` command and `READ?` query have three optional parameters: an expected power value, a resolution, and a source list. These parameters must be entered in the specified order. Parameters may be defaulted from the right by omitting them, or anywhere by substituting the keyword `DEFault`. The parameter `DEFault` is used as a placeholder.

The following examples use the source list parameter to specify the measurement. The expected power and resolution parameters are set to their default values, leaving them at their current settings.

Although the `READ?` and `FETCh?` queries have three optional parameters, it is not necessary to define them as shown in these examples. If they are defined, they must be identical to those defined in the `CONFigure` command, otherwise an error will occur.

Using READ?

`ABOR`	*Aborts the measurement.*
`CONF DEF,DEF,(@1)`	*Configures the measurement to make a measurement using the current expected power and resolution settings.*
`READ?`	*Initiates and retrieves the measurement.*

Using INITiate and FETCh?

`ABOR`	*Aborts the measurement.*
`CONF DEF,DEF,(@1)`	*Configures the measurement to measure using the current expected power and resolution settings.*
`INIT`	*Sets it to wait for a trigger state.*
`FETC? DEF,DEF,(@1)`	*Triggers a measurement, and then retrieves the measurement reading.*

Example 3 - Specifying the expected power parameter

The previous example details the three optional parameters which can be used with the **CONFigure** command and **READ?** query. The first optional parameter is used to enter an expected power value.

The following example uses the expected value parameter to specify an expected power of –20 dBm. The resolution parameter is set to its default value, leaving it at its current setting.

Using READ?

`ABOR`	*Aborts the measurement.*
`CONF -20,DEF,(@1)`	*Configures the measurement to use an expected power of –20 dBm and the current resolution setting.*
`READ?`	*Initiates and retrieves the measurement.*

Some finetuning of the measurements can be performed using the **CONFigure** command and **READ?** query. For example, in the earlier program segment, some finetuning can be performed by setting the filter length to 1024 and the trigger delay off.

```
1 ABOR
2 CONF -20,DEF,(@1)
3 SENS:AVER:COUN 1024
4 TRIG:DEL:AUTO OFF
5 READ?
```

Using INITiate and FETCh?

`ABOR`	*Aborts the measurement.*
`CONF -20,DEF,(@1)`	*Configures the measurement to use an expected power of –20 dBm and the current resolution setting.*
`INIT`	*Sets it to wait for a trigger state.*
`FETC? -20,DEF,(@1)`	*Triggers a measurement, and then retrieves the measurement reading.*

Some finetuning of measurements can be carried out using the **`CONFigure`** command, **`INITiate`** command, and **`FETCh?`** query. For example, in the above program segment, some finetuning can be carried out by setting the filter length to 1024 and the trigger delay off.

```
1 ABOR
2 CONF -20,DEF,(@1)
3 SENS:AVER:COUN 1024
4 TRIG:DEL:AUTO OFF
5 INIT
6 FETC? -20,DEF,(@1)
```

Example 4 - Specifying the resolution parameter

The previous examples detail the use of the expected value and source list parameters. The resolution parameter is used to set the measurement resolution. This parameter does not affect the resolution of the data; however it does affect the auto-averaging setting.

The following example uses the resolution parameter to specify a resolution setting of 3. This setting represents three significant digits if the measurement unit is W, and 0.01 dB if the unit is dBm (for further details on the resolution parameter, refer to the commands in Chapter 2, MEASurement Commands). Also, in this example, the expected power and source list parameters are set to their default values. The expected power value is left unchanged at its current setting. Note that as the source list parameter is the last specified parameter, you do not have to specify **`DEF`**.

Using READ?

`ABOR`	*Aborts the measurement.*
`CONF DEF,3`	*Configures the measurement to use the current setting of the expected power and source list and a resolution setting of 3.*
`READ?`	*Initiates and retrieves the measurement.*

Some finetuning of the above program segment can be carried out, for example, by setting the trigger delay off, as shown below.

```
1 ABOR
2 CONF DEF,3
3 TRIG:DEL:AUTO OFF
4 READ?
```

Using INITiate and FETCh?

`ABOR`	*Aborts the measurement.*
`CONF DEF,3`	*Configures the measurement to use the current setting of the expected power and source list and a resolution setting of 3.*
`INIT`	*Sets it to wait for a trigger state.*
`FETC? DEF,3`	*Triggers a measurement, and then retrieves the measurement reading.*

Some finetuning of the above program segment can be carried out, for example, by setting the trigger delay off, as shown below.

```
1 ABOR
2 CONF DEF,3
3 TRIG:DEL:AUTO OFF
4 INIT
5 FETC? DEF,3
```

Using Frequency-Dependent Offset Tables

This section describes how to use frequency-dependent offset tables. These tables give you the ability to compensate for frequency effects in your test setup.

Overview

If the `[SENSe[1]:]CORRection:CSET2:STATe` command is `OFF`, the frequency-dependent offset tables are not used. When `[SENSe[1]:]CORRection:CSET2:STATe` is `ON`, the frequency-dependent offset tables are used, providing you with a quick and convenient method of compensating for your external test setup over a range of frequencies. Note that when selected, frequency-dependent offset correction is IN ADDITION to any correction applied for sensor frequency response. The U8480 Series is capable of storing 10 frequency-dependent offset tables of 80 frequency points each.

To use the frequency-dependent offset table:

1. Edit a frequency-dependent offset table if necessary.
2. Select the frequency-dependent offset table.
3. Enable the frequency-dependent offset table.
4. Zero and calibrate the U8480 Series.
5. Specify the frequency of the signal you want to measure. The required offset is automatically set by the U8480 Series from the frequency-dependent offset table.
6. Make the measurement.

The figure below illustrates how frequency-dependent offset tables operate.

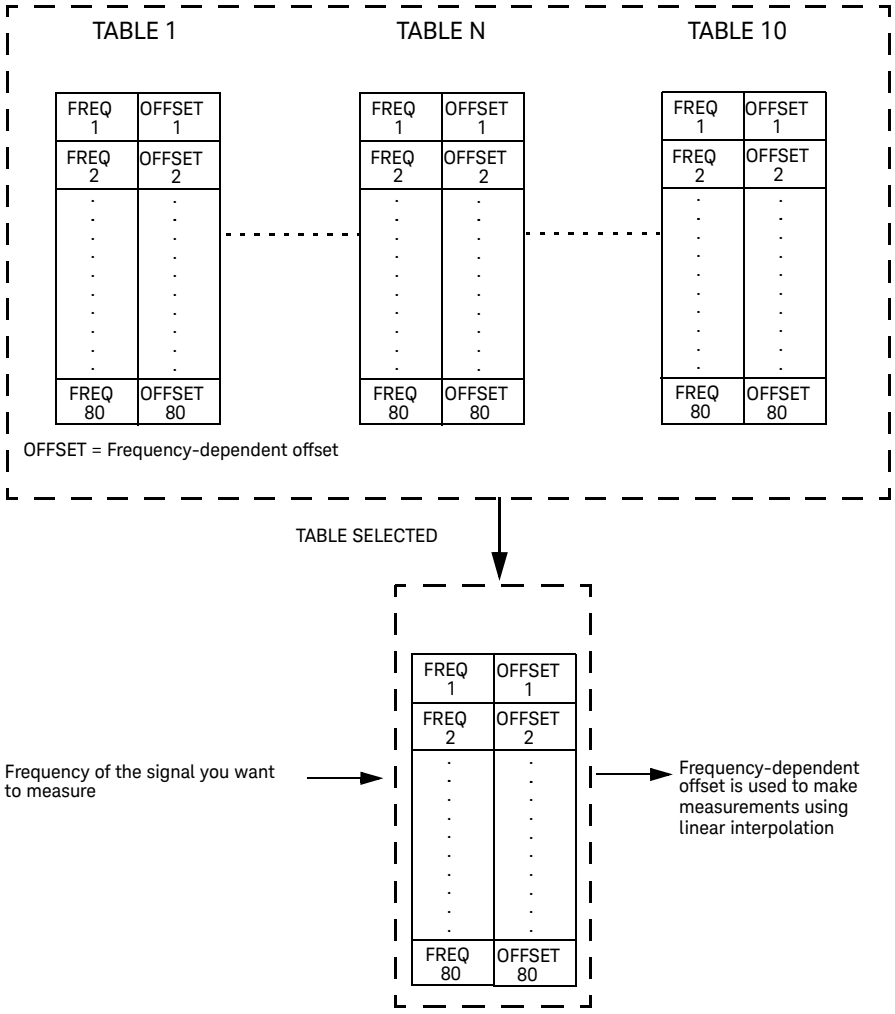


Figure 1-8 Frequency-dependent offset tables

Editing frequency-dependent offset tables

It is not possible to create any additional frequency-dependent offset tables. However, the 10 existing tables can be edited using the **MEMory** command subsystem. To do this:

1. Select one of the existing tables using
   `MEMory:TABLe:SELect <"character_data">`
   For information on naming frequency-dependent offset tables, see "Naming frequency-dependent offset tables" on page 40. For information on the current names which you can select, refer to "Listing the frequency-dependent offset table names" on page 39.
2. Enter the frequency data using
   `MEMory:TABLe:FREQuency <numeric_value>{,<numeric_value>}`
3. Enter the offset factors as shown in the table below using
   `MEMory:TABLe:GAIN <numeric_value>{,<numeric_value>}`

Frequency	Offset
Frequency 1	Offset 1
Frequency 2	Offset 2
"	"
Frequency n	Offset n

4. If required, rename the frequency-dependent offset table using `MEMory:TABLe:MOVE <"character_data">,<"character_data">`. The first `<string>` parameter identifies the existing table name, and the second identifies the new table name.

NOTE

The legal frequency unit multipliers are any of the IEEE unit multipliers, for example, KHZ, MHZ, and GHZ. If no units are specified, the U8480 Series assumes the data is Hz.

PCT is the only legal unit for offset factors and can be omitted.

The frequency and offset data must be within range. Refer to the individual commands in Chapter 8 for their specified ranges.

Ensure that the frequency points you use cover the frequency range of the signals you want to measure. If you measure a signal with a frequency outside the frequency range defined in the frequency-dependent offset table, then the U8480 Series uses the highest or lowest frequency point in the table to calculate the offset.

To make subsequent editing of a frequency-dependent offset table simpler, it is recommended that you retain a copy of your data in a program.

Listing the frequency-dependent offset table names

To list the frequency-dependent offset tables currently stored in the U8480 Series, use the following query:

```
MEMory:CATalog:TABLe?
```

The U8480 Series returns the data in the form of two numeric parameters and a string list representing all stored tables:

- `<numeric_value>,<numeric_value>{,<string>}`
  The first numeric parameter indicates the amount of memory, in bytes, used for storage of tables. The second parameter indicates the memory, in bytes, available for tables.

Each string parameter returned indicates the name, type, and size of a stored frequency-dependent offset table:

- `<string>,<type>,<size>`
  The `<string>, <type>` and `<size>` are all character data. The `<type>` is always `TABL`. The `<size>` is indicated in bytes.

For example, a sample of the response may look like:

```
560,8020,"Offset_1,TABL,220","Offset_2,TABL,340" ....
```

Naming frequency-dependent offset tables

To rename a frequency-dependent offset table use
`MEMory:TABLe:MOVE <string>,<string>`

The first `<string>` parameter identifies the existing table name, and the second identifies the new table name.

The following rules apply to frequency-dependent offset table names:

- Table names use a maximum of 12 characters.
- All characters must be upper or lower case alphabetic characters, or numeric (0-9), or an underscore (_).
- No spaces are allowed in the name.

Reviewing table data

To review the data stored in a frequency-dependent offset table, use the following command and queries:

`MEMory:TABLe:SELect "Offset1"`
Selects the frequency-dependent offset table named "Offset1".
`MEMory:TABLe:SELect?`
Returns the name of the currently selected table.
`MEMory:TABLe:FREQuency:POINts?`
Returns the number of stored frequency points.
`MEMory:TABLe:FREQuency?`
Returns the frequencies stored in the frequency-dependent offset table (in Hz).
`MEMory:TABLe:GAIN[:MAGNitude]:POINts?`
Returns the number of offset factor points stored in the frequency-dependent offset table.
`MEMory:TABLe:GAIN[:MAGNitude]?`
Returns the offset factors stored in the frequency-dependent offset table.

Modifying data

If you need to modify the frequency and offset factor data stored in a frequency-dependent offset table, you need to resend the complete data lists.

If you have retained the original data in a program, edit the program and resend the data.

Selecting a frequency-dependent offset table

After you have created the frequency-dependent offset table, you can select it using the following command:

```
[SENSe[1]:]CORRection:CSET2[:SELect] <string>
```

To find out which frequency-dependent offset table is currently selected, use the following query:

```
[SENSe[1]:]CORRection:CSET2[:SELect]?
```

Enabling a frequency-dependent offset table

To enable the frequency-dependent offset table, use the following command:

```
[SENSe[1]:]CORRection:CSET2:STATe ON
```

If you set `[SENSe[1]:]CORRection:CSET2:STATe` to `ON` and no frequency-dependent offset table is selected, error –221, “Settings conflict” occurs.

Making the measurement

To make the power measurement, set the U8480 Series for the frequency of the signal you want to measure. The U8480 Series automatically sets the offset factor. Use either `INITiate` and `FETCh?`, or `READ?` to initiate the measurement as shown in the following program segments:

INITiate example

```
ABOR
CONF DEF,1,(@1)
CORR:CSET2:SEL “Offset1”
CORR:CSET2:STAT ON
FREQ 50MHZ
INIT:IMM
FETC?
```

READ? example

```
ABOR
CONF DEF,2,(@1)
CORR:CSET2:SEL “Offset1”
CORR:CSET2:STAT ON
FREQ 50MHZ
READ?
```

NOTE

If the measurement frequency does not correspond directly to a frequency in the frequency- dependent offset table, the U8480 Series calculates the offset using linear interpolation.

If you enter a frequency outside the frequency range defined in the frequency-dependent offset table, then the U8480 Series uses the highest or lowest frequency point in the table to set the offset.

To find out the value of the offset being used by the U8480 Series to make a measurement, use the following query:
`[SENSe[1]:]CORRection:FDOFfset|GAIN4[:INPut][:MAGNitude]?`
The response may be an interpolated value.

Setting the Averaging

This section provides an overview of setting the averaging. For more detailed information on this feature, refer to the individual commands in Chapter 8, SENSe Subsystem.

Averaging

The U8480 Series has a digital filter to average power readings. The number of readings averaged can range from 1 to 1024. This filter is used to reduce noise, obtain the desired resolution, and to reduce the jitter in the measurement results. However, the time to take the measurement is increased. You can select the filter length, or you can set the U8480 Series to the auto-filter mode. To enable and disable averaging, use the following command:

```
[SENSe[1]:]AVERage[:STATe] <boolean>
```

Auto-averaging mode

To enable or disable auto-filter mode, use the following command:

```
[SENSe[1]:]AVERage:COUNt:AUTO <boolean>
```

When the auto-filter mode is enabled, the U8480 Series automatically sets the number of readings averaged together to satisfy the filtering requirements for most power measurements. The number of readings averaged together depends on the resolution and the power level currently being measured. Refer to "Auto-Averaging Settings" on page 374 for more information.

Filter length

You specify the filter length using the following command:

```
[SENSe[1]:]AVERage:COUNt <numeric_value>
```

The range of values for the filter length is 1 to 1024. Specifying this command disables automatic filter length selection. Increasing the value of the filter length reduces measurement noise. However, the time to take the measurement is increased.

Table 1-2 Settling time for normal speed, ×2 speed, and fast speed

Number of averages	Settling time[a](s) (Normal speed)	Settling time[a](s) (´2 speed)	Settling time[a](s) (Fast speed)
1	0.15	0.14	0.003
2	0.23	0.16	0.005
4	0.33	0.23	0.009
8	0.53	0.33	0.018
16	0.90	0.51	0.036
32	1.68	0.91	0.069
64	3.24	1.70	0.134
128	6.44	3.28	0.265
256	12.7	6.45	0.528
512	25.3	12.7	1.05
1024	50.5	25.3	2.10

[a] Manual filter, 10 dB decreasing power step

Setting Offsets

Channel offsets

The U8480 Series can be configured to compensate for signal loss or gain in your test setup (for example, to compensate for the loss of a 10 dB attenuator). You use the **SENSe** command subsystem to configure the U8480 Series. Gain and loss correction are a coupled system. If you enter an offset value, the state is automatically enabled. However, it can be enabled and disabled using the `[SENSe[1]:]CORRection:GAIN2:STATe <boolean>` command.

NOTE To enter a LOSS value, you can enter a negative value in the command:
`[SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude] <numeric_value>`

Setting Measurement Limits

You can configure the U8480 Series to detect when a measurement is outside of a predefined upper and/or lower limit value.

Setting limits

The U8480 Series can be configured to verify the power being measured against an upper and/or lower limit value. The range of values that can be set for lower and upper limits is –150.00 dBm to +230.00 dBm. The default upper limit is +90.00 dBm, and the default lower limit is –90.00 dBm.

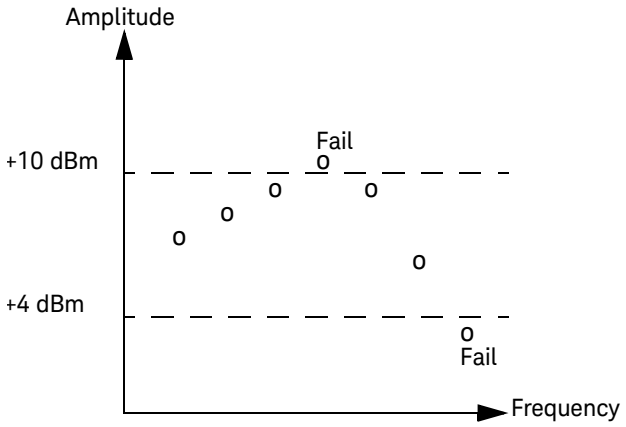


Figure 1-9 Limits checking results

The U8480 Series can be configured to verify the measurement in either Watts (W) or dBm against the predefined upper and/or lower limit values. The upper and lower limits can be set using the `CALCulate[1]:LIMit:UPPer[:DATA]` and `CALCulate[1]:LIMit:LOWer[:DATA]` commands respectively. The range of values that can be set for the limits and default values depends on the measurement unit that is currently selected (refer to Table 1-3).

Table 1-3 Range of values for measurement limits

Unit	Maximum	Minimum	Default maximum	Default minimum
dBm	230 dBm	–150 dBm	90 dBm	–90 dBm
W	1e20 W	1e–18 W	1e6 W	1e–12 W

An example of the programming sequence is shown as follows.

```
-> SYST:PRES DEF          // Presets the U8480 Series.
-> UNIT:POW DBM           // Sets the measurement unit to dBm.
-> CALC:LIM:STAT 1        // Enables the test limit feature.
-> CALC:LIM:LOW 4         // Sets the lower limit to 4 dBm.
-> CALC:LIM:UPP 10        // Sets the upper limit to 10 dBm.
```

The U8480 Series will start to monitor the RF power between 4 dBm (lower limit) and 10 dBm (upper limit). RF power that is either <4 dBm or >10 dBm will cause the U8480 Series to log an error.

NOTE "->" indicates the commands that you send to the U8480 Series.

Checking for limit failures

To check for limit failures, use the **CALCulate[1]:LIMit:FAIL?** and/or **CALCulate[1]:LIMit:FCOunt?** queries.

The **CALCulate[1]:LIMit:FAIL?** query will return **1** if one or more limit failures have occurred. If no limit failures have occurred, **0** will be returned.

The **CALCulate[1]:LIMit:FCOunt?** query will return the total number of limit failures.

An example of the programming sequence is shown as follows.

```
-> SYST:PRES DEF              // Presets the U8480 Series.
-> TRIG:SOUR EXT              // Sets the trigger source to external.
-> UNIT:POW DBM               // Sets the measurement unit to dBm.
-> CALC:LIM:STAT 1            // Enables the test limit feature.
-> CALC:LIM:LOW 4             // Sets the lower limit to 4 dBm.
-> CALC:LIM:UPP 10            // Sets the upper limit to 10 dBm.
-> CALC:LIM:CLE:AUTO OFF      // Disables auto-clearing of the fail counter.
-> CALC:LIM:CLE               // Clears the fail counter of any limit failure.
```

Provides 5 dBm of RF power to the U8480 Series, followed by sending an external trigger signal to the U8480 Series.

```
-> CALC:LIM:FAIL?             // Checks for limit failures.
<- 0                          // No limit failure, where the measured
                                 power is within the range of >4 dBm and
                                 <10 dBm.
-> CALC:LIM:FCO?              // Checks the total number of limit failures.
<- 0                          // No limit failure has been detected.
```

Provides 12 dBm of RF power to the U8480 Series, followed by sending an external trigger signal to the U8480 Series.

```
-> CALC:LIM:FAIL?             // Checks for limit failures.
<- 1                          // Limit failures have been detected, where
                                 the measured power is
                                 >10 dBm. Previous limit failures will not
                                 be cleared.
-> CALC:LIM:FCO?              // Checks the total number of limit failures.
<- 1                          // One limit failure has been detected.
```

Provides 8 dBm of RF power to the U8480 Series, followed by sending an external trigger signal to the U8480 Series.

```
-> CALC:LIM:FAIL?             // Checks for limit failures.
<- 1                          // No limit failure. "1" was caused by the
                                 previous limit failure.
-> CALC:LIM:FCO?              // Checks the total number of limit failures.
```

```
<- 1                                  // One limit failure has been detected.
```

Provides 2 dBm of RF power to the U8480 Series, followed by sending an external trigger signal to the U8480 Series.

```
-> CALC:LIM:FAIL?                     // Checks for limit failures.
<- 1                                  // Limit failures have been detected, where
                                         the measured power is
                                         <4 dBm. Previous limit failures will not be
                                         cleared.
-> CALC:LIM:FCO?                      // Checks the total number of limit failures.
<- 2                                  // Two limit failures have been detected.
```

NOTE "->" indicates the commands that you send to the U8480 Series.

"<-" indicates the response from the U8480 Series.

NOTE If `TRIGger[1]:DELay:AUTO` is set to `ON`, then the number of failures returned by `CALCulate[1]:LIMit:FCOunt?` is affected by the current filter settings.

Getting the Best Speed Performance

This section discusses the factors that influence the speed of operation (number of readings/sec) of the U8480 Series.

The following factors are those which have the greatest effect upon measurement speed (in no particular order):

- The selected measurement rate of either `NORMal`, `DOUBle`, or `FAST`.
- The trigger mode (for example, Free Run, Triggered Free Run, or Single Shot).
- The output format: `ASCii` or `REAL`.
- The units used for the measurement.
- The command used to take a measurement.

In addition, there are other influences in the `FAST` mode which are described in "Fast mode" on page 52.

The following paragraphs give a brief description of the above factors and how they are controlled using SCPI.

Measurement rate

There are three possible speed settings: `NORMal, DOUBle`, and `FAST`. These are set using the `[SENSe[1]:]MRATe` command.

In the `NORMal` and `DOUBle` modes, full instrument functionality is available, but in the `FAST` mode, limits are disabled.

Refer to the specifications in the *U8480 Series User's Guide* to determine the influence of these speed settings on the accuracy and noise performance of the U8480 Series.

Trigger mode

The U8480 Series has a very flexible triggering system. For simplicity, it can be described as having three modes:

- Free Run: When the U8480 Series is in the Free Run mode, it continuously takes measurements. A measurement is in free run when `INITiate:CONTinuous` is set to `ON` and `TRIGger:SOURce` is set to `IMMediate`.

- Triggered Free Run: When the U8480 Series is in the Triggered Free Run or Continuous Trigger mode, it takes a new measurement each time a trigger event is detected. A measurement is in triggered free run or continuous trigger when `INITiate:CONTinuous` is set to `ON` and `TRIGger:SOURce` is not set to `IMMediate`.
- Single Shot: When the U8480 Series is in the Single Shot mode, it takes a new measurement when a trigger event is detected and then returns to the idle state. A measurement is in single shot when `INITiate:CONTinuous` is set to `OFF`. Note that a measurement can take several EXT triggers depending on the filter settings. Refer to "TRIGger[1]:DELay:AUTO <boolean>" on page 311 for further information.

NOTE

A trigger event can be any of the following:

- The input signal meeting the trigger level criteria.
- Auto-level triggering being used.
- A `TRIGger[1][:IMMediate]` or `*TRG` command being sent.
- An external TTL level trigger being detected.

Trigger with delay

This can be achieved using the same sequences above (apart from the second) with `TRIG:DEL:AUTO` set to `ON`. Also, the `MEAS?` query operates in the trigger with delay mode.

In the trigger with delay mode, a measurement is not completed until the U8480 Series filter is full. In this way, the reading returned is guaranteed to be settled. In all other modes, the result returned is simply the current result from the filter and may or may not be settled. This depends on the current length of the filter and the number of readings that have been taken since a change in power level.

With trigger with delay enabled, the measurement speed can be calculated roughly using the following equation:

readings/sec = speed (as set by [SENSe[1]:]MRATe) / filter length

For example, with a filter length of 4 and `[SENSe[1]:]MRATe` set to `NORMal`, approximately 5 readings/sec is calculated by the U8480 Series.

Output format

The U8480 Series has two output formats for measurement results: **ASCii** and **REAL**. These formats are selected using the **FORMat** command. When **FORMat** is set to **REAL**, the returned result is in the IEEE-754 floating-point format (note that the byte order can be changed using **FORMat:BORDer**).

The **REAL** format is likely to be required only for the **FAST** mode as it reduces the amount of bus traffic.

Units

The U8480 Series can output results in either linear or log units. The internal units are linear; therefore optimal performance is achieved when the results output are also in linear units (since the overhead of performing a log function is removed).

Command used

In the Free Run mode, **FETCh?** must be used to return a result.

In other trigger modes, there are a number of queries that can be used, for example, **MEASure?**, **READ?**, **FETCh?**. Note that the **MEAS?** and **READ?** queries are compound commands – they perform a combination of other lower-level commands. Typically, the best speed performance is achieved using the low-level commands directly.

Trigger count

To get the fastest measurement speed, **TRIG:COUNt** must be set to return multiple measurements for each **FETCh?** query. For average only measurements, a count of 4 is required; however, 10 is recommended.

Fast mode

In the highest speed setting, the limiting factor tends to be the speed of the controller being used to retrieve results from the U8480 Series, and to a certain extent, the volume of remote traffic. The latter can be reduced using the **FORMat REAL** command to return results in the binary format. The former is a combination of two factors:

- the hardware platform being used
- the programming environment being used

How Measurements are Calculated

Figure 1-10 shows how measurements are calculated. It shows the order in which the various U8480 Series functions are implemented in the measurement calculation.

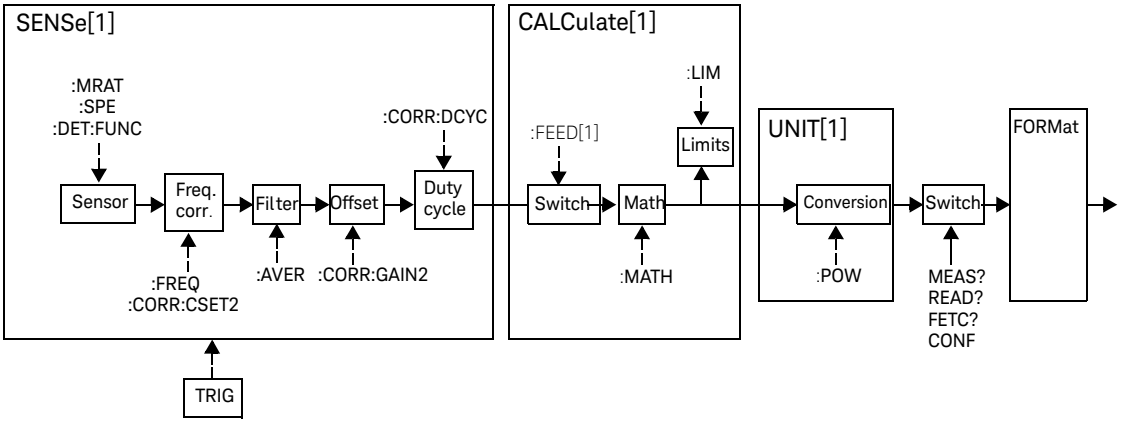


Figure 1-10 How measurements are calculated

Status Reporting

Status reporting is used to monitor the U8480 Series to determine when events have occurred. Status reporting is accomplished by configuring and reading status registers.

The U8480 Series has the following main registers:

- Status Register
- Standard Event Register
- Operation Status Register
- Questionable Status Register
- Device Status Register

There are other registers that exist “behind” the main registers, and they are described later in this chapter.

Status and Standard Event registers are read using the IEEE-488.2 common commands.

Operation and Questionable Status registers are read using the SCPI `STATus` command subsystem.

The general status register model

The generalized status register model shown in Figure 1-11 is the building block of the SCPI status system. This model consists of a condition register, a transition filter, an event register, and an enable register. A set of these registers is called a status group.

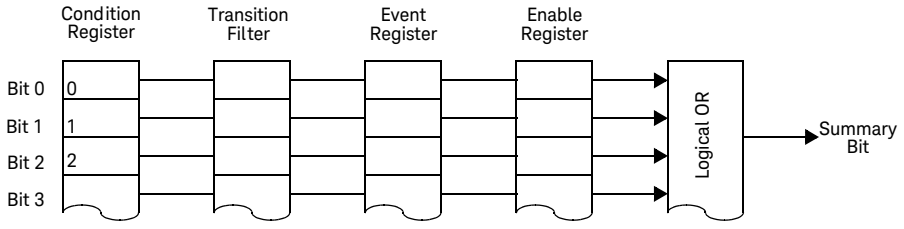


Figure 1-11 Generalized status register model

When a status group is implemented in an instrument, it always contains all of the component registers. However, there is not always a corresponding command to read or write to every register.

Condition register

The condition register continuously monitors the hardware and firmware status of the U8480 Series. There is no latching or buffering for this register; it is updated in real time. Condition registers are read-only.

Transition filter

The transition filter specifies which type of changes to the bit state in the condition register will set corresponding bits in the event register. Transition filter bits may be set for positive transitions (PTR), negative transitions (NTR), or both. Positive transition will cause the corresponding bit in the event register to be set when the condition bit changes from 0 to 1. Negative transition will cause the corresponding bit in the event register to be set when the condition bit changes from 1 to 0. Setting both positive and negative transitions will cause the corresponding bit in the event register to be set whenever the condition bit changes. Clearing both the positive and negative transition filters disables the corresponding bit in the event register to be set. Transition filters are read-write. They are unaffected by clear status (*CLS) or queries.

Event register

The event register latches transition events from the condition register as specified by the transition filter. Bits in the event register are latched, and once the bits are set, they will remain set until they are cleared by a query or clear status (*CLS). There is no buffering; therefore, while an event bit is set, subsequents events corresponding to that bit are ignored. Event registers are read-only.

Enable register

The enable register specifies which bits in the event register can generate a summary bit. The instrument logically ANDs corresponding bits in the event and enable registers, and ORs all the resulting bits to obtain a summary bit. Summary bits are, in turn, recorded in another register, usually the Status Byte. Enable registers are read-write. They are not affected by clear status (*CLS) or querying the enable registers. There is always a command to read and write to the enable register of a particular status group.

An example sequence

Figure 1-12 illustrates the response of a single bit position in a typical status group for various settings. The changing state of the condition in question is shown at the bottom of the figure. A small binary table shows the state of the chosen bit in each status register at selected times T1 to T5.

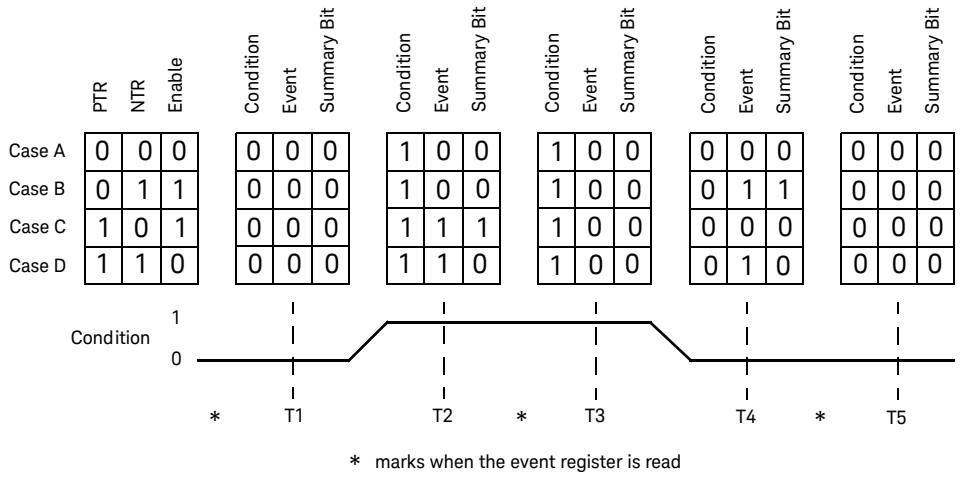


Figure 1-12 Typical status register bit changes

Consider **Case C**, where the positive transition filter is set to 1 and negative transition filter to 0. This configures the U8480 Series to set the corresponding bit in the event register whenever the condition bit changes from 0 to 1. The enable register is set to 1 to enable the summary bit to be generated each time there is a change in the event register.

At time **T1**, the condition bit is 0. Since there is no changes to the condition bit at this time, no corresponding bit in the event register will be set and the summary bit is 0.

At time **T2**, the condition bit changes from 0 to 1. Since the positive transition filter is set to detect condition bit changes from 0 to 1, the corresponding bit in the event register will be set to 1. The enable register is set to 1, which means that the summary bit will also be set to 1 whenever any bit in the event register is set to 1.

At time **T3**, the condition bit remains 1. The event register is cleared by a query. Hence, the event register bit and summary bit are set to 0.

At time **T4**, the condition bit changes from 1 to 0. Since the positive transition filter is set to detect condition bit changes from 0 to 1, the corresponding bit in the event register will be set to 0, signifying no event has been logged. The summary bit is set to 0 as no bit is set in the event register.

At time **T5**, the condition bit remains 0. Since there is no changes to the condition bit at this time, no corresponding bit in the event register will be set and the summary bit is 0.

Consider **Case D**, where the positive transition filter is set to 1 and negative transition filter to 1. This configures the U8480 Series to set the corresponding bit in the event register whenever there are changes to the condition bit. The enable register is set to 0 to disable the summary bit to be generated.

At time **T1**, the condition bit is 0. Since there is no changes to the condition bit at this time, no corresponding bit in the event register will be set and the summary bit is 0.

At time **T2,** the condition bit changes from 0 to 1. Since the positive and negative transition filters are set to detect any changes to the condition bit, the corresponding bit in the event register will be set to 1. The enable register is set to 0, which means that the summary bit will not be set.

At time **T3**, the condition bit remains 1. The event register is cleared by a query. Hence, the event register bit and summary bit are set to 0.

At time **T4**, the condition bit changes from 1 to 0. Since the positive and negative transition filters are set to detect any changes to the condition bit, the corresponding bit in the event register will be set to 1, signifying an event has been logged. The summary bit is 0 as the enable register is set to 0.

At time **T5**, the condition bit remains 0. The event register is cleared by a query. Hence, the event register bit and summary bit are set to 0.

How to read registers

The condition polling method is used to access the information in the status register groups. In this method, the U8480 Series has a passive role. It only informs the PC that conditions have changed when the PC “asks”. When you monitor a condition with the polling method, you must:

1. Determine which register contains the bit that monitors the condition.
2. Send the unique query that reads that register.
3. Examine the bit to see if the condition has changed.

The polling method works well if you do not need to know about the changes the moment they occur. Detecting an immediate change in a condition using the polling method requires your program to continuously read the registers at very short intervals. This is not particularly efficient, and there is a possibility that an event may be missed.

Status registers

The Status System in the U8480 Series is shown in Figure 1-13. The Operation Status and Questionable Status groups are 16 bits wide, while the Status Byte and Standard Event groups are 8 bits wide. In all 16-bit groups, the most significant bit (bit 15) is not used and is always set to 0.

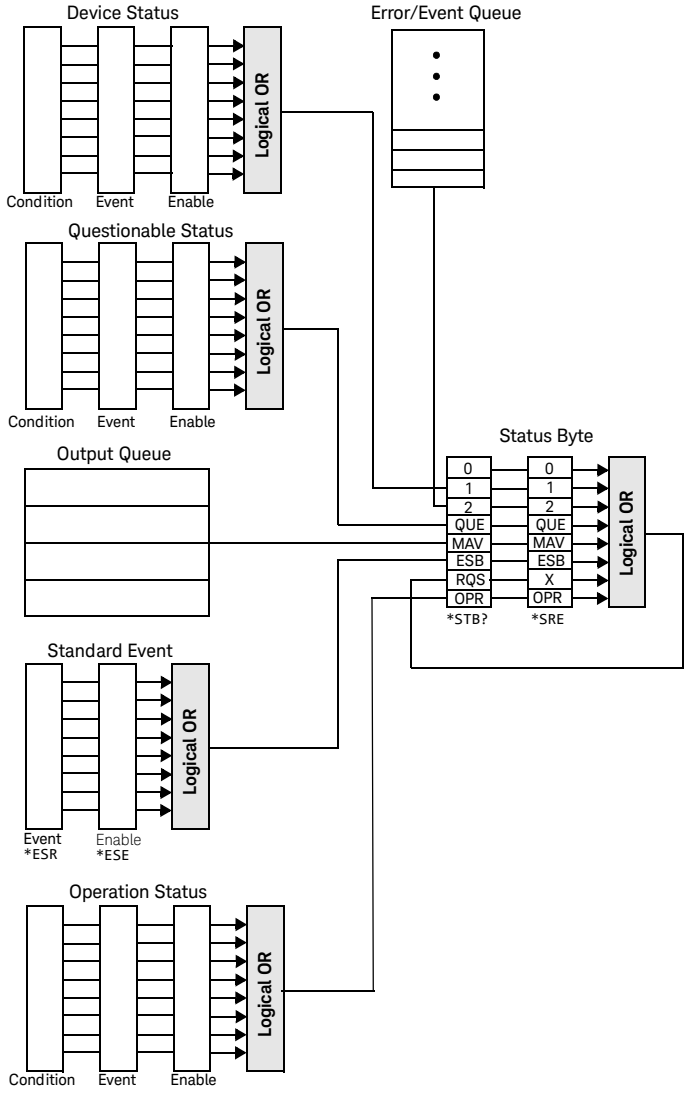


Figure 1-13 Status system

Status byte summary register

The status byte summary register reports conditions from other status registers. Query data waiting in the U8480 Series output buffer is immediately reported through the “message available” bit (bit 4). Clearing an event register clears the corresponding bits in the status byte summary register. Reading all messages in the output buffer, including any pending queries, clears the message available bit.

Table 1-4 Bit definitions - Status byte register

Bit number	Decimal weight	Definition
0	1	Not Used (Always set to 0)
1	2	Device Status Register summary bit One or more bits are set in the Device Status Register (bits must be “enabled” in the enable register)
2	4	Error/Event Queue
3	8	Questionable Status Register summary bit One or more bits are set in the Questionable Status Register (bits must be “enabled” in the enable register)
4	16	Data Available Data is available in the U8480 Series output buffer
5	32	Standard Event One or more bits are set in the Standard Event register (bits must be “enabled” in the enable register)
6	64	Request Service The U8480 Series is requesting service (serial poll)
7	128	Operation Status Register summary bit One or more bits are set in the Operation Status Register (bits must be “enabled” in the enable register)

Particular bits in the status byte register are cleared when:

- The standard event, questionable status, operation status, and device status are queried.
- The error or event queue becomes empty.
- The output queue becomes empty.

The status byte enable register (**SRE**, service request enable) is cleared when you:

- cycle the U8480 Series power.
- execute a ***SRE 0** command.

Using *STB? to read the status byte

The ***STB?** (status byte query) is similar to a serial poll except it is processed like any other U8480 Series command. ***STB?** returns the same result as an IEEE-488 serial poll except that the request service bit (bit 6) *is not* cleared if a serial poll has occurred. ***STB?** is not handled automatically by the IEEE-488 bus interface hardware, and the query is executed only after previous commands have completed. Using ***STB?** does not clear the status byte summary register.

Standard event register

The standard event register reports the following types of instrument events: power-on detected, command and syntax errors, command execution errors, self-test or calibration errors, query errors, or when an overlapped command completes following an ***OPC** command. Any or all of these conditions can be reported in the standard event summary bit through the enable register. You must write a decimal value using the ***ESE** (event status enable) command to set the enable register mask.

Table 1-5 Bit definitions - Standard event register

Bit number	Decimal value	Definition
0	1	Operation Complete All overlapped commands following an ***OPC** command have been completed
1	2	Not Used (always set to 0)
2	4	Query Error A query error occurred, refer to error numbers 410 to 440 in Error message list
3	8	Device-Dependent Error A device error occurred, refer to error numbers 310 to 350 in Error message list

Bit number	Decimal value	Definition
4	16	Execution Error An execution error occurred, refer to error numbers 211 to 231 in Error message list
5	32	Command Error A command syntax error occurred, refer to error numbers 101 to 178 in Error message list
6	64	User Request
7	128	Power On Power has been turned off and on since the last time the event register was read or cleared

The standard event register is cleared when you:

- send a `*CLS` (clear status) command.
- query the event register using `*ESR?` (event status register).

The standard event enable register is cleared when you:

- cycle the U8480 Series power.
- execute an `*ESE 0` command.

Questionable status register

The questionable status register provides information about the quality of the U8480 Series measurement results. Any or all of these conditions can be reported in the questionable data summary bit through the enable register. You must write a value using the `STATus:QUEStionable:ENABle` command to set the enable register mask.

The following bits in these registers are used by the U8480 Series.

Table 1-6 Bit definitions - Questionable status registers

Bit number	Decimal weight	Definition
0 to 2	-	Not Used
3	8	POWer Summary
4 to 7	-	Not Used
8	256	CALibration Summary
9	512	Power-On Self-Test
10 to 14	-	Not Used
15	-	Not Used (always 0)

The condition bits are set and cleared under the following conditions:

Table 1-7 Bit change conditions for Questionable status register

Bit number	Definition	EVENts causing bit changes
3	POWer Summary	This is a summary bit for the Questionable POWer Register – **SET:** Error –230, “Data corrupt or stale” Error –231, “Data questionable;Input Overload” Error –231, “Data questionable;ZERO ERROR” – **CLEARED:** When no errors are detected by the U8480 Series during a measurement covering the causes given for it to set

Bit number	Definition	EVENts causing bit changes

8	CALibration Summary	This is a summary bit for the Questionable CALibration Register
		– **SET:**
		These may be caused by
		`CALibration[1]:ZERO:AUTO ONCE or`
		`CALibration[1]:AUTO ONCE or`
		`CALibration[1][:ALL] or`
		`CALibration[1][:ALL]?`
		Error –231, "Data questionable;ZERO ERROR"
		Error –231, "Data questionable;CAL ERROR"
		– **CLEARED:**
		When any of the commands listed above succeed and no errors are placed on the error queue

9	Power-On Self-Test	– **SET:** This bit is set when the power-on self-test fails – **CLEARED:** When the power-on self-test passes

Operation status register

The Operation Status group monitors conditions in the U8480 Series measurement process.

The following bits in these registers are used by the U8480 Series:

Table 1-8 Bit definitions - Operation status

Bit number	Decimal weight	Definition
0	1	CALibrating Summary
1 to 3	-	Not Used
4	16	MEASuring Summary
5	32	Waiting for TRIGger Summary
6 to 9	-	Not Used
10	1024	SENSe Summary
11	2048	Lower Limit Fail Summary
12	4096	Upper Limit Fail Summary

Bit number	Decimal weight	Definition
13 to 14	-	Not Used
15	-	Not Used (always 0)

The condition bits are set and cleared under the following conditions:

Table 1-9 Bit change conditions for operation status

Bit number	Definition	EVENts causing bit changes
0	CALibrating	This is a summary bit for the Operation CALibrating Register – **SET:** At the beginning of zeroing (`CALibration[1]:ZERO:AUTO ONCE`) or calibration (`CALibration[1]:AUTO ONCE`). Also for the compound command/query `CALibration[1][:ALL]?`, this bit is set when calibration begins. – **CLEARED:** At the end of zeroing or calibration
4	MEASuring	This is a summary bit for the Operation MEASuring Register – **SET:** When the U8480 Series is taking a measurement – **CLEARED:** When the measurement is completed
5	Waiting for TRIGger	This is a summary bit for the Operation TRIGger Register – **SET:** When the U8480 Series enters the “wait-for-trigger” state – **CLEARED:** When the U8480 Series enters the “idle” state
10	SENSe	This is a summary bit for the Operation SENSe Register – **SET:** When the U8480 Series is reading data from the non-volatile memory – **CLEARED:** When the U8480 Series is not reading data from the non-volatile memory

Bit number	Definition	EVENTs causing bit changes
11	Lower Limit Fail	This is a summary bit for the Lower Limit Fail Register – **SET:** If a measurement is made and the lower limit test fails – **CLEARED:** If a measurement is made and the lower limit test is not enabled or the test is enabled and passes
12	Upper Limit Fail	This is a summary bit for the Upper Limit Fail Register – **SET:** If a measurement is made and the upper limit test fails – **CLEARED:** If a measurement is made and the upper limit test is not enabled or the test is enabled and passes

Device status register

The device status register set contains bits which give device-dependent information.

The following bits in these registers are used by the U8480 Series:

Table 1-10 Bit definitions - Device status register

Bit number	Decimal weight	Definition
0 to 2	-	Not Used
3	8	U8480 Series Error
4 to 14	-	Not Used
15	-	Not Used (always 0)

The condition bits are set and cleared under the following conditions:

Table 1-11 Bit change conditions for Device status

Bit number	Definition	EVENts causing bit changes
3	U8480 Series Error	– **SET:** If the U8480 Series non-volatile memory has failed or other hardware has failed – **CLEARED:** In every other condition

Using the Operation Complete commands

***OPC?** and ***OPC** allow you to maintain synchronization between the PC and the U8480 Series. ***OPC?** places a 1 into the U8480 Series output queue when all pending U8480 Series commands have completed. If your program reads this response before continuing program execution, you can ensure synchronization between one or more sensors and the PC.

The ***OPC** command sets bit 0 (Operation Complete) in the Standard Event Status Register when all pending U8480 Series operations have completed.

Procedure

1. Send a Device Clear message to clear the U8480 Series output buffer.
2. Clear the event registers with the ***CLS** (clear status) command.
3. Enable operation complete using the ***ESE 1** command (standard event register).
4. Send ***OPC?** (operation complete query) to assure synchronization.
5. Send your programming command string, and place the ***OPC** (operation complete) command as the last command.
6. Send ***STB?** (status byte query) to poll the register. This command does not clear the status byte summary register.

Examples

This example program uses ***OPC?** to determine when the U8480 Series has finished calibrating.

```
CAL:AUTO ONCE
*OPC?
MEAS?
```

Saving and Recalling U8480 Series Configurations

To reduce repeated programming, up to ten U8480 Series configurations can be stored in the U8480 Series non-volatile memory. The error list, remote addresses, calibration table data, and zeroing/calibration information are not stored.

How to save and recall a configuration

The U8480 Series configurations are saved and recalled with the following commands:

```
*SAV <NRf>
*RCL <NRf>
```

The range of values for `<NRf>` in the above commands is 1 to 10.

Using Device Clear to Halt Measurements

Device clear is an IEEE-488 low-level bus message which can be used to halt measurements in progress. The status registers, the error queue, and all configuration states are left unchanged when a device clear message is received. Device clear performs the following actions:

- All measurements in progress are aborted.
- The U8480 Series returns to the trigger “idle state”.
- The U8480 Series input and output buffers are cleared.
- The U8480 Series is prepared to accept a new command string.

2 MEASurement Commands

This chapter explains how to use the **MEASure** group of instructions to acquire data using a set of high-level instructions.

Measurement Commands

Measurement commands are high-level commands used to acquire data. They enable you to trade interchangeability against fine control of the measurement process.

Measurement command	Description
MEASure?	Provides the simplest way to program a U8480 Series for measurements. **MEASure?** is a compound command which is equivalent to a **CONFigure** followed by a **READ?**. It does not enable much flexibility or control over measurement settings.
CONFigure	Used to change the U8480 Series configuration values. **CONFigure** must then be followed by another command which takes the measurement, for example, **INITiate?** followed by **FETCh?**
READ?	Takes a measurement using parameters previously set up using either **CONFigure** or lower-level commands. **READ?** is equivalent to an **INITiate** (which performs the data acquisition) and a **FETCh?**
FETCh?	Retrieves measurements taken by **INITiate** [a].

[a] **INITiate** is described in Chapter 12, "TRIGger Subsystem" on page 297.

CONFigure, **FETCh?**, **READ?**, and **MEASure?** all have a numeric suffix which refers to a specific measurement window. The U8480 Series does not have the measurement window feature, so this suffix is always 1. Figure 2-1 shows an example of the configuration returned measurement result.

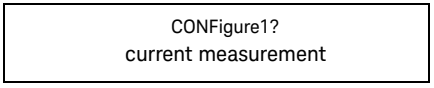


Figure 2-1 Measurement display CALCulate block channel

Optional parameters

`CONFigure`, `FETCh?`, `READ?`, and `MEASure?` have the following three optional parameters:

- An expected power value
- A resolution
- A source list

Refer to "Auto-Averaging Settings" on page 374 to configure the correct parameters for expected power and resolution.

Expected power value

The `<expected_value>` parameter sets the expected power level of the measurement.

Resolution

The `<resolution>` parameter sets the resolution of the measurement. This parameter does not affect the resolution of the returned data, but it does affect the auto-averaging setting.

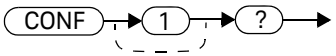
Source list

The `<source list>` parameter is used to define the measurement channel.

CONFigure[1]?

This query returns the present configuration of the measurement.

Syntax



The returned string depends on the setting of the **CALCulate:MATH** commands.

The configuration is returned as a quoted string in the following format:

<function> <expected_value>,<resolution>,<source list>”

<expected_value> returns the expected value sent by the last **CONFigure** command or +20 dBm by default.

Example

CONF? *Queries the measurement configuration.*

Reset condition

On reset:

- The command function is set to :**POWer:AC**.
- The expected power level is set to +20 dBm.
- The resolution is set to 3.
- The source list on the U8480 Series is set to (@1).

CONFigure[1] Command

The **CONFigure** command is used to set:

- the expected measurement power level.
- the measurement resolution.

The **CONFigure** command does not make the power measurement after setting the configuration. Use **READ?**, or alternatively use **INITiate** followed by a **FETCh?**, to make the measurement.

The **CONFigure** command also applies the following defaults to the measurement(s) which are specified in the <source list> parameter:

Default settings	Description
INITiate[1]:CONTinuous OFF	Sets the U8480 Series to make one trigger cycle when **INITiate** is sent
TRIGger[1]:SOURce IMMediate	When **TRIG:SOUR** is set to **BUS** or **HOLD**, sets the U8480 Series to make the measurement immediately once a trigger is received
TRIGger[1]:DELay:AUTO ON	Enables automatic delay before making the measurement
[SENSe[1]:]AVERage:COUNt:AUTO ON	Enables automatic filter length selection
[SENSe[1]:]AVERage:STATe ON	Enables averaging

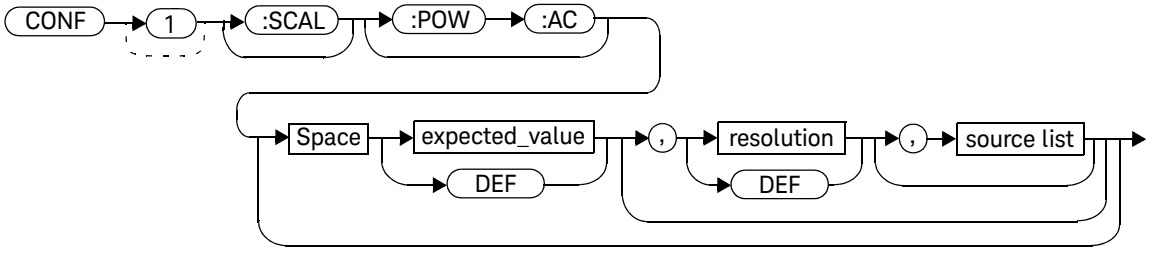
CONFigure[1][:SCALar][:POWer:AC] [<expected_value>[,<resolution>[,<source list>]]]

This command is used to set:

- the expected measurement power level.
- the measurement resolution.

Refer to "Auto-Averaging Settings" on page 374 to configure the correct parameters for the expected power and resolution.

Syntax



Parameters

Refer to "Optional parameters" on page 73 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value	A numeric value for the expected power level. The units of measurement are dBm and W. The default units are defined by `UNIT:POWer`	sensor-dependent **DEF**[a]
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4[b] 1.0, 0.1, 0.01, 0.001 **DEF**[a]
source list	The measurement channel which the command is implemented on. The U8480 Series supports only one channel. Therefore, the only valid value is (@1).	(@1)

[a] The mnemonic **DEF** means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying **DEF** leaves the parameter value unchanged.

[b] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represents 1, 0.1, 0.01, and 0.001 respectively.

Example

```
CONF DEF,2,(@1)
```

This command configures the measurement to measure power using the current range and a resolution setting of 2.

FETCh[1]? Query

The **FETCh?** query calculates the measurement and sends the result to the PC. The result format is set by **FORMat[:READing][:DATA]**. Refer to Chapter 5, "FORMat Subsystem," on page 125 for further information.

The query returns a measurement result when it is valid. The measurement result is invalid under the following conditions:

- when ***RST** is executed.
- whenever a measurement is initiated.
- when any **SENSe** parameter, such as frequency, is changed.

If the data is invalid, the **FETCh?** query is not completed until all data becomes valid. The exceptions to this are, if the U8480 Series is in the idle state and the data is invalid, or the U8480 Series has been reconfigured as defined above and no new measurement has been initiated. In such cases, the **FETCh?** routine generates the error –230, "Data corrupt or stale" and no result is returned. A common cause for this error is receiving a **FETCh?** after a ***RST**. If the expected value and resolution parameters are not the same as those that were used to collect the data, error –221, "Settings conflict" occurs.

NOTE

When TRIGger[1]:SOURce is EXT and a new acquisition has been initiated (using the INITiate command for example), FETCh? waits until the trigger takes place before executing. If trigger conditions are not satisfied – when the trigger level differs greatly from the signal level for example – this can give the impression that the U8480 Series has hung.

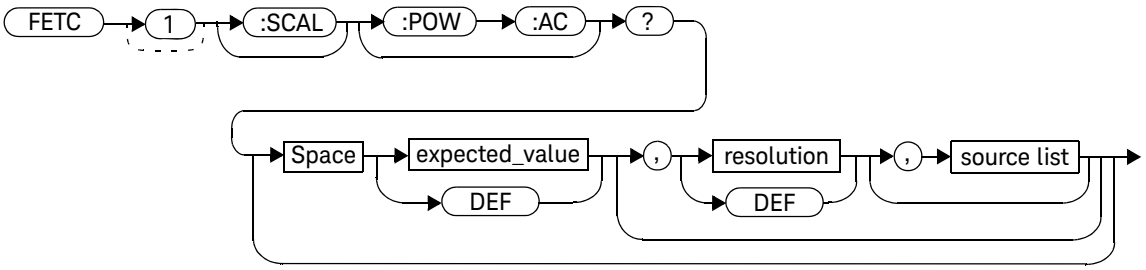
To unlock the U8480 Series and adjust trigger settings, a Device Clear command must be performed.

FETCh[1][:SCALar][:POWer:AC]? [<expected_value>[,<resolution>[,<source list>]]]

This command sets the measurement function, recalculates the measurement, and places the result on the bus. The result is a power-based measurement and is expressed in the units defined by **UNIT[1]:POWer**.

Refer to "Auto-Averaging Settings" on page 374 to configure the correct parameters for the expected power and resolution.

Syntax



Parameters

Refer to "Optional parameters" on page 73 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value	The expected power level parameter can be set to **DEF** or a numeric value. If a value is entered, it should correspond to that set by **CONFigure** otherwise an error occurs. The units of measurement are dBm and W. The default units are defined by **UNIT:POWer**.	sensor-dependent **DEF**[a]

Item	Description/Default	Range of values
resolution	A numeric value for the resolution. If it is unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4[b] 1.0, 0.1, 0.01, 0.001 **DEF**[a]
source list	The measurement channel which the command is implemented on. The U8480 Series supports only one channel. Therefore, the only valid value is (@1).	(@1)

[a] The mnemonic **DEF** means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying **DEF** leaves the parameter value unchanged.

[b] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represents 1, 0.1, 0.01, and 0.001 respectively.

Example

```
FETC?
```
Queries the measurement result.

Error messages

- If the last measurement is not valid, error –230, “Data corrupt or stale” occurs. A measurement is valid after it has been initiated. It becomes invalid when either a reset occurs or any measurement parameter, for example frequency, is changed.
- If the expected_value and resolution parameters are not the same as the current expected value and resolution settings on the measurement, error –221, “Settings conflict” occurs.

FETCh[1][:SCALar][:POWer:AC]:MUNC? [<expected_value>[,<resolution>[,<source list>]]]

This command sets the measurement function, recalculates the measurement and the corresponding measurement uncertainty, and places the result on the bus. The result is a power-based measurement and is expressed in the units defined by `UNIT[1]:POWer`.

When "`UNIT:POW W`" is set, the parameters returned will be:

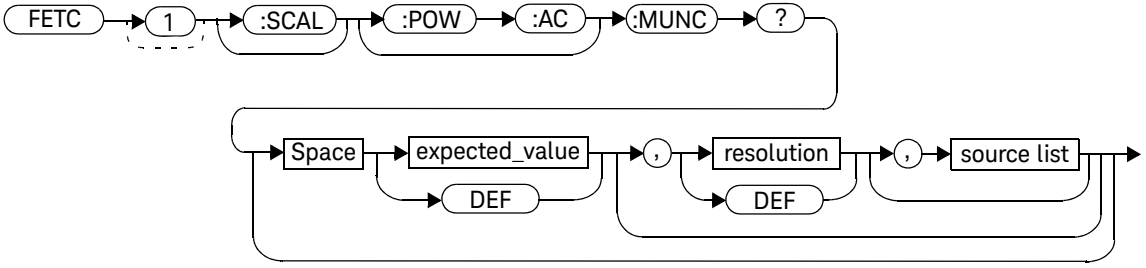
- measured power in Watts
- +measurement uncertainty value in %
- –measurement uncertainty value in %

When "`UNIT:POW DBM`" is set, the parameters returned will be:

- measured power in dBm
- +measurement uncertainty value in dB
- –measurement uncertainty value in dB.

NOTE This query is only supported for `TRIG:COUNT` of 1. It is not supported for power and frequency sweep modes.

Syntax



Parameters

Refer to "Optional parameters" on page 73 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value	The expected power level parameter can be set to **DEF** or a numeric value. If a value is entered, it should correspond to that set by **CONFigure** otherwise an error occurs. The units of measurement are dBm and W. The default units are defined by **UNIT:POWer**.	sensor-dependent **DEF**[a]
resolution	A numeric value for the resolution. If it is unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4[b] 1.0, 0.1, 0.01, 0.001 **DEF**[a]
source list	The measurement channel which the command is implemented on. The U8480 Series supports only one channel. Therefore, the only valid value is (@1).	(@1)

[a] The mnemonic **DEF** means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying **DEF** leaves the parameter value unchanged.

[b] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represents 1, 0.1, 0.01, and 0.001 respectively.

Error messages

– If this query is sent, and **TRIG:COUNT** > 1, error –221, "Settings conflict" occurs.

READ[1] Query

The **READ?** query is most commonly used with the **CONFigure** command to cause a new power measurement to be taken and the result returned to the output buffer. The result format is set by **FORMat[:READing][:DATA]**. Refer to Chapter 5, "FORMat Subsystem" on page 125 for further information.

The **READ?** query is equivalent to:

```
INITiate
FETCh?
```

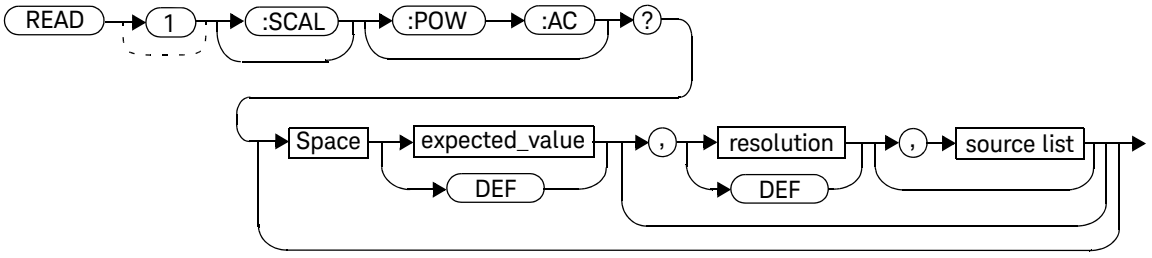
READ[1][:SCALar][:POWer:AC]? [<expected_value>[,<resolution>[,<source list>]]]

This query sets the measurement function, aborts then initiates the measurement, calculates the measurement result, and places the result on the bus. The result is a power-based measurement and is expressed in the units defined by `UNIT[1]:POWer`.

Refer to "Auto-Averaging Settings" on page 374 to configure the correct parameters for expected power and resolution.

NOTE `INITiate[1]:CONTinuous` must be set to `OFF`, otherwise error –213, "`INIT ignored`" occurs. If `TRIGger[1]:SOURce` is set to `BUS`, error –214, "Trigger deadlock" occurs.

Syntax



Parameters

Refer to "Optional parameters" on page 73 for additional details on the parameters in this query.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The expected power level parameter can be set to **DEF** or a numeric value. If a value is entered, it should correspond to that set by **CONFigure** otherwise an error occurs.	sensor-dependent **DEF**[a]
resolution	A numeric value for the resolution. If it is unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4[b] 1.0, 0.1, 0.01, 0.001 **DEF**[a]
source list	The measurement channel which the command is implemented on. The U8480 Series supports only one channel. Therefore, the only valid value is (@1).	(@1)

[a] The mnemonic **DEF** means DEFault. This is not equivalent to the DEFault parameter used in the command sub-systems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying **DEF** leaves the parameter value unchanged.

[b] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represents 1, 0.1, 0.01, and 0.001 respectively.

Example

READ? *Queries the measurement.*

Error messages

- **INITiate[1]:CONTinuous** must be set to **OFF**, otherwise error –213, "Init ignored" occurs.
- If **TRIGger[1]:SOURce** is set to **BUS** or **HOLD,** error –214, "Trigger deadlock" occurs.
- If the expected value and resolution parameters are not the same as the current expected value and resolution settings on the measurement, error –221, "Settings conflict" occurs.

READ[1][:SCALar][:POWer:AC]:MUNC? [<expected_value>[,<resolution>[,<source list>]]]

This command sets the measurement function, aborts then initiates the measurement and the corresponding measurement uncertainty, and places the result on the bus. The result is a power-based measurement and is expressed in the units defined by `UNIT[1]:POWer`.

When "`UNIT:POW W`" is set, the parameters returned will be:

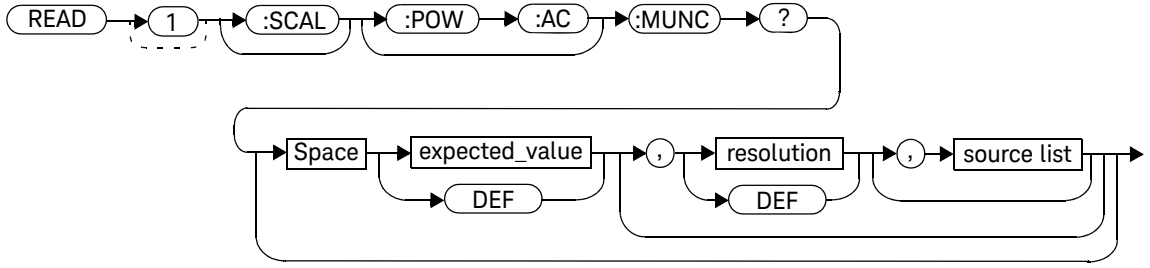
- measured power in Watts
- +measurement uncertainty value in %
- –measurement uncertainty value in %

When "`UNIT:POW DBM`" is set, the parameters returned will be:

- measured power in dBm
- +measurement uncertainty value in dB
- –measurement uncertainty value in dB.

NOTE This query is only supported for `TRIG:COUNT` of 1. It is not supported for power and frequency sweep modes.

Syntax



Parameters

Refer to "Optional parameters" on page 73 for additional details on the parameters in this query.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The expected power level parameter can be set to **DEF** or a numeric value. If a value is entered, it should correspond to that set by **CONFigure** otherwise an error occurs.	sensor-dependent **DEF**[a]
resolution	A numeric value for the resolution. If it is unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4[b] 1.0, 0.1, 0.01, 0.001 **DEF**[a]
source list	The measurement channel which the command is implemented on. The U8480 Series supports only one channel. Therefore, the only valid value is (@1).	(@1)

[a] The mnemonic **DEF** means DEFault. This is not equivalent to the DEFault parameter used in the command sub-systems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying **DEF** leaves the parameter value unchanged.

[b] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represents 1, 0.1, 0.01, and 0.001 respectively.

Error messages

- **INITiate[1]:CONTinuous** must be set to **OFF**, otherwise error –213, "Init ignored" occurs.
- If **TRIGger[1]:SOURce** is set to **BUS** or **HOLD,** error –214, "Trigger deadlock" occurs.
- If the expected value and resolution parameters are not the same as the current expected value and resolution settings on the measurement, error –221, "Settings conflict" occurs.
- If this query is sent, and **TRIG:COUNT** > 1, error –221, "Settings conflict" occurs.

MEASure[1] Query

The **MEASure?** query configures the U8480 Series to perform a power measurement with the given measurement function, range, and resolution, and then make the measurement. The format of the result is set by **FORMat[:READing][:DATA]**. Refer to Chapter 5, "FORMat Subsystem" on page 125 for further information.

The **MEASure?** compound command is equivalent to:

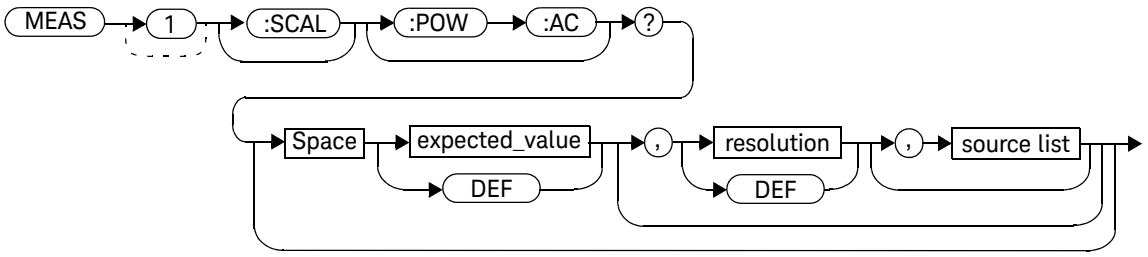
```
CONFigure
READ?
```

MEASure[1][:SCALar][:POWer:AC]? [<expected_value>[,<resolution>[,<source list>]]]

This query aborts any measurement in progress, configures the U8480 Series, calculates the measurement result, and places the result on the bus.

Refer to "Auto-Averaging Settings" on page 374 to configure the correct parameters for the expected power and resolution.

Syntax



Parameters

Refer to "Optional parameters" on page 73 for additional details on the parameters in this query.

Item	Description/Default	Range of values
expected_value (for the expected power level)	A numeric value for the expected power level. The units of measurement are dBm and W. The default units are defined by **UNIT:POWer**.	sensor-dependent **DEF**[a]
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4[b] 1.0, 0.1, 0.01, 0.001 **DEF**[a]
source list	The measurement channel which the command is implemented on. The U8480 Series supports only one channel. Therefore, the only valid value is (@1).	(@1)

[a] The mnemonic **DEF** means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying **DEF** leaves the parameter value unchanged.

[b] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represents 1, 0.1, 0.01, and 0.001 respectively.

Example

```
MEAS? -10DBM,1,(@1)
```

Queries the measurement using an expected power level of –10 dBm and a resolution setting of 1.

MEASure[1][:SCALar][:POWer:AC]:MUNC? [<expected_value>[,<resolution>[,<source list>]]]

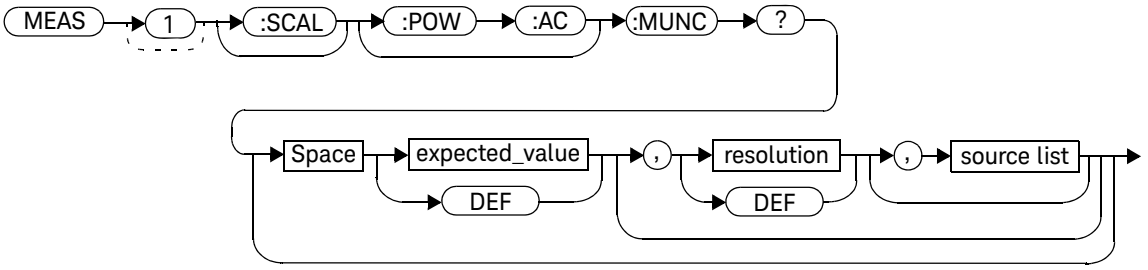
This command aborts any measurement in progress, configures the U8480 Series, calculates the measurement result and the corresponding measurement uncertainty, and places the result on the bus. The result is a power-based measurement and is expressed in the units defined by `UNIT[1]:POWer`.

When "`UNIT:POW W`" is set, the parameters returned will be:

- measured power in Watts
- +measurement uncertainty value in %
- –measurement uncertainty value in %
- When "`UNIT:POW DBM`" is set, the parameters returned will be:
- measured power in dBm
- +measurement uncertainty value in dB
- –measurement uncertainty value in dB.

NOTE This query is only supported for `TRIG:COUNT` of 1. It is not supported for power and frequency sweep modes.

Syntax



Parameters

Refer to "Optional parameters" on page 73 for additional details on the parameters in this query.

Item	Description/Default	Range of values
expected_value (for the expected power level)	A numeric value for the expected power level. The units of measurement are dBm and W. The default units are defined by **UNIT:POWer**.	sensor-dependent **DEF**[a]
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4[b] 1.0, 0.1, 0.01, 0.001 **DEF**[a]
source list	The measurement channel which the command is implemented on. The U8480 Series supports only one channel. Therefore, the only valid value is (@1).	(@1)

[a] The mnemonic **DEF** means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying **DEF** leaves the parameter value unchanged.

[b] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represents 1, 0.1, 0.01, and 0.001 respectively.

Error messages

– If this query is sent, and **TRIG:COUNT** > 1, error –221, "Settings conflict" occurs.

3 CALCulate Subsystem

This chapter explains how the **`CALCulate`** command subsystem is used to perform post-acquisition data processing.

CALCulate Command Subsystem

The **CALCulate** command subsystem performs post-acquisition data processing. Functions in the **SENSe** command subsystem are related to data acquisition, while the **CALCulate** command subsystem operates on the data acquired by a **SENSe** function.

There is an independent **CALCulate** block in the U8480 Series, as shown below.

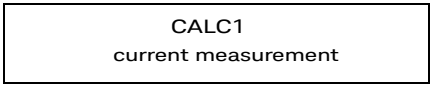


Figure 3-1 Measurement display CALCulate block channel

Figure 3-2 details where the commands are applied within the **CALCulate** block.

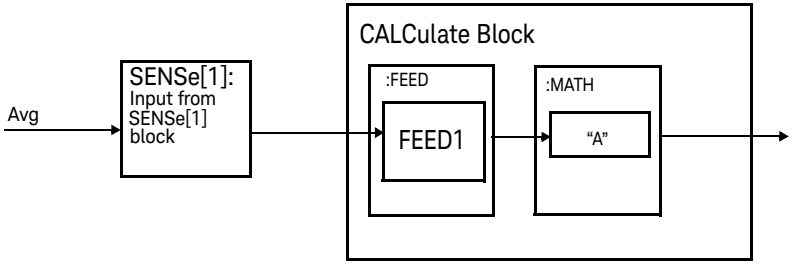


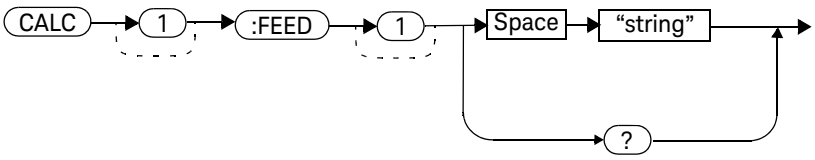
Figure 3-2 CALCulate block

CALCulate[1]:FEED[1] <“string”>

This command sets the input measurement mode to be fed to the specified input on the **CALC** block. It is applied to the measurement after the **CALCulate[1]:MATH[:EXPRession]** command has been used to specify which measurement the feed is taken from.

Under certain circumstances, the measurement mode is changed by the **CALCulate[1]:MATH[:EXPRession]** command. Refer to “CALCulate[1]:MATH[:EXPRession] <“string”>” on page 111 for further information.

Syntax



Parameters

Item	Description	Range of values
string	The input measurement type to be fed to the specific input on the CALC block is **AVER** (average).	“POW:AVER”

Example

CALC:FEED “POW:AVER”

*This command selects the input for FEED of the CALC block to be average power. The measurement from which the feed is taken is determined by **CALC:MATH:EXPR**.*

Reset condition

On reset, the feed is set to **POW:AVER**.

Query

CALCulate[1]:FEED[1]?

The query returns the current value of the string.

Query example

CALC:FEED?	*Queries the current setting of the CALC block on FEED.*

Error message

If the command parameter is not **"POW:AVER"**, error –224, "Illegal parameter value" occurs.

CALCulate[1]:LIMit Commands

These commands set the measurement limits which enable you to:

- set upper-level and lower-level limits
- query if there has been a failure
- count the number of failures
- clear the counter

The following commands or queries are detailed in this section:

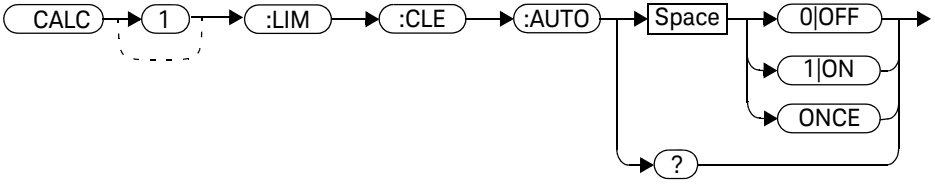
```
CALCulate[1]:LIMit:CLEar:AUTo <boolean>|ONCE
CALCulate[1]:LIMit:CLEar[IMMediate]
CALCulate[1]:LIMit:FAIL?
CALCulate[1]:LIMit:FCOunt?
CALCulate[1]:LIMit:LOWer[:DATA] <numeric_value>
CALCulate[1]:LIMit:UPPer[:DATA] <numeric_value>
CALCulate[1]:LIMit:STATe <boolean>
```

CALCulate[1]:LIMit:CLEar:AUTO <boolean>|ONCE

This command controls when the FCO (fail counter) is cleared of any limit failure. The FCO is used to determine the results returned by the `CALCulate[1]:LIMit:FAIL?` query.

- If `ON` is specified, the FCO is set to 0 each time a measurement is:
  - initiated using `INITiate[1][:IMMediate]`
  - initiated using `INITiate[1]:CONTinuous ON`
  - measured using `MEASure?`
  - read using `READ?`
- If `OFF` is specified, the FCO is not cleared by the above commands or queries.
- If `ONCE` is specified, the FCO is cleared only after the first initialization, and then starts accumulating any limit failures.

Syntax



Example

`CALC:LIM:CLE:AUTO 1` *This command switches on automatic clearing of the FCO.*

Reset condition

On reset, **CALCulate[1]:LIMit:CLEar:AUTO** is set to **ON**.

Query

CALCulate[1]:LIMit:CLEar:AUTO?

The query enters a 1 or 0 into the output buffer indicating whether limit failures are cleared automatically when a new measurement is initiated.

- 1 is entered into the output buffer when limit failures are cleared automatically when a new measurement is initiated.
- 0 is entered into the output buffer when limit failures are not cleared automatically when a new measurement is initiated.

In the case where limit failures are cleared once, when a query occurs, 1 is entered into the output buffer if no measurement is initiated. If a measurement is initiated, then 0 is entered.

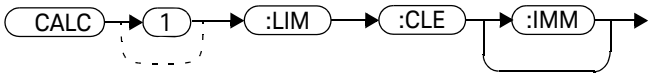
Query example

CALC:LIM:CLE:AUTO? *Queries when the fail counter is cleared.*

CALCulate[1]:LIMit:CLEar[:IMMediate]

This command immediately clears the FCO (fail counter) of any limit failure. The FCO is used to determine the results returned by the **CALCulate[1]:LIMit:FAIL?** query.

Syntax



Example

```
CALC:LIM:CLE:IMM
```

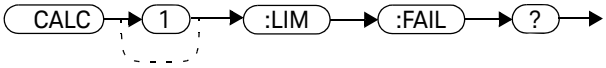
This command clears the FCO.

CALCulate[1]:LIMit:FAIL?

This query enters a 1 or 0 into the output buffer indicating whether there have been any limit failures. A limit failure is defined as **CALCulate[1]:LIMit:FCOunt?** being non-zero. The FCO (fail counter) can be zeroed using the **CALCulate[1]:LIMit:CLEar** command.

- 1 is returned when one or more limit failures have occurred
- 0 is returned when no limit failures have occurred

Syntax



Example

CALC:LIM:FAIL? *Queries if there have been any limit failures.*

Reset condition

On reset, the buffer is set to zero.

CALCulate[1]:LIMit:FCOunt?

This query returns the total number of limit failures.

If the appropriate **STATe** commands are set to **ON**, each time a measurement is initiated and the result is outside the limits, the counter is incremented by one.

If the measured value is equal to a limit, this is a limit pass.

The counter is reset to zero by any of the following commands:

- `*RST`
- `CALCulate[1]:LIMit:CLEar[:IMMediate]`
- `CALCulate[1]:LIMit:CLEar:AUTO ON`

When **CALCulate[1]:LIMit:CLEar:AUTO** is set to **ON**, the counter is set to zero *each* time a measurement is:

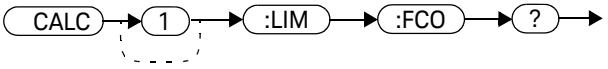
- measured using **MEASure?**
- read using **READ?**
- initiated using:
  - **INITiate[1][:IMMediate]**, or
  - **INITiate[1]:CONTinuous ON**

When **CALCulate[1]:LIMit:CLEar:AUTO** is set to **ONCE**, the counter is set to zero the *first* time a measurement is:

- measured using **MEASure?**
- read using **READ?**
- initiated using:
  - **INITiate[1][:IMMediate]**, or
  - **INITiate[1]:CONTinuous ON**

The maximum number of errors is 65535. If more than 65535 errors are detected, the counter returns to zero.

Syntax



Example

`CALC:LIM:FCO?` *Queries the number of limit failures.*

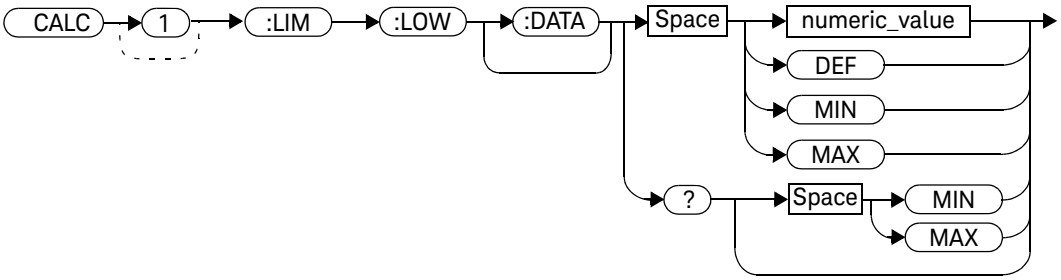
Reset condition

On reset, the counter is set to zero.

CALCulate[1]:LIMit:LOWer[:DATA] <numeric_value>

This command enters a value for the lower test limit for the measurement used in the **CALCulate[1]:LIMit:FAIL?** test. The units used are dependent on the current setting of **UNIT:POWer**. When the measured value is less than the value specified in **CALCulate[1]:LIMit:LOWer[:DATA]**, **CALCulate[1]:LIMit:FAIL?** reports a fail. When the measured value is greater than or equal to the limit, a fail is not reported.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value for the lower test limit: – **DEF**: The default is –90.00 dBm – **MIN**: –150 dBm – **MAX**: +230 dBm	–150 to +230 dBm **DEF** **MIN** **MAX**

Example

`CALC:LIM:LOW 0.1`	*This command enters a lower limit for the measurement depending on the unit as follows:* *dBm = 0.1 dBm* *W = 100 mW*

Reset condition

On reset, the lower limit is set to –90.00 dBm (**DEF**).

Query

`CALCulate[1]:LIMit:LOWer[:DATA]? [MIN|MAX]`

The query returns the current setting of the lower limit or the values associated with **MIN** and **MAX**.

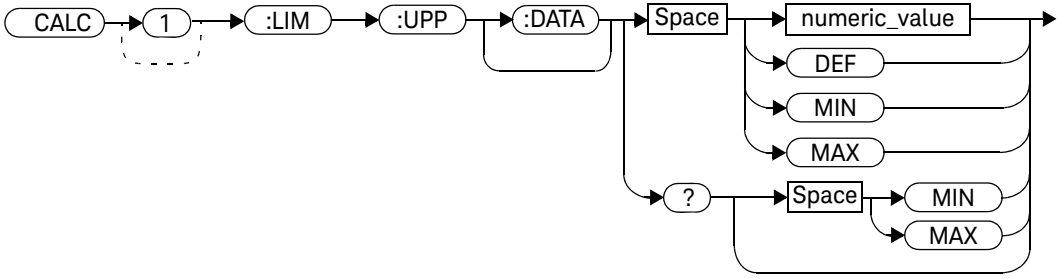
Query example

`CALC:LIM:LOW?`	*Queries the lower limit set.*

CALCulate[1]:LIMit:UPPer[:DATA] <numeric_value>

This command enters a value for the upper test limit for the measurement used in the **CALCulate[1]:LIMit:FAIL?** test. The units used are dependent on the current setting of **UNIT:POWer**. When the measured power is greater than the value specified in **CALCulate[1]:LIMit:UPPer[:DATA]**, **CALCulate[1]:LIMit:FAIL?** reports a fail. When the measured level is less than or equal to the limit, a fail is not reported.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value for the upper test limit: – **DEF**: The default is +90.00 dBm – **MIN**: –150 dBm – **MAX**: +230 dBm	–150 to +230 dBm **DEF** **MIN** **MAX**

Example

`CALC:LIM:UPP 5`	*This command enters an upper limit for the measurement depending on the unit as follows:* *dBm = 5 dBm* *W = 5 W*

Reset condition

On reset, the measurement limit is set to +90.00 dBm (**DEF**).

Query

`CALCulate[1]:LIMit:UPPer[:DATA]? [MIN|MAX]`

The query returns the current setting of the upper limit or the values associated with **MIN** and **MAX**.

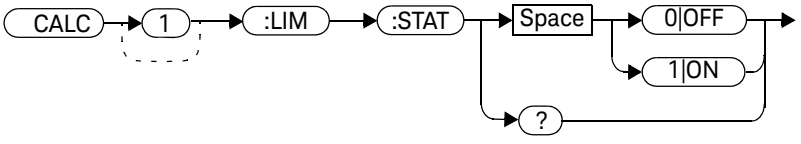
Query example

`CALC:LIM:UPP?`	*Queries the setting of the upper limit.*

CALCulate[1]:LIMit:STATe <boolean>

This command enables/disables the test limits.

Syntax



Example

```
CALC:LIM:STAT 1
```

This command enables the limit checking function.

Reset condition

On reset, limit checking is disabled.

Query

```
CALCulate[1]:LIMit:STATe?
```

The query enters 1 or 0 into the output buffer indicating the status of the limits testing feature.

- 1 is returned when limit testing is enabled
- 0 is returned when limit testing is disabled

Query example

`CALC:LIM:STAT?`	*Queries whether the limit checking function is turned on or off.*

Error message

If **CALCulate[1]:LIMit:STATe** is set to **ON** while **[SENSe[1]:]MRATe** is set to **FAST** or **[SENSe[1]:]SPEed** is set to **400**, error –221, “Settings conflict” occurs.

CALCulate[1]:MATH Commands

These commands define and carry out the following mathematical transformations on **SENSe** data for a single measurement.

The following command and query are detailed in this section:

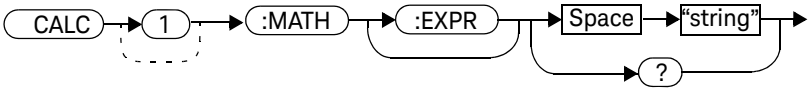
CALCulate[1]:MATH[:EXPRession] <“string”>

CALCulate[1]:MATH[:EXPRession]:CATalog?

CALCulate[1]:MATH[:EXPRession] <“string”>

This command sets to a single measurement.

Syntax



Parameters

Item	Description/Default	Range of values
string	A single string value detailing the measurement type: The default is **SENS1**.	**“(SENS1)”**[a], [b]

[a] Quotes are mandatory. Either single or double quotes may be used.

[b] Must be uppercased.

Example

```
CALC:MATH “(SENS1)”
```

This command sets to a single measurement.

Reset condition

On reset, the measurements are set to **“**(**SENS1**)**”**.

Query

```
CALCulate[1]:MATH[:EXPRession]?
```

The query returns the current math measurement setting.

Query example

```
CALC:MATH?
```

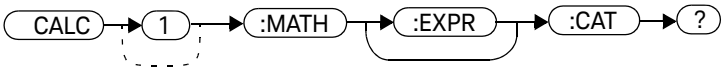
Queries the current setting of the math expression.

CALCulate[1]:MATH[:EXPRession]:CATalog?

This query lists all the defined expressions. The response is a list of comma-separated strings. Each string contains an expression.

For the single measurement, the string is **“(SENS1)”**.

Syntax



Example

`CALC:MATH:CAT?`	*This query lists all the defined math expressions.*

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

4 CALibration Subsystem

This chapter explains how the **`CALibration`** command subsystem is used to zero and calibrate the U8480 Series.

CALibration Command Subsystem

The **CALibration** command subsystem is used to zero and calibrate the U8480 Series.

The numeric suffix of the CALibration command (CALibration1) refers to Channel A.

- Zeroing and calibration of the U8480-Series is recommended:
- When a 5°C change in temperature occurs
- When connection to the U8480-Series is established
- Every 24 hours
- Prior to measuring low-level signals.

The following **CALibration** commands are overlapped commands:

- `CAL:ALL`
- `CAL:AUTO`
- `CAL:ZERO:AUTO`

An overlapped command allows the U8480 Series to continue parsing and executing subsequent commands while it is still executing.

CALibration[1][:ALL]

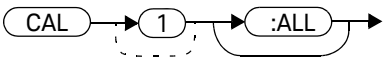
NOTE **This command is identical to `CALibration[1][:ALL]?`; however, unlike the query, it does not provide a response to indicate whether the calibration has been successful or not.**

This command causes the U8480 Series to perform a calibration sequence. For internal calibration (**`CALibration:TYPE INTernal`**), the calibration sequence consists of:

1. Zeroing the U8480 Series (**`CALibration:ZERO:AUTO ONCE`**) and
2. Calibrating the U8480 Series (**`CALibration:AUTO ONCE`**).

This command is not applicable for external calibration as the U8480 Series does not have control of the power reference.

Syntax



Example

CAL *This command causes the U8480 Series to perform a calibration sequence.*

Error messages

- If this command is sent and the U8480 Series is in the external calibration mode (**`CALibration:TYPE EXTernal`**), the error –224, “Illegal parameter value” occurs.
- If calibration was not carried out successfully, the error –231, “Data Questionable; CAL ERROR” occurs.
- If zeroing was not carried out successfully, the error –231, “Data Questionable; ZERO ERROR” occurs..

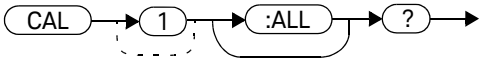
CALibration[1][:ALL]?

NOTE **This query is identical to `CALibration[1][:ALL]`; however, unlike the command, it provides a response to indicate whether the calibration has been successful or not.**

When the calibration sequence has completed, 0 or 1 is entered into the output buffer to indicate if the sequence was successful. If the result is:

- 0, the calibration has passed
- 1, the calibration has failed

Syntax



Query example

`CAL?` *Causes the U8480 Series to perform a calibration sequence, and returns a result.*

Error messages

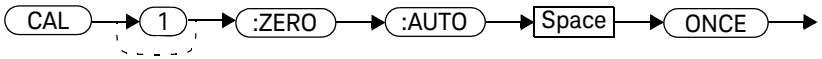
- If calibration was not carried out successfully, the error –231, “Data Questionable; CAL ERROR” occurs.
- If zeroing was not carried out successfully, the error –231, “Data Questionable; ZERO ERROR” occurs.

CALibration[1]:ZERO:AUTO ONCE

This command performs zeroing on the U8480 Series.

Zeroing will be performed only once assuming that no power is supplied to the U8480 Series.

Syntax



Example

```
CAL:ZERO:AUTO ONCE
```

This command causes the U8480 Series to perform a zeroing routine.

Error message

If zeroing was not carried out successfully, error –231, “Data Questionable; ZERO ERROR” occurs.

CALibration[1]:AUTO [ONCE|ON|OFF|0|1]

This command calibrates the U8480 Series when enabled.

The response of this command is based on the setting for **CALibration:TYPE**. If the calibration type is set to **EXTernal** (**CALibration:TYPE EXTernal**), the command assumes that the U8480 Series is connected to a 1 mW 50 MHz reference signal and performs the calibration. **1|ON** and **0|OFF** are not supported, and if received will execute an error message.

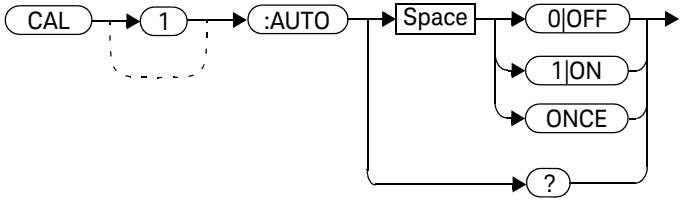
If the calibration type is set to **INTernal (CALibration:TYPE INTernal)**, the U8480 Series will perform an internal calibration.

When **1|ON** is enabled for the internal calibration, the calibration is updated if the U8480 Series temperature changes by ±10 ^{o}C from the last internal calibration.

The **0|OFF** parameter disables the automatic internal calibration process. Use the **CALibration:AUTO ONCE** command to manually perform the calibration.

NOTE **The U8480 Series should be zeroed before calibration using the CALibration:ZERO:AUTO ONCE command.**

Syntax



Example

`CAL:AUTO ONCE`	*This command causes the U8480 Series to perform calibration.*

Reset condition

On reset, automatic calibration is enabled.

Error messages

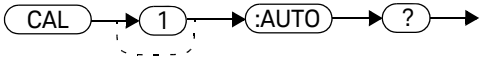
- If this command is set to `1|ON` and `CALibration:TYPE EXTernal` is selected, the error –224, “Illegal parameter value” occurs.
- If calibration was not carried out successfully, the error –231, “Data Questionable; CAL ERROR” occurs.

CALibration[1]:AUTO?

This query returns a result which indicates whether auto-calibration is enabled or disabled. If the result is:

- 0, auto-calibration is disabled
- 1, auto-calibration is enabled

Syntax



Query example

CAL : AUTO? *Queries the auto-calibration state.*

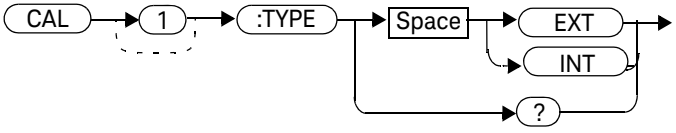
CALibration[1]:TYPE EXTernal|INTernal

This command sets the U8480 Series to the external or internal calibration mode. External calibration requires a 50 MHz 1 mW power reference, while internal calibration utilizes the internal reference circuit to perform calibration and does not require the 50 MHz 1 mW power reference.

Upon power up, the U8480 Series defaults to the internal calibration mode.

NOTE **Setting `CALibration[1]:TYPE` to `EXTernal` will automatically set `CALibration[1]:AUTO` to `OFF`.**

Syntax



Example

`CAL:TYPE EXT` *This command sets the external calibration mode.*

Reset condition

On reset, the calibration mode is set to internal.

Query

`CALibration[1]:TYPE?`

This query returns the current calibration mode of either “INT” or “EXT”.

Query example

`CAL:TYPE?` *Queries the calibration mode for the U8480 Series.*

Error message

This command is only able to set the calibration mode to “EXT” or “INT”. Error –224, “Illegal parameter value” occurs for any other value.

5 FORMat Subsystem

This chapter explains how the **FORMat** command subsystem is used to set a data format for transferring numeric information.

FORMat Command Subsystem

The `FORMat` command subsystem sets a data format for transferring numeric information. This data format is used only for response data by commands that are affected by the `FORMat` command subsystem.

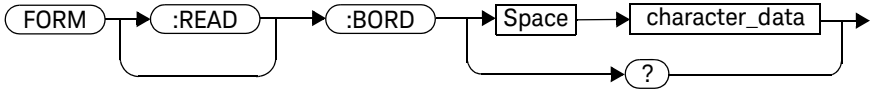
The queries affected are:

- `FETCh?`
- `READ?`
- `MEASure?`

FORMat[:READings]:BORDer <character_data>

This command controls whether the binary data is transferred in normal or swapped Byte ORDer. It is only used when **FORMat[:READings][:DATA]** is set to **REAL**.

Syntax



Parameters

Item	Description/Default	Range of values
character_data	Byte order of binary data transfer: – NORMal – SWAPped	NORMal SWAPped

Example

```
FORM:BORD SWAP
```

This command sets the byte order to swapped.

Reset condition

On reset, this value is set to **NORMal**.

Query

FORMat[:READings]:BORDer?

The query returns the current setting of the byte order. The format of the response is NORMal or SWAPped.

Query example

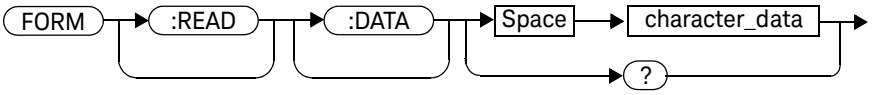
FORM:BORD? *Queries the current byte order setting.*

FORMat[:READings][:DATA] <character_data>

This command sets the data format for transferring numeric information to either **ASCii** or **REAL**:

- When the format type is **ASCii**, numeric data is output as ASCII bytes in the **<NR3>** format.
- When the format type is **REAL**, numeric data is output as IEEE 754 64-bit floating point numbers in a definite length block. The result is an 8-byte block per number. Each complete block is terminated by a line feed character.

Syntax



Parameters

Item	Description/Default	Range of values
character_data	Data format for transferring data: – ASCii – REAL	ASCii REAL

Example

```
FORM REAL
```

*This command sets the format to **REAL**.*

Reset condition

On reset, the format is set to **ASCii**.

Query

FORMat[:READings][:DATA]?

The query returns the current setting of format: **ASCii** or **REAL**.

Query example

FORM? *Queries the current format setting.*

6 MEMory Subsystem

This chapter explains how the **MEMory** command subsystem is used to configure U8480 Series frequency-dependent offset tables, Gamma tables, the S-Parameter table, and save/recall registers.

MEMory Command Subsystem

The **MEMory** command subsystem is used to:

- edit and review frequency-dependent offset tables.
- store frequency-dependent offset tables.
- edit and review save/recall registers.
- edit and review Gamma tables.
- store Gamma tables.
- edit and review an S-Parameter table.
- store an S-Parameter table.

Stored tables remain in the U8480 Series memory during power down. The U8480 Series is capable of storing 10 frequency-dependent offset tables of 80 frequency points each, 3 Gamma Tables of 1024 frequency points each, and an S-Parameter table of 1024 points.

MEMory:CATalog Queries

This group is used to query information on the current contents of the U8480 Series:

- Frequency-dependent offset tables
- Save/recall registers
- Gamma tables
- The S-Parameter table

The following queries are detailed in this section:

`MEMory:CATalog[:ALL]?`

`MEMory:CATalog:STATe?`

`MEMory:CATalog:TABLe?`

MEMory:CATalog[:ALL]?

This query lists stored frequency-dependent offset tables, Gamma tables, the S-Parameter table, and save/recall registers.

There are 10 frequency-dependent offset tables named **CUSTOM_A** through CUSTOM_J which do not contain any data when the U8480 Series is shipped from the factory.

There are also three Gamma tables named Gamma1 through Gamma3, and one S-Parameter table named SParam1, which do not contain any data when the U8480 Series is shipped from the factory.

The U8480 Series returns the data in the form of two numeric parameters and as many strings as there are stored tables and save/recall registers:

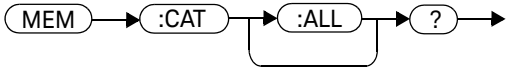
<numeric_value>,<numeric_value>{,<string>}

- The first numeric parameter indicates the amount of memory, in bytes, used for the storage of tables and registers.
- The second numeric parameter indicates the memory, in bytes, available for the storage of tables and registers.
- Each string parameter returned indicates the name, type, and size of a stored table or save/recall register:
  - <string>,<type>,<size>
  - <string> indicates the name of the table or save/recall register.
  - <type> indicates TABL for frequency-dependent offset tables, Gamma tables, or the S-Parameter table, or STAT for a save/recall register.
  - <size> indicates the size of the table or save/recall register in bytes.

A sample of a response may look like the following:

```
1178,26230,"DEFAULT,TABL,14","TABLE1,TABL,116",
"TABLE2,TABL,74",...........,"State0,STAT,1619",
"State1,STAT,1619","State2,STAT,1619" ..........
```

Example 1: Syntax



Example

```
MEM:CAT?
```

Queries the list of tables and save/recall registers.

MEMory:CATalog:STATe?

This query is used to list the save/recall registers.

The U8480 Series returns the data in the form of two numeric parameters and as many strings as there are save/recall registers.

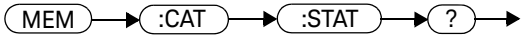
`<numeric_value>,<numeric_value>{,<string>}`

- The first numeric parameter indicates the amount of memory, in bytes, used for the storage of registers.
- The second parameter indicates the memory, in bytes, available for the storage of registers.
- Each string parameter returned indicates the name, type, and size of a save/recall register:
  - `<string>,<type>,<size>`
  - `<string>` indicates the name of the save/recall register.
  - `<type>` indicates `STAT` for a save/recall register.
  - `<size>` indicates the size of the save/recall register in bytes.

For example, a sample of a response may look like:

`0,16190,"State0,STAT,0","State1,STAT,0" .....`

Syntax



Example

`MEM:CAT:STAT?` *Queries the list of save/recall registers.*

MEMory:CATalog:TABLe?

This query is used to list the stored frequency-dependent offset tables, Gamma tables, and the S-Parameter table.

There are 10 frequency-dependent offset tables named `CUSTOM_A` through `CUSTOM_J` which do not contain any data when the U8480 Series is shipped from the factory.

There are also three Gamma tables named `Gamma1` through `Gamma3`, and one S-Parameter table named `SParam1`, which do not contain any data when the U8480 Series is shipped from the factory.

The U8480 Series returns the data in the form of two numeric parameters and as many strings as there are stored tables.

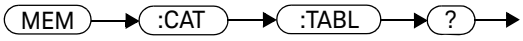
`<numeric_value>,<numeric_value>{,<string>}`

- The first numeric parameter indicates the amount of memory, in bytes, used for the storage of tables.
- The second parameter indicates the memory, in bytes, available for the storage of tables.
- Each string parameter returned indicates the name, type, and size of a stored table:
  - `<string>,<type>,<size>`
  - `<string>` indicates the name of the table.
  - `<type>` indicates `TABL` for a table.
  - `<size>` indicates the size of the table in bytes.

For example, a sample of a response may look like:

```
1178,10040,”DEFAULT,TABL,14”,”TABLE1,TABL,116”,
“TABLE2,TABL,74”,”TABLE3,TABL,62”..........
```

Syntax



Example

```
MEM:CAT:TABL?
```

Queries the list of stored tables.

MEMory:CLEar Commands

These commands are used to remove the frequency-dependent offset tables, Gamma tables, the S-Parameter table, and save/recall registers. This command subsystem removes the data contents but does not affect the name of the associated table or save/recall register.

The following commands are detailed in this section:

`MEMory:CLEar[:NAME] <“character_data”>`

`MEMory:CLEar:TABLe`

NOTE The contents cleared using these commands are non-recoverable.

MEMory:CLEar[:NAME] <“character_data”>

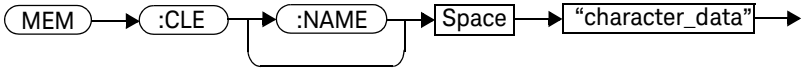
This command clears the contents of a specified frequency-dependent offset table, Gamma table, S-Parameter table, or save/recall register.

Although the table remains, a `MEMory:TABLe:FREQuency|GAIN:POINts|MEMory:TABLe:SGAMma:POIN|MEM:TABLe:SPAR:POIN?` query returns a 0 as there are no contents in the table.

For frequency-dependent offset tables, Gamma tables, and the S-Parameter table, this command is an alternative form of the `MEMory:CLEar:TABLe` command. The only difference is the method in which the table is selected.

NOTE The contents cleared using this command are non-recoverable.

Syntax



Parameter

Item	Description/Default	Range of values
character_data	Contains an existing table name or save/recall register.	Any existing table name or save/recall register.

Example

```
MEM:CLE “TABLE5”
```

This command clears the contents of frequency-dependent offset table, TABLE5.

Error message

If the table or save/recall register name does not exist, error –224, “Illegal parameter value” occurs.

MEMory:CLEar:TABLe

This command is used to clear the contents of the table currently selected using `MEMory:TABLe:SELect`. Although the table remains, a `MEMory:TABLe:FREQuency|GAIN:POINts|MEMory:TABLe:SGAMma:POIN|MEM:TABLe:SPAR:POIN?` query returns a 0 as the table contents are empty.

This command is an alternative form of the `MEMory:CLEar[:NAME]` command. The difference is the method in which the table is selected.

NOTE The contents cleared using this command are non-recoverable.

Syntax

Example

`MEM:CLE:TABL`	*This command clears the contents of the currently selected table.*

Error message

If no table is selected, error –221, “Settings conflict” occurs.

MEMory:FREE Queries

These queries are used to return information on the amount of free memory space available for frequency-dependent offset tables, Gamma tables, the S-Parameter table, and save/recall registers.

The following queries are described in this section:

`MEMory:FREE[:ALL]?`

`MEMory:FREE:STATe?`

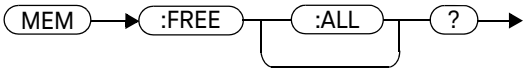
`MEMory:FREE:TABLe?`

MEMory:FREE[:ALL]?

This query returns the amount of memory free for frequency-dependent offset tables, Gamma tables, the S-Parameter table, and save/recall registers. The format of the response is:

`<bytes_available>,<bytes_in_use>`

Syntax



Example

`MEM:FREE?`

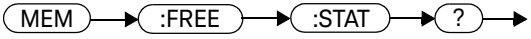
Queries the amount of memory free for frequency-dependent offset tables, Gamma tables, the S-Parameter table, and save/recall registers.

MEMory:FREE:STATe?

This query returns the amount of memory free for save/recall registers. The format of the response is:

```
<bytes_available>,<bytes_in_use>
```

Syntax



Example

`MEM:FREE:STAT?`	*Queries the amount of free memory for save/ recall registers.*

MEMory:FREE:TABLe?

This query returns the amount of memory free for frequency-dependent offset tables, Gamma tables, and the S-Parameter table. The format of the response is:

```
<bytes_available>,<bytes_in_use>
```

Syntax

Example

```
MEM:FREE:TABL?
```

Queries the amount of free memory for frequency-dependent offset tables, Gamma tables, and the S-Parameter table.

MEMory:NSTates?

This query returns the number of registers available for save/recall. As there are 10 registers, this query always returns a 10.

Syntax

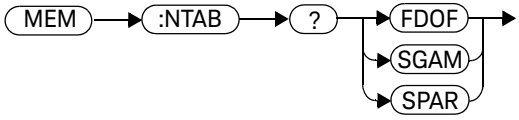
Example

`MEM:NST?`	*Queries the number of registers available for save/recall.*

MEMory:NTABles? FDOFset|SGAMma|SPARam

This query returns the number of tables for the frequency-dependent offset, Gamma, or S-Parameter correction.

Syntax



Example

`MEM:NTAB? FDOF|SGAM|SPAR`	*Queries the number of tables for frequency-dependent offset, Gamma, or S-Parameter correction.*

MEMory:STATe Commands

These commands are used to query and define register names.

The following command and query are described in this section:

`MEMory:STATe:CATalog?`

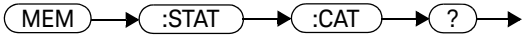
`MEMory:STATe:DEFine <“character_data”>,<numeric_value>`

MEMory:STATe:CATalog?

This query returns a list of the save/recall register names in the ascending order of register number. The format of the response is:

`<string>,<string>,.....,<string>`

Syntax



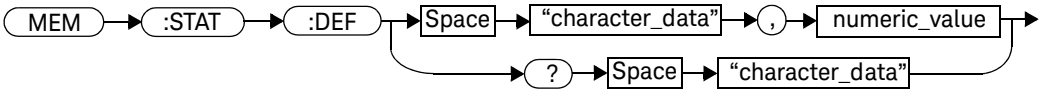
Example

`MEM:STAT:CAT?` *Queries the save/recall register names.*

MEMory:STATe:DEFine <“character_data”>,<numeric_value>

This command is used to associate a name with a save/recall register number.

Syntax



Parameters

Item	Description/Default	Range of values
character_data	Details the register name. A maximum of 12 characters can be used.	A to Z (uppercase) a to z (lowercase) 0 to 9_ (underscore)
numeric_value	A numeric value (<NRf>) for the register number.	0 to 9

Example

```
MEM:STAT:DEF “SETUP1”,4
```
*This command names register 4 SET*UP1.

Query

```
MEMory:STATe:DEFine? <“character_data”>
```

The query returns the register number for the given register name.

Query example

```
MEM:STAT:DEF? “SETUP1”
```
Queries the register number of SETUP1.

Error messages

- If the register number is out of range, error –222, “Data out of range” occurs.
- If the name is invalid, error –224, “Illegal parameter value” occurs.
- If a register with the same name already exists, error –257, “File name error” occurs (command only).

MEMory:TABLe Commands

These commands are used to define a frequency-dependent offset table, a Gamma table, or an S-Parameter table, and to write to and read data from it.

The following commands and queries are described in this section:

```
MEMory:TABLe:FREQuency <numeric_value>{,<numeric_value>}
MEMory:TABLe:FREQuency:POINts?
MEMory:TABLe:GAIN[:MAGNitude] <numeric_value>{,<numeric_value>}
MEMory:TABLe:GAIN[:MAGNitude]:POINts?
MEMory:TABLe:MOVE <“character_data”>,<“character_data”>
MEMory:TABLe:SELect <“character_data”>
MEMory:TABLe:SGAMma <numeric_value>,<numeric_value>
{,<numeric_value>}{,<numeric_value>}
MEMory:TABLe:SGAMma:POINts?
MEMory:TABLe:SPARam
<S11|S12|S21|S22>,<numeric_value>,<numeric_value>
{,<numeric_value>}{,<numeric_value>}
MEMory:TABLe:SPARam:POINts? <S11|S12|S21|S22>
```

MEMory:TABLe:FREQuency <numeric_value>{,<numeric_value>}

This command is used to enter frequency data into the currently selected table. Any previous frequency list is cleared before the new frequency list is stored. The frequencies must be entered in the ascending order. Entries in the frequency lists correspond with:

- entries in the offset factor lists, as shown in Table 6-1.
- entries in the magnitude and phase lists, as shown in Table 6-2.
- entries under S11, S12, S21, and S22, as shown in Table 6-3.

Table 6-1 Frequency and offset factor list

Frequency	Offset
Frequency 1	Offset 1
"	"
Frequency 80	Offset 80

Table 6-2 Gamma frequency, magnitude, and phase list

Frequency	Magnitude	Phase
Frequency 1	Magnitude 1	Phase 1
"	"	"
Frequency 1024	Magnitude 1024	Phase 1024

Table 6-3 S-Parameter frequency, magnitude, and phase list

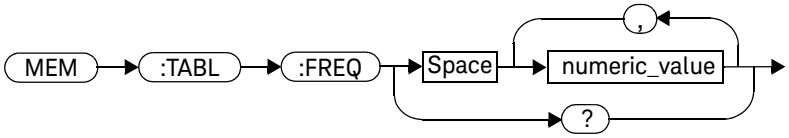
Frequency	S11	S12	S21	S22
Frequency 1	Magnitude 1	Magnitude 1	Magnitude 1	Magnitude 1
	Phase 1	Phase 1	Phase 1	Phase 1
"	"	"	"	"
Frequency 1024	Magnitude 1024	Magnitude 1024	Magnitude 1024	Magnitude 1024
	Phase 1024	Phase 1024	Phase 1024	Phase 1024

Ensure that the frequency points you use cover the frequency range of the signals that you want to measure. If you measure a signal with a frequency outside the frequency range defined in the table, then the U8480 Series uses the highest or lowest point in the table to calculate the offset.

Depending on the available memory, the U8480 Series is capable of storing the following:

- up to 10 frequency-dependent offset tables, each containing 80 points.
- up to 3 Gamma tables, each containing up to 1024 points.
- one S-Parameter table, containing up to 1024 points.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value for the frequency. The default unit is Hz.	0 Hz to 1000 GHz [a],[b]

[a] The following measurement units can be used:

Hz

kHz (10^3)

MHz (10^6)

GHz (10^9)

[b] All frequencies are truncated to a multiple of 1 kHz.

Example

`MEM:TABL:FREQ 200MHz,600MHz` *This command enters frequencies of 200 MHz and 600 MHz into the currently selected table.*

Query

`MEMory:TABLe:FREQuency?`

The query returns a list of frequency points for the table currently selected. The frequencies are returned in Hz.

Query example

`MEM:TABL:FREQ?` *Queries the frequency points in the currently selected table.*

Error messages

- If more than 80 frequencies for a frequency-dependant offset table, or more than 1024 frequencies a Gamma or S-Parameter table are in the list, error –108, “Parameter not allowed” occurs.
- If the frequencies are not entered in the ascending order, error –220, “Parameter error; Frequency list must be in ascending order” occurs.
- If a table has not been specified using the `MEMory:TABLe:SELect` command, the data cannot be entered into the table and error –221, “Settings conflict” occurs.
- If a frequency is set outside of the allowed frequency range, error –222, “Data out of range” occurs.

MEMory:TABLe:FREQuency:POINts?

This query returns the number of frequency points for the table currently selected. The response format is **<NRf>**. If no frequency values have been set, this query returns a 0. If no table is selected, this query returns **NAN**.

Syntax

Example

`MEM:TABL:FREQ:POIN?`	*Queries the number of frequency points in the current table.*

MEMory:TABLe:GAIN[:MAGNitude] <numeric_value>{,<numeric_value>}

This command is used to enter offsets into the frequency-dependent offset table, currently selected using `MEMory:TABLe:SELect`. Any previous offset list is cleared before the new offsets are stored.

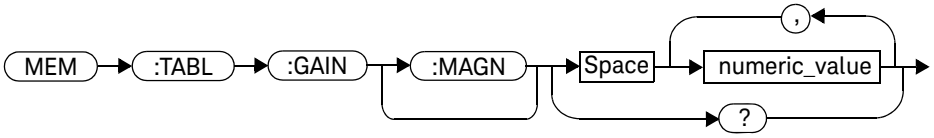
A maximum of 80 parameters for frequency-dependent offset tables can be sent with this command.

Entries in the frequency lists correspond as shown in Table 6-4 with entries in the offset factor lists.

Table 6-4 Frequency and offset factor list

Frequency	Offset
Frequency 1	Offset 1
"	"
Frequency 80	Offset 80

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value for the offset factor. The unit is PCT.	1.0 to 150.0

Example

```
MEM:TABL:SEL “Sensor_1”
MEM:TABL:GAIN 97,99.5,97.4
```

This command enters a reference offset factor of 97%, 99.5%, and 97.4% into the frequency-dependent offset table.

Query

```
MEMory:TABLe:GAIN[:MAGNitude]?
```

The query returns a list of offset points for the currently selected table.

Query example

```
MEM:TABL:GAIN?
```

Queries the offset in the current table.

Error messages

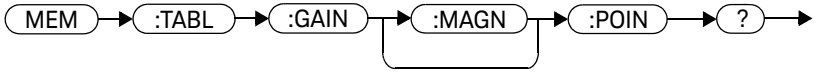
- If more than 80 offsets for the frequency-dependent offset tables are in the list, error –108, “Parameter not allowed” occurs.
- If a table is not specified using the `MEMory:TABLe:SELect` command, the data cannot be entered and error –221, “Settings conflict” occurs.
- If any of the offset factors are outside of the allowed range, error –222, “Data out of range” occurs.

MEMory:TABLe:GAIN[:MAGNitude]:POINts?

This query is used to return the number of offset points for the currently selected table.

If no values have been set, 0 is returned. If no table is selected, **NAN** is returned.

Syntax



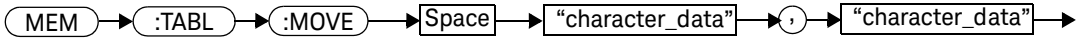
Example

`MEM:TABL:GAIN:POIN?`	*Queries the number of offset points in the current table.*

MEMory:TABLe:MOVE <“character_data”>,<“character_data”>

This command is used to rename a frequency-dependent offset table, a Gamma table, or an S-Parameter table.

Syntax



Parameters

Item	Description/Default	Range of values
character_data (1st parameter)	Contains the existing table name.	Existing table name.
character_data (2nd parameter)	Details the new table name. A maximum of 12 characters can be used.	A to Z (uppercase) a to z (lowercase) 0 to 9_ (underscore)

Example

```
MEM:TABL:MOVE “tab1”,”tab1a”
```

This command renames a table named tab1 to tab1a.

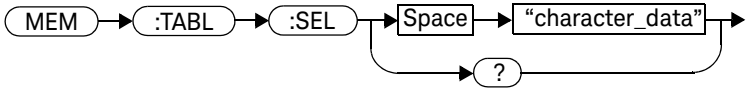
Error messages

- If either table names are invalid, error –224, “Illegal parameter value” occurs.
- If the first parameter does not match an existing table name, error –256, “File name not found” occurs.
- If the second parameter matches an existing table name or a save/recall register, error –257, “File name error” occurs.

MEMory:TABLe:SELect <“character_data”>

This command is used to activate a frequency-dependent offset table, a Gamma table, or an S-Parameter table. A table must be activated before any operation can be performed on it.

Syntax



Parameters

Item	Description/Default	Range of values
character_data	Details the new table name. A maximum of 12 characters can be used.	A to Z (uppercase) a to z (lowercase) 0 to 9_ (underscore)

Example

```
MEM:TABL:SEL “Sensor1”
```

This command selects a frequency-dependent offset table named “Sensor1”.

Query

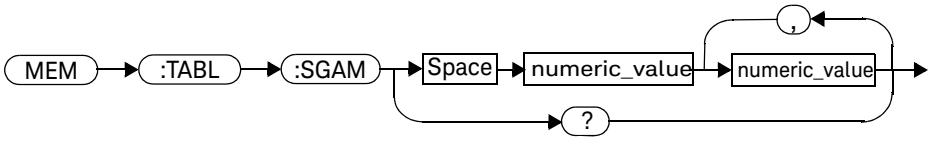
```
MEMory:TABLe:SELect?
```

The query returns the name of the currently selected table.

MEMory:TABLe:SGAMma <numeric_value>,<numeric_value> {,<numeric_value>}{,<numeric_value>}

This command sets the magnitude-phase pairs for the source gamma; for the currently selected Gamma Table. A Gamma Table needs to be selected before this command can be used. The maximum number of magnitude-phase pairs is 1024.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	Sets the magnitude-phase pair values	Magnitude: 0.0 to 0.999 Phase: $-180.0^0 \leq p < +180.0^0$

Example

`MEM:TABL:SGAM 1.0,160,0.45,60` *This command sets the magnitude-phase pair as 1.0 (mag1), 160 (phase1), 0.45 (mag2), and 60 (phase2).*

Query

`MEMory:TABLe:SGAMma?`

The query returns a list of magnitude-phase pairs for the currently selected Gamma table.

Query example

`MEM:TABL:SGAM?`	*Queries the magnitude-phase pairs for the currently selected Gamma table.*

Error messages

- If a table has not been specified using the `MEMory:TABLe:SELect` command, the data cannot be entered into the table and error –221, “Settings conflict” occurs.
- If a magnitude or phase which is outside the allowed range is sent, error –222, “Data out of range” occurs.

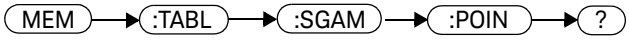
MEMory:TABLe:SGAMma:POINts?

This query is used to list the number of magnitude-phase pairs for the source gamma for the currently selected Gamma table.

If no magnitude-phase values have been set, this query returns a 0.

If no table is selected, this query returns **NAN**.

Syntax



Example

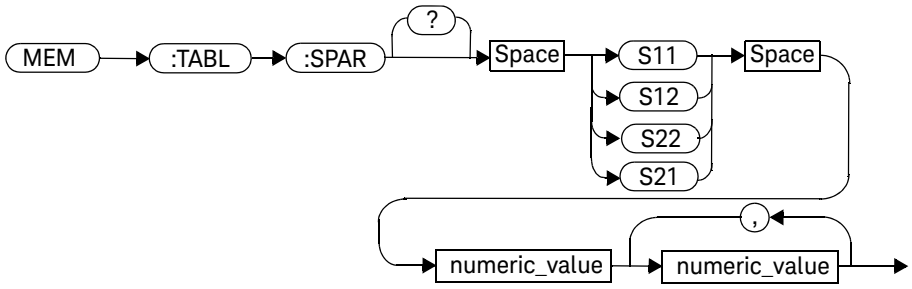
```
MEM:TABL:SGAM:POIN?
```

Queries the number of magnitude-phase pairs for the currently selected Gamma table.

MEMory:TABLe:SPARam <S11|S12|S21|S22>,<numeric_value>,<numeric_value> {,<numeric_value>}{,<numeric_value>}

This command sets the magnitude-phase pairs for the selected S-Parameter type; for the currently selected S-Parameter table. The maximum number of magnitude-phase pairs is 1024.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	Sets the magnitude-phase pair values.	For S11, S22 – Magnitude: (0.0 to 0.999) – Phase: ($-180.0^0 \leq p < +180.0^0$) – For S12, S21 – Magnitude: (1.0×10^{-5} to $1.0 \times 10^{+5}$) – Phase: ($-180.0^0 \leq p < +180.0^0$)

Example

`MEM:TABL:SPAR S11,0.3,100`	*This command sets the values 0.3 and 100 as a magnitude-phase pair for the S11 S-Parameter.*

Query

`MEMory:TABLe:SPARam? <S11|S12|S21|S22>`

The query returns a list of magnitude-phase pairs for the currently selected S-Parameter table.

Query example

`MEM:TABL:SPAR? S11`	*Queries the S11 magnitude-phase pairs for the currently selected S-Parameter.*

Error messages

- If a table has not been specified using the `MEMory:TABLe:SELect` command, the data cannot be entered into the table and error –221, “Settings conflict” occurs.

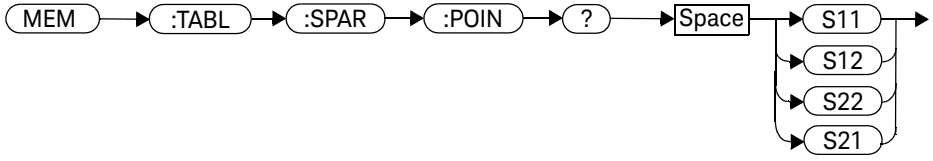
MEMory:TABLe:SPARam:POINts? <S11|S12|S21|S22>

This query is used to list the number of magnitude-phase pairs for the selected S-Parameter for the currently selected S-Parameter table.

If no magnitude-phase values have been set, this query returns a 0.

If no table is selected, this query returns **NAN**.

Syntax



Example

`MEM:TABL:SPAR:POIN? S11` *Queries the number of S11 magnitude-phase pairs for the currently selected S-Parameter table.*

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

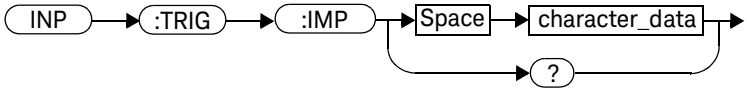
7 INPut Subsystem

This chapter explains how the **INPut** command subsystem is used to set the impedance of the U8480 Series trigger input port.

INPut:TRIGger:IMPedance [HIGH|LOW]

This command sets the impedance of the trigger input port.

Syntax



Parameters

Item	Description/Default	Range of values
character_data	Trigger input impedance: – **LOW**: 50 Ω (default) – **HIGH**: 1 MΩ	LOW HIGH

Example

```
INP:TRIG:IMP LOW
```
*Sets the trigger input i*mpedance to low (50 *Ω*) .

Reset condition

On reset, the trigger input impedance is set to **LOW**.

Query

`INPut:TRIGger:IMPedance?`

The query returns the current trigger input impedance setting.

Query example

`INP:TRIG:IMP?`	*Queries the setting of the trigger input impedance.*

Error message

If **<character_data>** is not set to **HIGH** or **LOW**, error –224, "Illegal parameter value" occurs.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

8 SENSe Subsystem

This chapter explains how the **SENSe** command subsystem directly affects device-specific settings used to make measurements.

[SENSe] Command Subsystem

The **SENSe** command subsystem directly affects device-specific settings used to make measurements. The **SENSe** command subsystem is optional because this is the primary function of the U8480 Series. The high-level command **CONFigure** uses the **SENSe** commands to prepare the U8480 Series for making measurements. At a lower level, **SENSe** enables you to change the parameters without completely re-configuring the U8480 Series.

The **SENSe** command subsystem also allows you to select the measurement and a frequency-dependent offset table, a Gamma table, or an S-Parameter table.

[SENSe[1]:]AVERage Commands

These commands control the measurement averaging which is used to improve measurement accuracy. They combine successive measurements to produce a new composite result.

The following commands are detailed in this section:

```
[SENSe[1]:]AVERage:COUNt <numeric_value>
[SENSe[1]:]AVERage:COUNt:AUTO <boolean>
[SENSe[1]:]AVERage:SDETect <boolean>
[SENSe[1]:]AVERage[:STATe] <boolean>
```

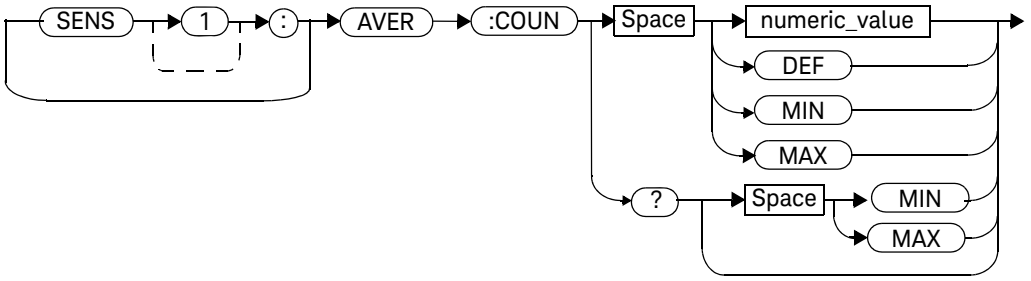
[SENSe[1]:]AVERage:COUNt <numeric_value>

This command is used to enter a value for the filter length. If `[SENSe[1]:]AVERage:COUNt:AUTO` is set to `ON`, then entering a value for the filter length automatically sets it to `OFF`. Increasing the value of filter length increases measurement accuracy but also increases the time taken to make a power measurement.

Entering a value using this command automatically turns the `[SENSe[1]:]AVERage:STATe` command to `ON`.

NOTE For most applications, automatic filter length selection (`[SENSe[1]:]AVERage:COUNt:AUTO ON`) is the best mode of operation. However, manual filter length selection (`[SENSe[1]:]AVERage:COUNt <numeric_value>`) is useful in applications requiring either high resolution or fast settling times, where signal variations rather than measurement noise need filtering, or when approximate results are needed quickly.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value defining the filter length: – DEF: The default value is 4 – MIN: 1 – MAX: 1024	1 to 1024 DEF MIN MAX

Example

```
AVER:COUN 400
```
This command sets a filter length of 400.

Reset condition

On reset, the filter length is set to four.

Query

```
[SENSe[1]:]AVERage:COUNt? [MIN|MAX]
```

The query returns the current setting of the filter length or the values associated with MIN and MAX. The format of the response is <NR1>.

Query example

```
AVER:COUN?
```
Queries the filter length.

Error message

- If a filter length value entered is outside the allowable range of values, error –222, “Data out of range” occurs.
- If a filter length value is entered using [SENSe[1]:]AVERage:COUNt while [SENSe[1]:]MRATe is set to FAST and TRIG:COUNt > 1, error –221, “Settings conflict” occurs. However, the filter length value is set but the [SENSe[1]:]AVERage:STATe command is not automatically set to ON.

[SENSe[1]:]AVERage:COUNt:AUTO <boolean>

This command enables and disables automatic averaging.

When the auto-filter mode is enabled, the U8480 Series automatically sets the number of readings averaged together to satisfy the averaging requirements for most power measurements. The number of readings averaged together depends on the resolution and the power level in which the U8480 Series is currently operating. Figure 8-1 is an example of the averaged number of readings for each range and resolution when the U8480 Series is in the auto measurement average mode.

Setting this command to ON automatically sets the `[SENSe[1]:]AVERage:STATe` command to ON.

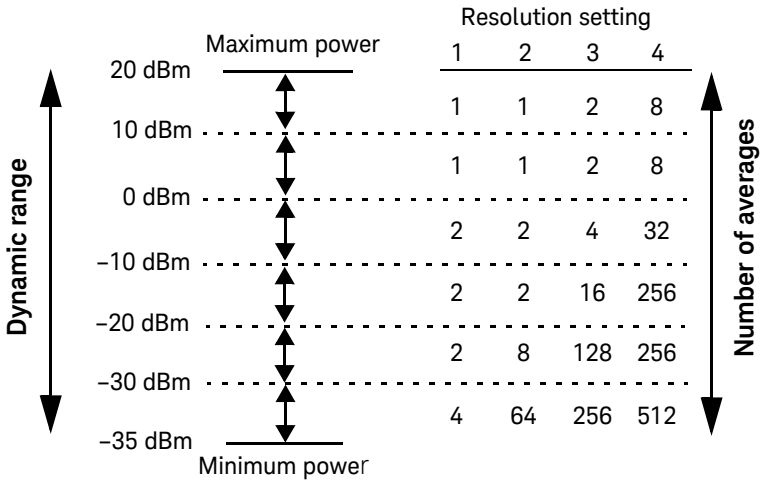


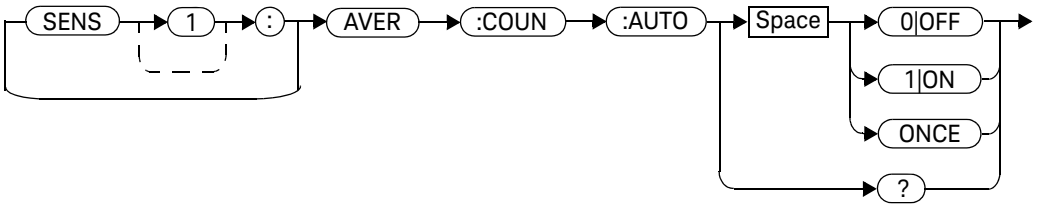
Figure 8-1 Example of averaged readings

If `[SENSe[1]:]AVERage:COUNt:AUTO` is set to `OFF`, the filter length is set by the `[SENSe[1]:]AVERage:COUNt` command. Using the `[SENSe[1]:]AVERage:COUNt` command disables automatic averaging.

Auto-averaging is enabled by the `MEASure:POWer:AC?` and `CONFigure:POWer:AC?` queries.

NOTE For most applications, automatic filter length selection ([SENSe[1]:]AVERage:COUNt:AUTO ON) is the best mode of operation. However, manual filter length selection ([SENSe[1]:]AVERage:COUNt <numeric_value>) is useful in applications requiring either high resolution or fast settling times, where signal variations rather than measurement noise need filtering, or when approximate results are needed quickly.

Syntax



Example

```
AVER:COUN:AUTO OFF
```

This command disables automatic filter length selection for the U8480 Series.

Reset condition

On reset, automatic averaging is enabled.

Query

[SENSe[1]:]AVERage:COUNt:AUTO?

The query enters a 1 or 0 into the output buffer indicating whether automatic filter length is enabled or disabled.

- 1 is returned when automatic filter length is enabled
- 0 is returned when automatic filter length is disabled

Query example

AVER:COUN:AUTO?	*Queries whether automatic filter length selection is turned on or off.*

[SENSe[1]:]AVERage:SDETect <boolean>

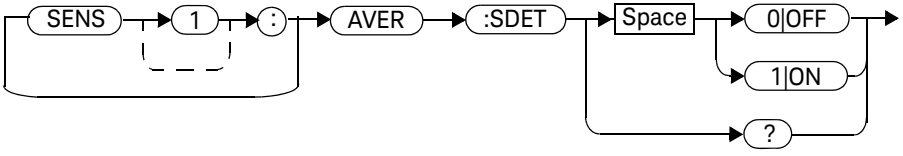
This command enables and disables step detection. In **AUTO** filter mode, the average of the last four values entered into the filter is compared to the average of the entire filter. If the difference between the two averages is greater than 12.5%, the digital filter is cleared. The filter then starts storing new measurement values. This feature shortens the filter time when the input power changes substantially for the filter output to get to its final value. Note that this result appears to settle faster, although true settling to the final value is unaffected.

NOTE Step detection is automatically disabled when `TRIGger[1]:DELay:AUTO` is `ON` and the trigger mode is not set to free run.

Under these circumstances, the value of `[SENSe[1]:]AVERage:SDETect` is ignored. `[SENSe[1]:]AVERage:SDETect` is not set by the U8480 Series (that is, `[SENSe[1]:]AVERage:SDETect` retains its current setting which may indicate that step detection is `ON`).

NOTE With certain pulsing signals, step detect may operate on the pulses, preventing the final average being completed and making the results unstable. Under these conditions, `SDETect` should be set to `OFF`.

Syntax



Example

```
AVER:SDET OFF
```
This command disables step detection.

Reset condition

On reset, step detection is enabled.

Query

```
[SENSe[1]:]AVERage:SDETect?
```

The query enters a 1 or 0 into the output buffer indicating the status of step detection.

- 1 is returned when step detection is enabled
- 0 is returned when step detection is disabled

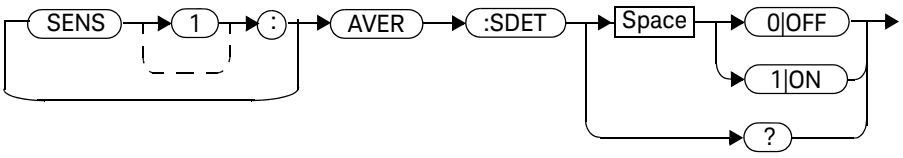
Query example

```
AVER:SDET?
```
Queries whether step detection is turned on or off.

[SENSe[1]:]AVERage[:STATe] <boolean>

This command is used to enable and disable averaging.

Syntax



Example

```
AVER 1
```
This command enables averaging.

Reset condition

On reset, averaging is turned **ON**.

Query

```
[SENSe[1]:]AVERage[:STATe]?
```

The query enters a 1 or 0 into the output buffer indicating the status of averaging.

- 1 is returned when averaging is enabled
- 0 is returned when averaging is disabled

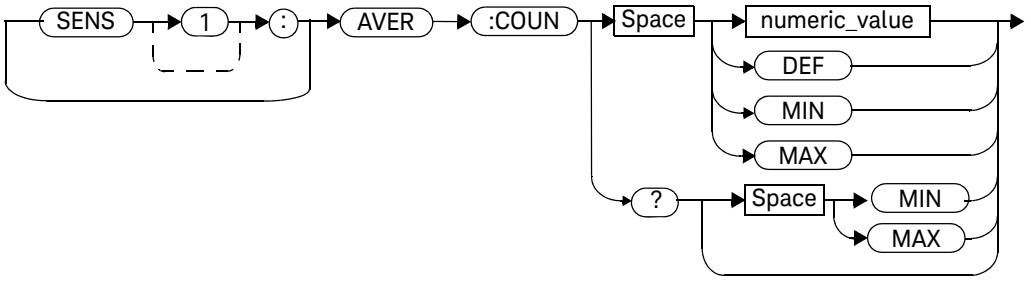
Query example

```
AVER?
```
Queries whether averaging is enabled or disabled.

[SENSe[1]:]BUFFer:COUNt <numeric_value>

This command sets the buffer size for average trigger measurement. It must be used in conjunction with an external trigger.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value for the buffer size: – **DEF**: 1 – **MIN**: 1 – **MAX**: 250	1 to 250 **DEF** **MIN** **MAX**

Example

```
BUFF:COUN 100
```

This command sets the average trigger measurement buffer size to 100.

Reset condition

On ***RST**, the value is set to 1.

Query

`[SENSe[1]:]BUFFer:COUNt? [MIN|MAX]`

This query returns the average trigger measurement buffer size or the values associated with **MIN** and **MAX**.

Query example

BUFF:COUN? *This query returns the average trigger measurement buffer size.*

Error messages

- If **TRIGger:SOURce** is not set to **EXT**, error –221, “Settings conflict” occurs.
- If the **[SENSe[1]:]BUFFer:COUNt** parameter is set <1, error –222, “Data out of range” occurs.
- If the **[SENSe[1]:]BUFFer:COUNt** parameter is set >250, error –222, “Data out of range” occurs.
- If the frequency sweep step is non-zero, error –221, “Settings conflict” occurs.

[SENSe[1]:]CORRection:CSET2 Commands

These commands are used to select the active frequency-dependent offset table.

The following commands are detailed in this section:

```
[SENSe[1]:]CORRection:CSET2[:SELect] <string>
[SENSe[1]:]CORRection:CSET2:STATe <boolean>
```

[SENSe[1]:]CORRection:CSET2[:SELect] <“string”>

This command enters the name of the frequency-dependent offset table which is to be used.

NOTE If `[SENSe[1]:]CORRection:CSET2:STATe` is set to `OFF`, the selected frequency-dependent offset table is not being used.

Syntax

Parameters

Item	Description/Default	Range of values
string	String data representing a frequency-dependent offset table name.	Any existing table name (Existing table names can be listed using `MEMory:CATalog:TABLe?`).

Example

`CORR:CSET2 “PW1”` *This command enters the name of the frequency-dependent offset table which is to be used.*

Reset condition

On reset, the selected table is not affected.

Query

`[SENSe[1]:]CORRection:CSET2[:SELect]?`

The query returns the name of the selected table as a quoted string. If no table is selected, an empty string is returned.

Query example

`CORR:CSET2?` *Queries the frequency-dependent offset table currently used.*

Error messages

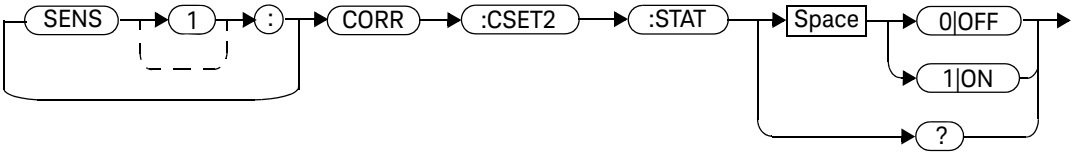
- If **<"string">** is invalid, error –224, "Illegal parameter value" occurs.
- If a table called **<"string">** does not exist, error –256, "File name not found" occurs.
- When a frequency-dependent offset table is selected, the U8480 Series verifies that the number of offset points defined is equal to the number of frequency points defined. If this is not the case, error –226, "Lists not the same length" occurs.

[SENSe[1]:]CORRection:CSET2:STATe <boolean>

This command enables and disables the use of the currently active frequency-dependent offset table (**CSET2**). When a table has been selected and enabled, the frequency-dependent offsets stored in it can be used by specifying the required frequency using the **[SENSe[1]:]FREQuency[:CW|:FIXed]** command.

When the **CSET2** command is set to **ON**, the frequency-dependent offset is taken from the frequency-dependent offset table.

Syntax



Example

```
CORR:CSET2:STAT 1
```

This command enables the use of the currently active frequency-dependent offset table.

Reset condition

On reset, the selected table is not affected.

Query

`[SENSe[1]:]CORRection:CSET2:STATe?`

The query returns a 1 or 0 into the output buffer indicating whether a table is enabled or disabled.

- 1 is returned when the table is enabled
- 0 is returned when the table is disabled

Query example

`CORR:CSET2:STAT?`	*Queries whether there is currently an active frequency-dependent offset table for the U8480 Series.*

Error messages

If you attempt to set this command to `ON` and no table has been selected using `[SENSe[1]:]CORRection:CSET2[:SELect]`, error –221, "Settings conflict" occurs and `[SENSe[1]:]CORRection:CSET2:STATe` remains `OFF`.

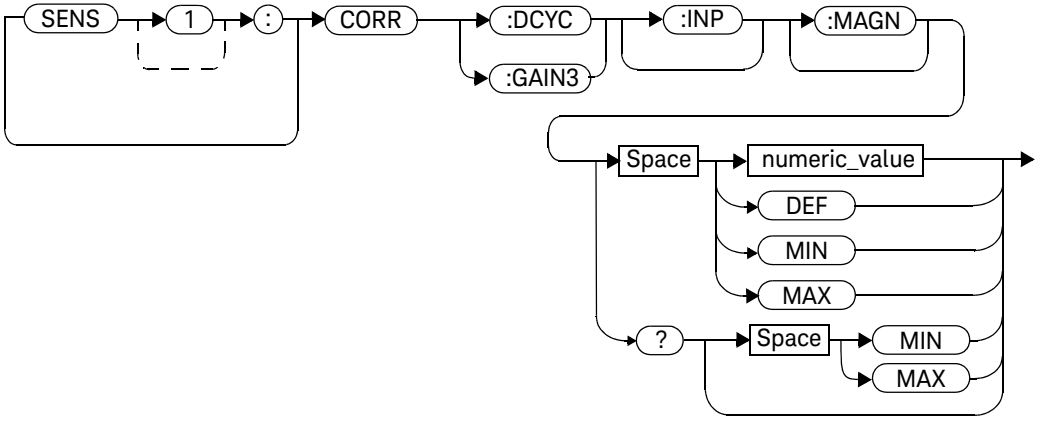
[SENSe[1]:]CORRection:DCYCle|GAIN3[:INPut][:MAGNitude] <numeric_value>

This command is used to set the duty cycle for the pulse power measurement. Pulse power measurements average out any aberrations in the pulse such as overshoot or ringing. The result returned for a pulse power measurement is a mathematical representation of the pulse power rather than an actual measurement.

The U8480 Series measures the average power in the pulsed input signal and then divides the result by the duty cycle value to obtain a pulse power reading.

Entering a value using this command automatically turns the `[SENSe[1]:]CORRection:DCYCle|GAIN3:STATe` command to `ON`.

Syntax



Parameters

Item	Description/Default	Range of values

numeric_value	A numeric value for the duty cycle:
	– `DEF`: The default value is 1%
	– `MIN`: 0.001%
	– `MAX`: 99.999%
	The unit is PCT and is optional.
		0.001 to 99.999 PCT
		`DEF`
		`MIN`
		`MAX`

Example

```
CORR:DCYC 90PCT
```
This command sets a duty cycle of 90%.

Reset condition

On reset, the duty cycle is set to 1% (`DEF`).

Query

```
[SENSe[1]:]CORRection:DCYCle|GAIN3[:INPut][:MAGNitude]?
[MIN|MAX]
```

The query returns the current setting of the duty cycle or the values associated with `MIN` and `MAX`.

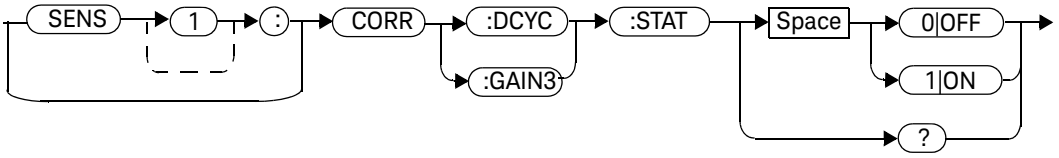
Query example

```
CORR:GAIN3?
```
Queries the current setting of the duty cycle.

[SENSe[1]:]CORRection:DCYCle|GAIN3:STATe <boolean>

This command is used to enable and disable the pulse power measurement.

Syntax



Example

```
CORR:DCYC:STAT 1
```

This command enables the pulse power measurement.

Reset condition

On reset, the pulse power measurement is disabled.

Query

`[SENSe[1]:]CORRection:DCYCle|GAIN3:STATe?`

The query enters a **`1`** or **`0`** into the output buffer indicating the status of the pulse power measurement.

- 1 is returned when the pulse power measurement is enabled.
- 0 is returned when the pulse power measurement is disabled.

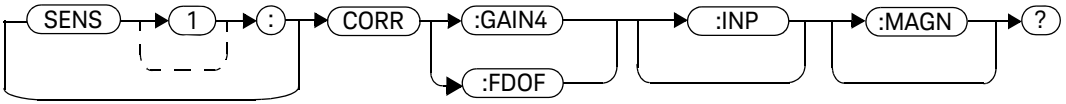
Query example

`CORR:GAIN3:STAT?`	*Queries whether the pulse power measurement is turned on or off.*

[SENSe[1]:]CORRection:FDOFset|GAIN4[:INPut][:MAGNitude]?

This query returns the frequency-dependent offset currently being applied.

Syntax



Example

`CORR:GAIN4?` *Queries the current frequency-dependent offset being applied to the measurement.*

Reset condition

On reset, the frequency-dependent offset is not affected.

[SENSe[1]:]CORRection:GAIN2 Commands

These commands provide a simple correction to a measurement for an external gain/loss.

The following commands are detailed in this section:

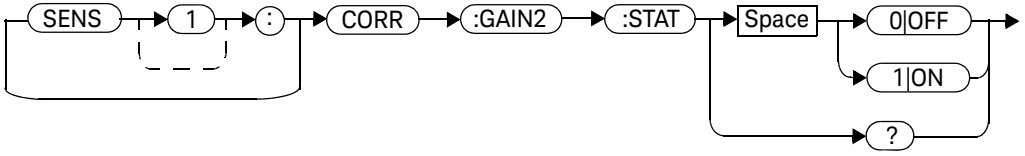
```
[SENSe[1]:]CORRection:GAIN2:STATe <boolean>
[SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude] <numeric_value>
```

[SENSe[1]:]CORRection:GAIN2:STATe <boolean>

This command is used to enable and disable a channel offset for the U8480 Series setup.

The **[SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude]** command is used to enter the loss/gain value.

Syntax



Example

```
CORR:GAIN2:STAT ON
```
This command enables a channel offset.

Reset condition

On reset, the channel offset is disabled.

Query

`[SENSe[1]:]CORRection:GAIN2:STATe?`

The query enters **1** or **0** into the output buffer indicating the status of the channel offset.

- 1 is returned if a channel offset is enabled.
- 0 is returned if a channel offset is disabled.

Query example

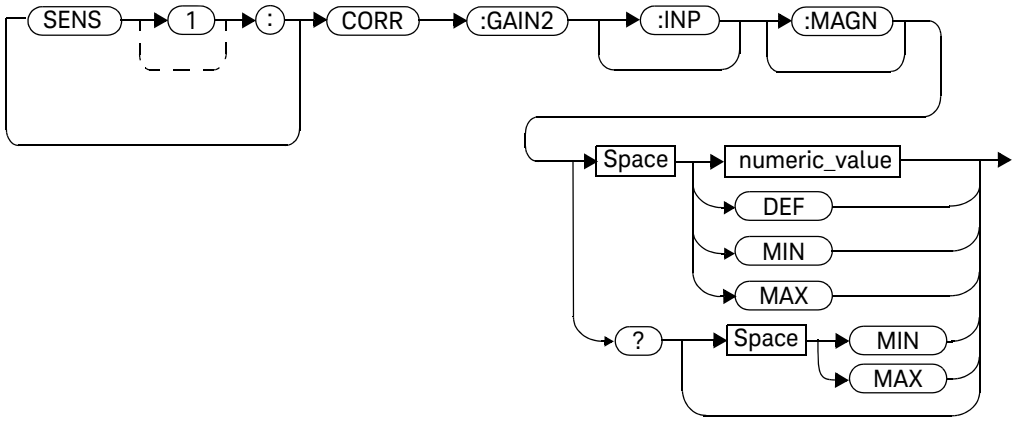
`CORR:GAIN2:STAT?`	*Queries whether or not there is a channel offset applied.*

[SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude] <numeric_value>

This command is used to enter a channel offset value for the U8480 Series setup. The U8480 Series then corrects every measurement by this factor to compensate for the gain/loss.

Entering a value for **GAIN2** using this command automatically turns the **[SENSe[1]:]CORRection:GAIN2:STATe** command to **ON**.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value: – **DEF**: The default is 0.00 dB – **MIN**: –100 dB – **MAX**: +100 dB	–100 to +100 dB **DEF** **MIN** **MAX**

Example

```
CORR:GAIN2 50
```
This command sets a channel offset of 50 dB.

Reset condition

On reset, **GAIN2** is set to 0.00 dB.

Query

```
[SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude]? [MIN|MAX]
```

The query returns the current setting of the channel offset or the values associated with **MIN** and **MAX**.

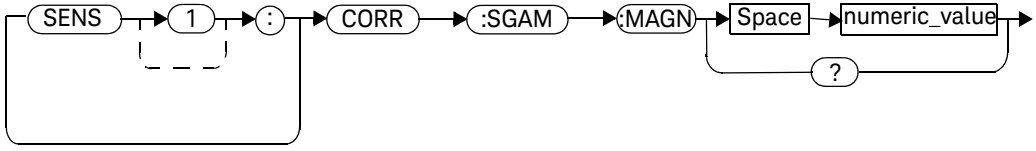
Query example

```
CORR:GAIN2?
```
Queries the current setting of the channel offset.

[SENSe[1]:]CORRection:SGAMma:MAGNitude <numeric_value>

This command is used to set the magnitude of the source reflection coefficient, Γ_G.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value: – The default is 0.0	0.0 to 0.999

Example

`CORR:SGAM:MAGN 0.5` *This command sets the magnitude of the source reflection coefficient at 0.5.*

Reset condition

On preset (`SYSTem:PRESet`) and U8480 Series power-up, `[SENSe[1]:]CORRection:SGAMma:MAGNitude` is set to `0.0`.

Query

`[SENSe[1]:]CORRection:SGAMma:MAGNitude?`

The query returns the magnitude of the source reflection coefficient.

Query example

`CORR:SGAM:MAGN?` *Queries the current magnitude of the source reflection coefficient.*

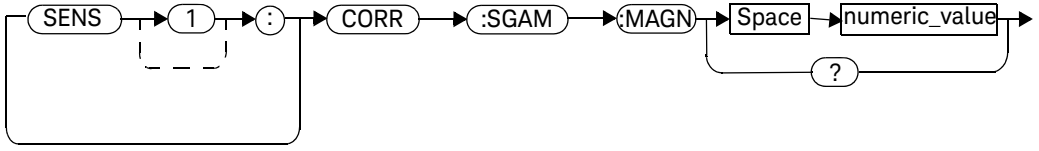
Error message

– If the input values are outside the acceptable range of values, error –222 “Data out of range” occurs.

[SENSe[1]:]CORRection:SGAMma:PHASe <numeric_value>

This command is used to set the phase of the source reflection coefficient, Γ_G.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value: – The default is 0	$-180.0^0 \leq p < +180.0^0$

Example

`CORR:SGAM:PHAS 45`	*This command sets the phase of the source reflection coefficient at 45.*

Reset condition

On preset (`SYSTem:PRESet`) and U8480 Series power-up, `[SENSe[1]:]CORRection:SGAMma:PHASe` is set to `0.0`.

Query

`[SENSe[1]:]CORRection:SGAMma:PHASe?`

The query returns the phase of the source reflection coefficient.

Query example

`CORR:SGAM:PHAS?`	*Queries the current phase of the source reflection coefficient.*

Error message

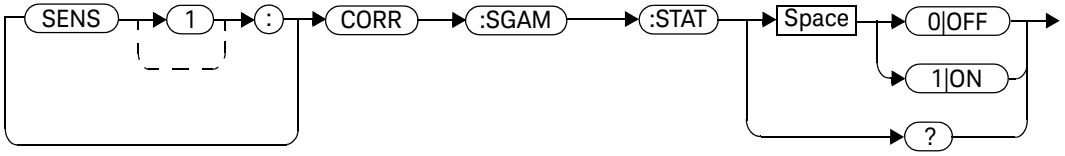
– If the input values are outside the acceptable range of values, error –222 “Data out of range” occurs.

[SENSe[1]:]CORRection:SGAMma:STATe <boolean>

This command is used to enable or disable Single Point Gamma correction.

Values from **[SENSe[1]:]CORRection:SGAMma:MAGNitude** and **[SENSe[1]:]CORRection:SGAMma:PHASe** apply across all frequency values and are used for correction when this is enabled.

Syntax



Reset condition

On preset (**SYSTem:PRESet**) and U8480 Series power-up, **[SENSe[1]:]CORRection:SGAMma:STATe** is set to **OFF**.

Query

[SENSe[1]:]CORRection:SGAMma:STATe?

The query returns the Single Point Gamma correction state.

Query example

`CORR:SGAM:STAT?`	*Queries the current state of the Single Point Gamma correction.*

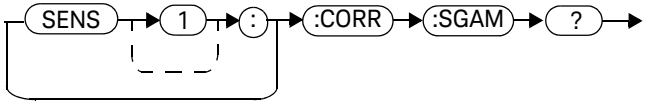
Error message

– If you set this command to ON and [`SENSe[1]:]CORRection:CSET3:STATe` is currently ON, it will set `[SENSe[1]:]CORRection:CSET3:STATe` to OFF and error –221, “Settings conflict; Table based gamma is being switched off” will occur. This behaviour indicates that both `[SENSe[1]:]CORRection:SGAMma:STATe` and `[SENSe[1]:]CORRection:CSET3:STATe` are mutually exclusive.

[SENSe[1]:]CORRection:SGAMma?

This query returns the source gamma magnitude-phase pair which is currently being used in gamma correction.

Syntax



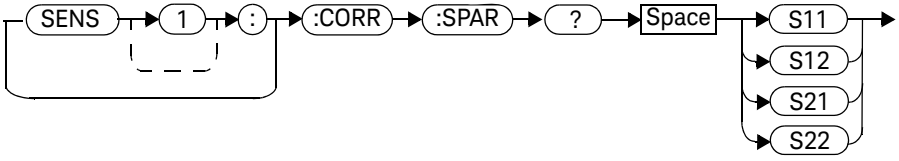
Example

CORR:SGAM? *This query returns the source gamma magnitude-phase pair which is currently being used in gamma correction.*

[SENSe[1]:]CORRection:SPARam? <S11|S12|S21|S22>

This query returns the current magnitude-phase values for the selected S-Parameter type.

Syntax



Parameters

Item	Description/Default	Range of values
<S11|S12|S21|S22>	Magnitude-phase values	For S11, S22 – Magnitude: (0.0 to 0.999) – Phase: ($-180.0^{o} \leq p < +180.0^{o}$) For S12, S21 – Magnitude: (1.0×10^{-5} to $1.0 \times 10^{+5}$) – Phase: ($-180.0^{o} \leq p < +180.0^{o}$)

Example

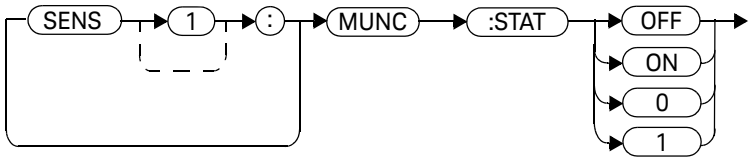
`CORR:SPAR?` *This query returns the current magnitude-phase values for the selected S-Parameter type.*

[SENSe[1]:]MUNC:STATe OFF|ON|0|1

This command is used to enable or disable the measurement uncertainty feature.

The default state is OFF.

Syntax



Reset condition

On preset (**SYSTem:PRESet**) and U8480 Series power-up, **[SENSe[1]:]MUNC:STATe** is set to **OFF**.

Query

[SENSe[1]:]MUNC:STATe?

The query returns the state of the measurement uncertainty feature.

Query example

MUNC:STAT?	*Queries the current state of the measurement uncertainty feature.*

[SENSe[1]:]MUNC:SGAMma:TYPE? SINGle|TABLe|SPARam

The command selects the type of source Gamma that will be used in the calculation of the Device-Under-Test (DUT) mismatch for the Measurement Uncertainty feature. There are three types of source gamma that can be selected; Single Point Gamma, Table-Based Gamma, and the S-Parameter Table.

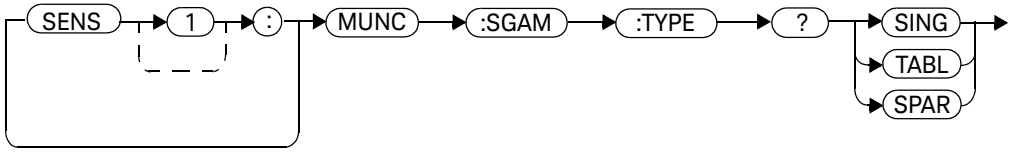
The default source gamma is Single Point Gamma.

The value of `[SENSe[1]:]MUNC:SGAM:TYPE` will automatically change to reflect the setting of `[SENSe[1]:]CORRection:SGAMma:STATe` and `[SENSe[1]:]CORRection:CSET3:STATe`, unless its current selection is `SPARam`

If `[SENSe[1]:]CORRection:CSET4:STATe` is set to ON, `[SENSe[1]:]MUNC:SGAM:TYPE` will change to `SPARam`.

NOTE **If there is a 2-port device connected to the sensor, select `SPARam` as the source gamma.**

Syntax



Reset condition

On preset (`SYSTem:PRESet`) and U8480 Series power-up, `[SENSe[1]:]MUNC:SGAM:TYPE` is set to `SINGle`.

Query

SENSe[1]:]MUNC:SGAM:TYPE?

The query returns the type of the source gamma used for the measurement uncertainty feature.

Query example

[SENSe[1]:]MUNC:SGAM:TYPE?	*Queries the type of the source gamma used for the measurement uncertainty feature.*

Error message

- If you attempt to set **[SENSe[1]:]MUNC:SGAM:TYPE to SINGle|TABLe** when **[SENSe[1]:]CORRection:CSET4:STATe** is currently ON, error –221, “Settings conflict” occurs and the current setting of **[SENSe[1]:]MUNC:SGAM:TYPE** remains unchanged.

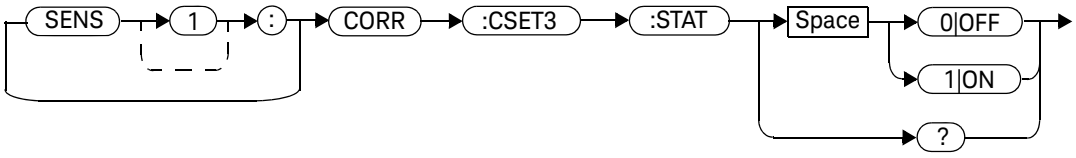
[SENSe[1]:]CORRection:CSET3:STATe <boolean>

This command is used to enable or disable the Table-Based Gamma correction.

If this is enabled, gamma values from the currently selected Gamma table will be used for correction.

NOTE **This is mutually exclusive with the `[SENSE[1]:]CORRection:SGAMma:STATe` command.**

Syntax



Reset condition

On preset (**`SYSTem:PRESet`**) and U8480 Series power-up, **`[SENSe[1]:]CORRection:CSET3:STATe`** is set to **`OFF`**.

Query

`[SENSe[1]:]CORRection:CSET3:STATe?`

The query returns the state of the Table-Based Gamma correction.

Query example

`CORR:CSET3:STAT?` *Queries the current state of the Table-Based Gamma correction.*

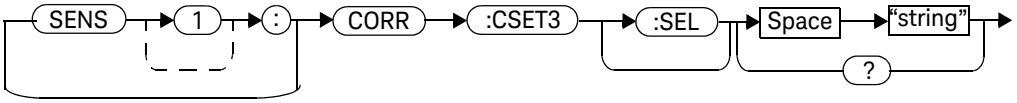
Error message

- If you attempt to set this command to ON and no table has been selected using `[SENSe[1]:]CORRection:CSET3[:SELect]`, then error –221, "Settings conflict" occurs and `[SENSe[1]:]CORRection:CSET3:STATe` remains `OFF`.
- If you set this command to ON and `[SENS[1]:]CORRection:SGAMma:STATe` is currently ON, `[SENS[1]:]CORRection:SGAMma:STATe` will be set to OFF and error –221, "Settings conflict; Single point gamma is being switched off" occurs.

[SENSe[1]:]CORRection:CSET3:[SELect] <“string”>

This command is used to select the Gamma table to be used in the Table-Based Gamma correction.

Syntax



Query

```
[SENSe[1]:]CORRection:CSET3:[SELect]?
```

The query returns the currently selected Gamma table used in the Table-Based Gamma correction.

If no table is selected, the query returns an empty string.

Query example

`CORR:CSET3:SEL?`	*Queries the current table used in the Table-Based Gamma correction.*

Error message

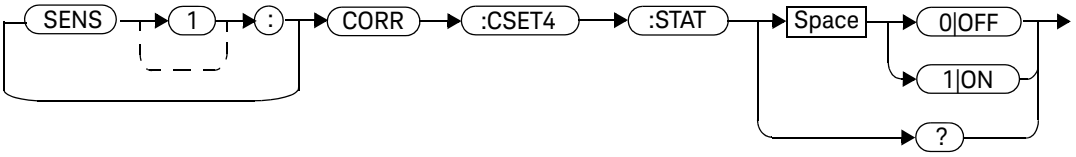
- If the string is not valid, error –224, “Illegal parameter value” occurs.
- If a table does not exist, error –256, “File name not found” occurs.
- When a gamma table is selected, the power sensor verifies that the number of magnitude-phase offset pairs defined is equal to the number of frequency points defined. If they do not match, error –226, “Lists not the same length” occurs.

[SENSe[1]:]CORRection:CSET4:STATe <boolean>

This command is used to enable or disable S-Parameter correction.

When this is enabled, S-Parameter values from the selected S-Parameter table will be used for correction.

Syntax



Reset condition

On preset (**SYSTem:PRESet**) and U8480 Series power-up, **[SENSe[1]:]CORRection:CSET4:STATe** is set to **ON**.

Query

[SENSe[1]:]CORRection:CSET4:STATe?

The query returns the S-Parameter correction state.

Query example

CORR:CSET4:STAT? *Queries the current state of the S-Parameter correction.*

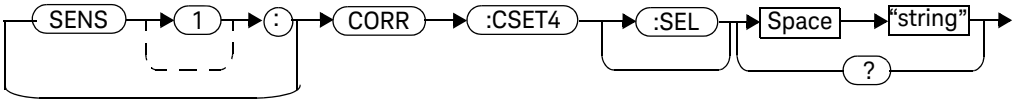
Error message

– If you set this command to ON and no table has been selected using **[SENSe[1]:]CORRection:CSET4[:SELect]**, then error –221, "Settings conflict" occurs and **[SENSe[1]:]CORRection:CSET4:STATe** remains OFF

[SENSe[1]:]CORRection:CSET4:[SELect] <“string”>

This command is used to select the S-Parameter table to be used for S-Parameter correction.

Syntax



Query

`[SENSe[1]:]CORRection:CSET4:[SELect]?`

The query returns the currently selected S-Parameter table used for S-Parameter correction.

If no table is selected, the query returns an empty string.

Query example

`CORR:CSET4:SEL?` *Queries the current table used in the S-Parameter correction.*

Error message

- If the string is not valid, error –224, “Illegal parameter value” occurs.
- If a table does not exist, error –256, “File name not found” occurs.
- When an S-Parameter table is selected, the power sensor verifies that the number of magnitude-phase pairs defined for S11, S12, S21, and S22 is equal to the number of frequency points defined. If they do not match, error –226, “Lists not the same length” occurs.

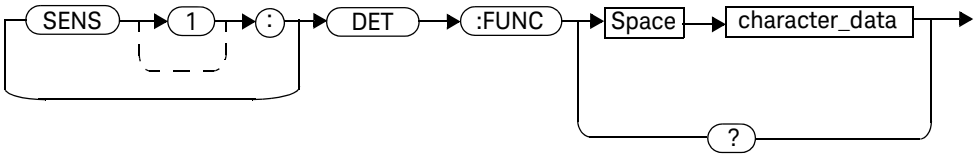
[SENSe[1]:]DETector:FUNCtion <character_data>

This command is used to set the measurement mode for the U8480 Series to **AVERage** (chopper-based measurement).

When **AVERage** is set, the following events occur:

- If **TRIGger:SOURce** is set to **EXTernal**, it is set automatically to **IMMediate**.
- **INITiate:CONTinuous** is set automatically to **ON**.
- **CALCulate:FEED** is set automatically to **"POW:AVER"** for the **CALC** block.

Syntax



Parameters

Item	Description/Default	Range of values
character_data	Defines the measurement mode: – **AVERage**: sets the U8480 Series to the average only mode	AVERage

Example

```
DET:FUNC AVER
```

Sets the U8480 Series to the average only mode.

Reset condition

On reset, the measurement mode is set to **AVERage**.

Query

```
[SENSe[1]:]DETector:FUNCtion?
```

The query returns the current measurement mode for the U8480 Series.

Query example

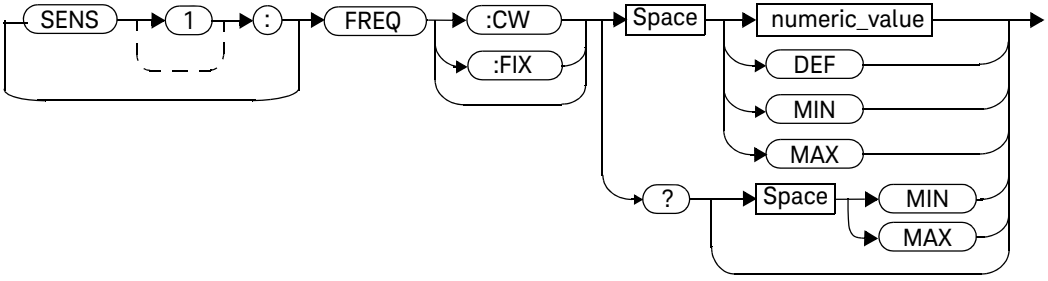
```
DET:FUNC?
```

Queries the current measurement mode for the U8480 Series.

[SENSe[1]:]FREQuency[:CW|:FIXed] <numeric_value>

This command is used to enter a frequency. The appropriate frequency-dependent offset, Gamma, or S-Parameter corrections are applied for the frequency selected, dependent on the frequency-dependent offset, Gamma, or S-Parameter data stored in the non-volatile memory.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value for the frequency: – **DEF**:The default value is 50 MHz – **MIN** : 0 Hz – **MAX** : 1000 GHz The default unit is Hz.	0 Hz to 1000 GHz[a] **DEF** **MIN** **MAX**

[a] The following measurement units can be used:

- Hz
- kHz (10^3)
- MHz (10^6)
- GHz (10^9)

Example

```
FREQ 50MHz
```
This command enters a frequency of 50 MHz.

Reset condition

On reset, the frequency is set to 50 MHz (**DEF**).

Query

```
[SENSe[1]:]FREQuency[:CW|:FIXed]? [MIN|MAX]
```

The query returns the current frequency setting or the values associated with **MIN** and **MAX**. The units in which the results are returned are Hz.

Query example

```
FREQ?
```
Queries the frequency setting.

[SENSe[1]:]FREQuency[:CW|:FIXed]:STARt <numeric_value> <unit>

This command sets the start frequency of the frequency sweep. It must be used in conjunction with an external trigger.

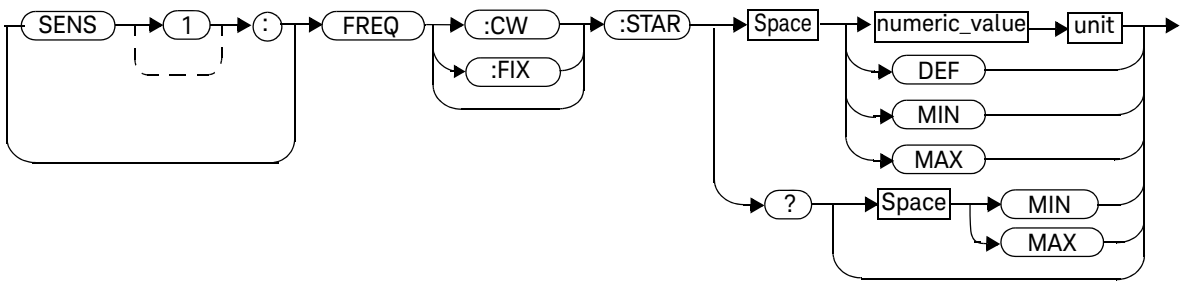
If frequency sweep is disabled (frequency sweep step set to 0), the start frequency will be set, but it will not take effect.

`[SENSe[1]:]FREQuency[:CW|:FIXed]:STARt`,
`[SENSe[1]:]FREQuency[:CW|:FIXed]:STOP`, and
`[SENSe[1]:]FREQuency[:CW|:FIXed]:STEP` are allowed to be set in any desirable sequence.

When the frequency sweep mode is configured with a frequency step size within the allowable range, the following conditions apply:

- If the frequency stop point is greater than the frequency start point, the frequency range will be swept in an ascending order.
- If the frequency stop point is less than the frequency start point, the frequency range will be swept in a descending order.
- If the frequency stop point and the frequency start point are equal, it is equivalent to the power sweep mode.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value for the start frequency: – **DEF**: the default value is 50 MHz – **MIN**: 0 Hz – **MAX**: 1000.0 GHz The default unit is Hz.	0 Hz to 1000.0 GHz[a] **DEF** **MIN** **MAX**

[a] The following measurement units can be used:

- Hz
- kHz (10^3)
- MHz (10^6)
- GHz (10^9)

Example

```
FREQ:STAR 10MHz
```

Sets the frequency sweep to start at 10 MHz.

Reset condition

On reset, the value is set to 50 MHz.

Query

`[SENSe[1]:]FREQuency[:CW|:FIXed]:STARt?`

This query retrieves the start frequency of the average trigger frequency sweep. Frequency is returned in Hz.

Query example

`FREQ:STAR?` *Returns the start frequency of the frequency sweep in Hz.*

Error message

If the parameter set is <0 Hz or >1000 GHz, error –222, “Data out of range” occurs.

[SENSe[1]:]FREQuency[:CW|:FIXed]:STEP <numeric_value>

This command sets the number of steps in a frequency sweep. It must be used in conjunction with an external trigger.

The frequency will be swept in linearly spaced intervals from the start to the stop frequencies, by the number of steps.

Determine the right step to be set

The number of frequency steps can be calculated with the equation below:

$$Step = \frac{fstop - fstart + Interval}{Interval}$$

where,

Step = Number of frequency steps

fstart = Start frequency

fstop = Stop frequency

Interval = Interval between each frequency points

Example

If you wish to sweep from 1 GHz to 5 GHz, in intervals of 0.5 GHz, the step is derived as follows:

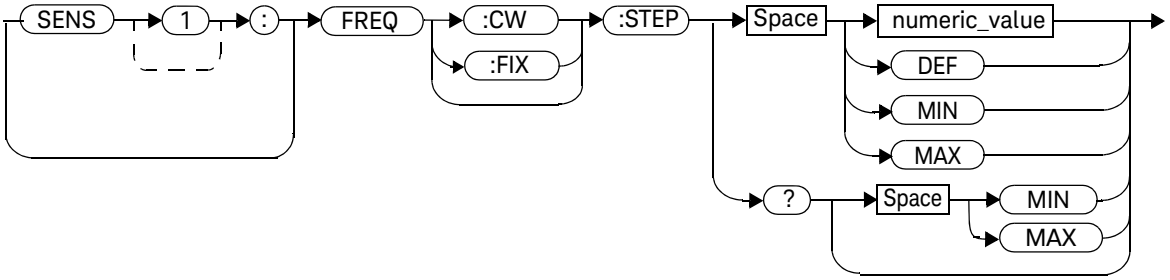
$$Step = \frac{fstop - fstart + Interval}{Interval}$$

$$Step = \left(\frac{5 - 1 + 0.5}{0.5}\right) = 9$$

`[SENSe[1]:]FREQuency[:CW|:FIXed]:STARt`, `[SENSe[1]:]FREQuency[:CW|:FIXed]:STOP`, and `[SENSe[1]:]FREQuency[:CW|:FIXed]:STEP` may be set in any sequence.

The frequency interval is rounded to the nearest kHz, with a minimum interval of 1 kHz. If the stop frequency exceeds the maximum frequency of the U8480 Series, the remaining steps will be repeated with the last frequency point.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	Number of steps in the triggered frequency sweep: – **DEF** : the default value is 0 – **MIN** :0 – **MAX** :250	0 to 250 **DEF** **MIN** **MAX**

Example

```
FREQ:STEP 10
```
Sets the frequency sweep with 10 steps.

Reset condition

On reset, the value is set to 0.

Query

`[SENSe[1]:]FREQuency[:CW|:FIXed]:STEP?`

This query retrieves the number of steps in the triggered frequency sweep.

Query example

`FREQ:STEP?` *Returns the number of steps in the frequency sweep.*

Error messages

- If the parameter is set <**0** or >**250**, error –222, “Data out of range” occurs.
- If **TRIGger:SOURce** is not set to **EXT**, error –221, “Settings conflict” occurs.

[SENSe[1]:]FREQuency[:CW|:FIXed]:STOP <numeric_value> <unit>

This command sets the stop frequency of the frequency sweep. It must be used in conjunction with an external trigger.

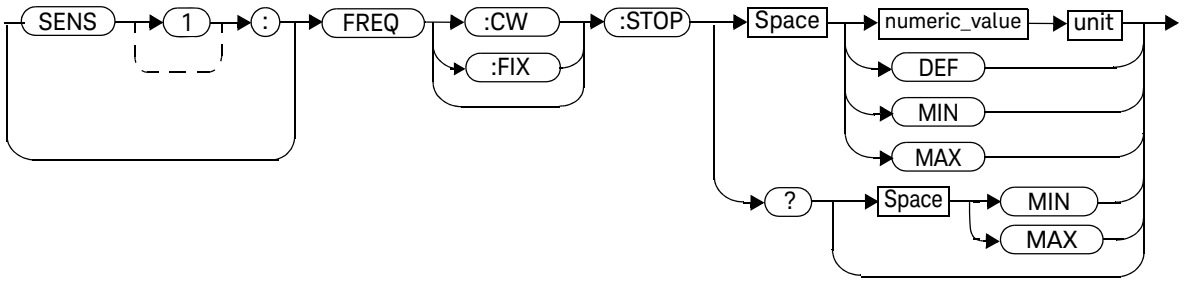
If the frequency sweep is disabled (frequency sweep step set to 0), the stop frequency will be set but it will not take effect.

`[SENSe[1]:]FREQuency[:CW|:FIXed]:STARt`,
`[SENSe[1]:]FREQuency[:CW|:FIXed]:STOP`, and
`[SENSe[1]:]FREQuency[:CW|:FIXed]:STEP` may be set in any sequence.

When the frequency sweep mode is configured with a frequency step size within the allowable range, the following conditions apply:

- If the stop frequency point is greater than the frequency start point, the frequency range will be swept in an ascending order.
- If the stop frequency point is less than the frequency start point, the frequency range will be swept in a descending order.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value for the stop frequency: – **DEF**: the default value is 50 MHz – **MIN**: 0 Hz – **MAX**: 1000.0 GHz The default unit is Hz.	0 Hz to 1000.0 GHz[a] **DEF** **MIN** **MAX**

[a] The following measurement units can be used:

- Hz
- kHz (10^3)
- MHz (10^6)
- GHz (10^9)

Example

```
FREQ:STOP 10MHz
```
Sets the frequency sweep to stop at 10 MHz.

Reset condition

On reset, the value is set to 50 MHz.

Query

`[SENSe[1]:]FREQuency[:CW|:FIXed]:STOP?`

This query retrieves the stop frequency of the triggered frequency sweep. Frequency is returned in Hz.

Query example

`FREQ:STOP?` *Returns the stop frequency of the frequency sweep in Hz.*

Error message

If the parameter set is <0 Hz or >1000 GHz, error –222, “Data out of range” occurs.

[SENSe[1]:]MRATe <character_data>

This command sets the measurement rate on the U8480 Series.

The U8480 Series can operate in three different measurement rates as shown in the following table.

Measurement rate	Measurement speed	Minimum acquisition time[a]
Normal	20 readings/s	51.2 ms/reading
Double	40 readings/s	24.576 ms/reading
Fast[b]	400 readings/s	2.048 ms/reading

[a] Minimum acquisition time is the minimum time for the U8480 Series to complete a measurement cycle. The time includes integration time and settling time.

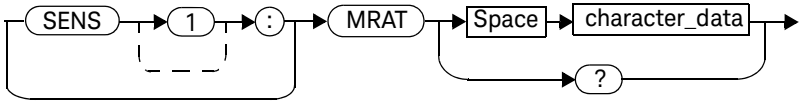
[b] In the Fast mode, the minimum acquisition time is the same as integration time. Integration time is defined as the period during which the analog-to-digital (A/D) converter of the U8480 Series samples the input signal for a measurement.

When the measurement rate is set to Fast, the following couplings occur:

Command	Status
`[SENSe[1]:]AVERage:STATe`	`OFF`[a]
`[SENSe[1]:]CORRection:DCYCle:STATe`	`OFF`[a]
`[SENSe[1]:]CORRection:GAIN2:STATe`	`OFF`[a]
`CALCulate[1]:MATH:EXPRession`	`"(SENS1)"`

[a] This change only occurs on the measurement specified in the `[SENSe[1]:]MRATe` command. When the measurement rate is changed from Fast to either Double or Normal, the settings that were in place when Fast was entered are restored.

Syntax



Parameters

Item	Description/Default	Range of values
character_data	A numeric value for the measurement : – NORMal: 20 readings/s – DOUBle: 40 readings/s – FAST: up to 400 readings/s[a] – The default is NORMal.	NORMal[a] [d] DOUBle[b] [d] FAST[c] [d]

[a] The measurement is taken with SENS:AVER:STAT set to OFF.

[b] When NORMal or DOUBle is set, TRIGger[:SEQuence[1]]:COUNt is set automatically to 1.

[c] To reduce the sensor-dependent delay time, use the measurement buffer by setting TRIGger[:SEQuence[1]]:COUNt higher than 1.

[d] Calibration of the U8480 Series is recommended after measurement rate is changed.

Example

```
MRAT DOUB
```

This command sets the measurement rate to 40 readings/s.

Reset condition

On reset, the measurement rate is set to NORMal.

Query

`[SENSe[1]:]MRATe?`

The query returns the current setting of either **NORMal**, **DOUBle**, or **FAST**.

Query example

MRAT? *Queries the current measurement rate setting.*

Error message

If **<character_data>** is not set to **NORMal**, **DOUBle**, or **FAST**, error –224, “Illegal parameter value” occurs.

[SENSe[1]:]SPEed <numeric_value>

This command sets the measurement speed.

The U8480 Series can operate in three different measurement speed as shown in the following table.

Measurement speed	Minimum acquisition time[a]
20 readings/s	51.2 ms/reading
40 readings/s	24.576 ms/reading
400 readings/s[b]	2.048 ms/reading

[a] Minimum acquisition time is the minimum time for the U8480 Series to complete a measurement cycle. The time includes integration time and settling time.

[b] For the 400 readings/s measurement speed, the minimum acquisition time is the same as integration time. Integration time is defined as the period during which the analog-to-digital (A/D) converter of the U8480 Series samples the input signal for a measurement.

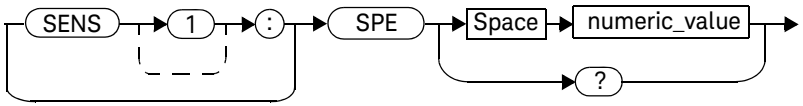
When the speed is set to 400 readings/s, the following couplings occur:

Command	Status
`[SENSe[1]:]AVERage:STATe`	`OFF`[a]
`[SENSe[1]:]CORRection:DCYCle:STATe`	`OFF`[a]
`[SENSe[1]:]CORRection:GAIN2:STATe`	`OFF`[a]
`CALCulate[1]:MATH:EXPRession`	`"(SENS1)"`

[a] This change only occurs on the measurement specified in the `[SENSe[1]:]SPEed` command. When the speed is changed from 400 readings/s to either 20 or 40 readings/s, the settings that were in place when 400 readings/s was entered are restored.

NOTE This command is included for compatibility purpose only. It has the same purpose as the `[SENSe[1]:]MRATe <NORMal|DOUBle|FAST>` command (with 20 readings/s referring to `NORMal`, 40 readings/s to `DOUBle`, and up to 400 readings/s to `FAST`).

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	A numeric value for the speed in readings/s. The default is 20 readings/s.	20 readings/s 40 readings/s Up to 400 readings/s

Example

```
SPE 40
```
This command sets the speed to 40 readings/s.

Reset condition

On reset, the speed is set to 20 readings/s.

Query

`[SENSe[1]:]SPEed?`

The query returns the current speed setting of either **20**, **40**, or a value of up to **400**.

Query example

`SPE?` *Queries the current speed setting.*

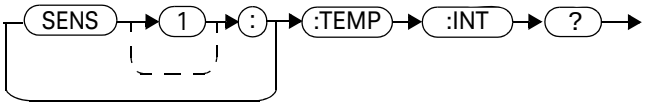
Error message

If **<numeric_value>** is not set to **20**, **40**, or up to **400**, error –224, “Illegal parameter value” occurs.

[SENSe[1]:]TEMPerature:INTernal?

This query returns the internal temperature of the U8480 Series, such as the board temperature, in degrees Celsius.

Syntax



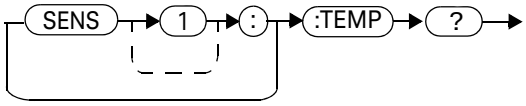
Example

`TEMP:INT?` *This query returns the U8480 Series internal temperature.*

[SENSe[1]:]TEMPerature?

This query returns the U8480 Series temperature in degrees Celsius.

Syntax



Example

`TEMP?` *This query returns the current U8480 Series temperature.*

9 SERVice Subsystem

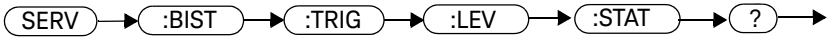
This chapter explains how the **SERVice** command subsystem is used to obtain and set information useful for servicing the U8480 Series.

SERVice:BIST:TRIGger:LEVel:STATe?

This query returns the trigger level.

- 1 is returned when the external trigger input is high.
- 0 is returned when the external trigger input is low.

Syntax



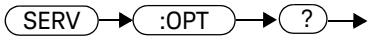
Example

```
SERV:BIST:TRIG:LEV:STAT?
```
Queries the trigger level.

SERVice:OPTion?

This query is used to determine the option of your U8480 Series.

Syntax



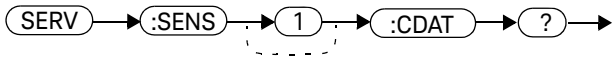
Example

`SERV:OPT?` *Returns the current option string.*

SERVice:SENSor[1]:CDATe?

This query returns the calibration date of the U8480 Series. The calibration date information is stored in the non-volatile memory.

Syntax



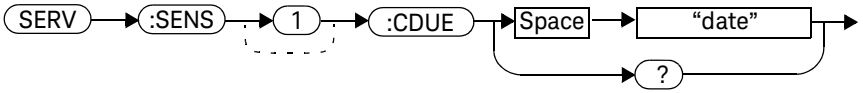
Example

`SERV:SENS:CDAT?` *Returns the calibration date of the U8480 Series.*

SERVice:SENSor[1]:CDUEdate <“date”>

This command is used to enter the calibration due date of the U8480 Series.

Syntax



Example

`SERV:SENS:CDUE “2012,09,21”` *This command enters the calibration due date as 21st September 2012.*

Query

`SERVice:SENSor[1]:CDUEdate?`

This query returns the calibration due date of the U8480 Series. The calibration due date information is stored in the non-volatile memory.

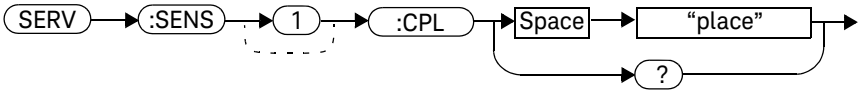
Query example

`SERV:SENS:CDUE?` *Returns the calibration due date of the U8480 Series.*

SERVice:SENSor[1]:CPLace <“place”>

This command is used to enter the place of calibration of the U8480 Series. A maximum of eight alphanumeric characters can be entered.

Syntax



Example

`SERV:SENS:CPL “Keys-Pen”` *This command enters the place of calibration as Keysight Penang.*

Query

`SERVice:SENSor[1]:CPLace?`

This query returns a string specifying the place of calibration of the U8480 Series, as stored in the non-volatile memory.

Query example

`SERV:SENS:CPL?` *Returns the U8480 Series place of calibration.*

NOTE If the place of calibration is not pre-programmed, “NONE” is returned.

SERVice:SENSor[1]:FREQuency:MAXimum?

This query returns the maximum frequency that can be measured by the U8480 Series, in MHz. The maximum frequency information is stored in the non-volatile memory.

Syntax

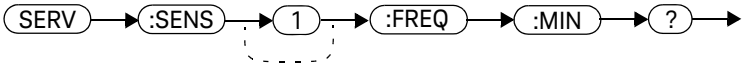
Example

`SERV:SENS:FREQ:MAX?`	*Returns the maximum frequency that can be measured by the U8480 Series.*

SERVice:SENSor[1]:FREQuency:MINimum?

This query returns the minimum frequency that can be measured by the U8480 Series, in MHz. The minimum frequency information is stored in the non-volatile memory.

Syntax



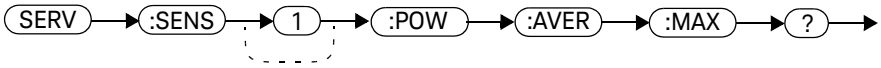
Example

`SERV:SENS:FREQ:MIN?`	*Returns the minimum frequency that can be measured by the U8480 Series.*

SERVice:SENSor[1]:POWer:AVERage:MAXimum?

This query returns the maximum average power that can be measured by the U8480 Series, in dBm. The maximum average power information is stored in the non-volatile memory.

Syntax



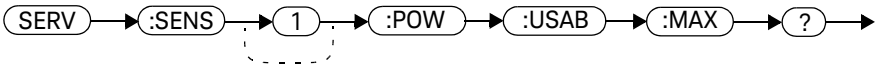
Example

`SERV:SENS:POW:AVER:MAX?` *Returns the maximum average power that can be measured by the U8480 Series.*

SERVice:SENSor[1]:POWer:USABle:MAXimum?

This query returns the maximum power that can be accurately measured by the U8480 Series, in dBm. The maximum power information is stored in the non-volatile memory.

Syntax



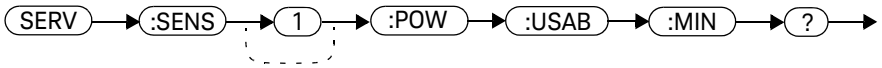
Example

`SERV:SENS:POW:USAB:MAX?` *Returns the maximum power that can be accurately measured by the U8480 Series.*

SERVice:SENSor[1]:POWer:USABle:MINimum?

This query returns the minimum power that can be accurately measured by the U8480 Series, in dBm. The minimum power information is stored in the non-volatile memory.

Syntax



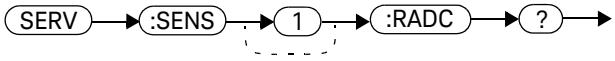
Example

`SERV:SENS:POW:USAB:MIN?` *Returns the minimum power that can be accurately measured by the U8480 Series.*

SERVice:SENSor[1]:RADC?

This query returns a new raw uncorrected measurement in volts, as a 32-bit signed integer.

Syntax



Example

```
SERV:SENS:RADC?
```
Returns a new raw uncorrected measurement.

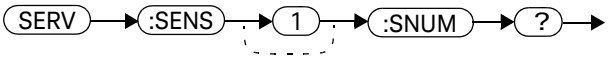
NOTE The raw uncorrected measurement is returned as a floating value.

SERVice:SENSor[1]:SNUMber?

This query is used to acquire the serial number of the U8480 Series in the form of MY12345678.

NOTE **This query performs the same function as the `SERVice:SNUMber?` query.**

Syntax



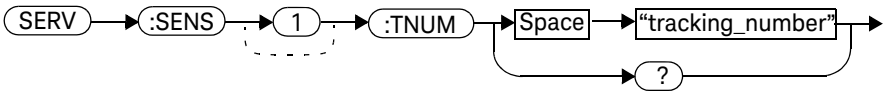
Example

`SERV:SENS:SNUM?` *Returns the U8480 Series serial number in the form of MY12345678.*

SERVice:SENSor[1]:TNUMber <“tracking_number”>

This command is used to enter the tracking number for the U8480 Series.

Syntax



Example

```
SERV:SENS:TNUM “PEN12345”
```

This command enters the tracking number of PEN12345.

Query

SERVice:SENSor[1]:TNUMber?

This query returns the tracking number of the U8480 Series. The tracking number information is stored in the non-volatile memory.

Query example

```
SERV:SENS:TNUM?
```

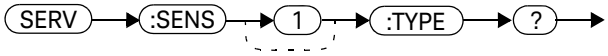
Returns the tracking number of the U8480 Series.

NOTE If the tracking number is not pre-programmed, “NONE” is returned.

SERVice:SENSor[1]:TYPE?

This query identifies the sensor type.

Syntax



Example

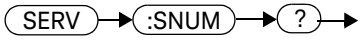
`SERV:SENS:TYPE?` *Returns the model name of the connected sensor.*

SERVice:SNUMber?

This query returns the U8480 Series serial number in the form of MY12345678.

NOTE This query performs the same function as the `SERVice:SENSor[1]:SNUMber?` query.

Syntax



Example

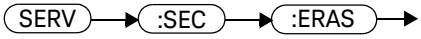
`SERV:SNUM?` *Returns the U8480 Series serial number in the form of MY12345678.*

SERVice:SECure:ERASe

This command sanitizes the U8480 Series memory, for example, before you return it to Keysight for repair or calibration, of all data stored in it.

The memory data sanitized include the save/recall states, FDO tables, Gamma tables and the S-Parameter table.

Syntax



Example

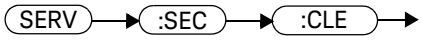
`SERV:SEC:ERAS` *Sanitizes the U8480 Series memory.*

SERVice:SECure:CLEar

This command clears the U8480 Series memory.

The memory data cleared includes the save/recall states, FDO tables, Gamma Tables, and the S-Parameter table.

Syntax



Example

`SERV:SEC:CLE` *Clears the U8480 Series memory.*

10 STATus Subsystem

This chapter explains how the **STATus** command subsystem enables you to examine the status of the U8480 Series by monitoring the Device Status Register, Operation Status Register, and Questionable Status Register.

STATus Command Subsystem

The **STATus** command subsystem enables you to examine the status of the U8480 Series by monitoring the following status registers:

- Device Status Register
- Operation Status Register
- Questionable Status Register

The contents of these and other registers in the U8480 Series are determined by one or more status registers.

The following table summarizes the effects of various commands and events on these status registers:

Table 10-1 Commands and events affecting status registers

Status register	*RST	*CLS	Power on	STATus:PRESet
SCPI transition filters (NTR and PTR registers)	none	none	preset	preset
SCPI enable registers	none	none	preset	preset
SCPI event registers	none	clear	clear	none
SCPI error/event queue enable	none	none	preset	preset
SCPI error/event queue	none	clear	clear	none
IEEE-488.2 registers ESE SRE	none	none	clear	none
IEEE-488.2 registers ESR STB	none	clear	clear	none

The contents of the status registers are examined using the following status register set commands:

```
:CONDition?
:ENABle <NRf>|<non-decimal numeric>
[:EVENt?]
:NTRansition <NRf>|<non-decimal numeric>
:PTRansition <NRf>|<non-decimal numeric>
```

Each of these can be used to examine any of the following status registers:

`STATus:DEVice` (page 266)

`STATus:OPERation` (page 268)

`STATus:OPERation:CALibrating[:SUMMary]` (page 269)

`STATus:OPERation:LLFail[:SUMMary]` (page 270)

`STATus:OPERation:MEASuring[:SUMMary]` (page 271)

`STATus:OPERation:SENSe[:SUMMary]` (page 272)

`STATus:OPERation:TRIGger[:SUMMary]` (page 273)

`STATus:OPERation:ULFail[:SUMMary]` (page 274)

`STATus:QUEStionable` (page 277)

`STATus:QUEStionable:CALibration[:SUMMary]` (page 278)

`STATus:QUEStionable:POWer[:SUMMary]` (page 279)

Examples

- To use the `:CONDition?` command to examine the `STATus:DEVice` register:

  ```
  STATus:DEVice:CONDition?
  ```

- To use the `:NTRansition` command to examine the `STATus:OPERation:SENSe[:SUMMary]` register:

  ```
  STATus:OPERation:SENSe[:SUMMary]:NTRansition
  ```

Status Register Set Commands

This section describes the five status register set commands. Each can be used to examine all of the status registers listed on page 261.

To apply a command to a specific register, prefix the command with the name of the appropriate register. For example, to apply the **:ENABle** command to the **STATus:QUEStionable** register, use the following command:

```
STATus:QUEStionable:ENABle
```

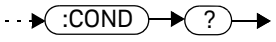
The status register set commands detailed in this section are as follows:

Keyword	Parameter form	Notes	Page
:CONDition?		[query only]	page 262
:ENABle	<NRf>|<non-decimal numeric>		page 263
[:EVENt?]		[query only]	page 263
:NTRansition	<NRf>|<non-decimal numeric>		page 264
:PTRansition	<NRf>|<non-decimal numeric>		page 265

:CONDition?

This query returns a 16-bit decimal-weighted number representing the bits set in the condition register of the SCPI register set you require to control. The format of the return is **<NR1>** in the range of 0 to 32767 (2^{15}–1). The contents of the condition register remain unchanged after it is read.

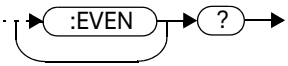
Syntax



[:EVENt]?

This query returns a 16-bit decimal-weighted number representing the bits set in the event register of the SCPI register set you require to control. The format of the return is **<NR1>** in the range of 0 to 32767 (2^{15}–1). This query clears all bits in the register to 0.

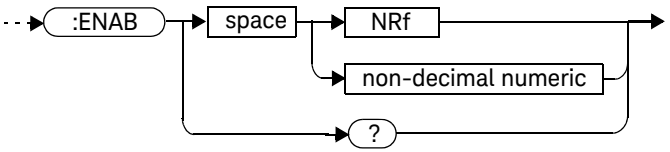
Syntax



:ENABle <NRf>|<non-decimal numeric>

This command sets the enable register of the particular SCPI register set you require to control. The parameter value, when rounded to an integer and expressed in base 2, has its first 15 bits written into the enable register of the SCPI register set concerned. The last bit (bit 15) is always set to 0.

Syntax



Parameters

Type	Description	Range of values
NRf non-decimal numeric	The value used to set the enable register	0 to 2^{15}–1

Query

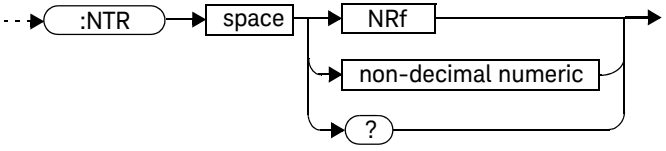
```
:ENABle?
```

The query returns a 15-bit decimal-weighted number representing the contents of the enable register of the SCPI register set being queried. The format of the return is **<NR1>** in the range of 0 to 32767 (2^{15}–1).

:NTRansition <NRf>|<non-decimal numeric>

This command sets the negative transition register of the SCPI register set you require to control. The parameter value, when rounded to an integer and expressed in base 2, has its first 15 bits written into the negative transition register of the SCPI register set concerned. The last bit (bit 15) is always set to 0.

Syntax



Parameters

Type	Description	Range of values
NRf non-decimal numeric	The value used to set the NTR register.	0 to 2^{15}–1

Query

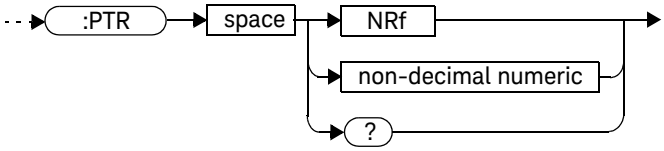
```
:NTRansition?
```

The query returns a 15-bit decimal-weighted number representing the contents of the negative transition register of the SCPI register set being queried. The format of the return is **<NR1>** in the range of 0 to 32767 (2^{15}–1).

:PTRansition <NRf>|<non-decimal numeric>

This command is used to set the positive transition register of the SCPI register set you require to control. The first 15 bits of the input parameter are written into the positive transition register of the SCPI register set concerned. The last bit (bit 15) is always set to 0.

Syntax



Parameters

Type	Description	Range of values
NRf non-decimal numeric	The value used to set the PTR register.	0 to $2^{15}-1$

Query

`:PTRansition?`

The query returns a 15-bit decimal-weighted number representing the contents of the positive transition register of the SCPI register set being queried. The format of the return is **<NR1>** in the range of 0 to 32767 ($2^{15}-1$).

Device Status Register Sets

The status registers contain information which gives device status information. The contents of the individual registers of these register sets may be accessed by appending the commands listed in "Status Register Set Commands" on page 262.

The following command descriptions detail the SCPI register you require to control but do not detail the register set commands.

The one device status register set is:

`STATus:DEVice`

The following bit in these registers is used by the U8480 Series:

Bit number	Decimal weight	Definition
0 to 2	-	Not used
3	8	U8480 Series error
4 to 15	-	Not used (bit 15 is always 0)

The U8480 Series error bit (3) is set to:

- 1, if the U8480 Series non-volatile memory has failed.
- 0, for every other condition.

Operation Register Sets

The following registers contain information which is part of the U8480 Series normal operation. The contents of the individual registers of these register sets may be accessed by appending the commands listed in "Status Register Set Commands" on page 262.

The following command descriptions detail the SCPI register you require to control but do not detail the register set commands.

The seven operation register sets are:

```
STATUS:OPERation
STATus:OPERation:CALibrating[:SUMMary]
STATus:OPERation:LLFail[:SUMMary]
STATus:OPERation:MEASuring[:SUMMary]
STATus:OPERation:SENSe[:SUMMary]
STATus:OPERation:TRIGger[:SUMMary]
STATus:OPERation:ULFail[:SUMMary]
```

Further information on these register sets is provided on the following pages.

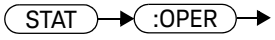
STATus:OPERation

The operation status register set contains conditions which are part of the U8480 Series operation as a whole.

The following bits in these registers are used by the U8480 Series:

Bit number	Decimal weight	Definition
0	1	CALibrating summary
1 to 3	-	Not used
4	16	MEASuring summary
5	32	Waiting for TRIGger summary
6 to 9	-	Not used
10	1024	SENSe summary
11	2048	Lower limit fail summary
12	4096	Upper limit fail summary
13 to 15	-	Not used (bit 15 is always 0)

Syntax



STATus:OPERation:CALibrating[:SUMMary]

The operation status calibrating summary register set contains information on the calibrating status of the U8480 Series.

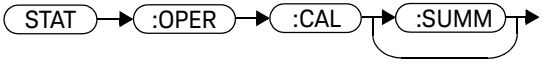
The following bit in these registers is used by the U8480 Series:

Bit number	Decimal weight	Definition
0	-	Not used
1	2	CALibrating status
2 to 15	-	Not used (bit 15 is always 0)

Bit 1 is set at the beginning of zeroing (**CALibration:ZERO:AUTO ONCE**) or calibration (**CALibration:AUTO ONCE**). Also for the compound command/query **CALibration[:ALL]?**, this bit is set at the beginning of the calibration sequence.

This bit is cleared at the end of zeroing or calibration.

Syntax



STATus:OPERation:LLFail[:SUMMary]

The operation status lower limit fail summary register set contains information on the lower limit fail status of the U8480 Series.

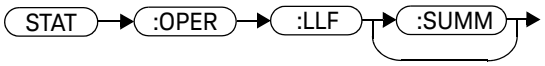
The following bit in these registers is used by the U8480 Series:

Bit number	Decimal weight	Definition
0	-	Not used
1	2	LLFail status
2 to 15	-	Not used (bit 15 is always 0)

Bit 1 is set if the lower limit test fails.

This bit is cleared if a measurement is made and the test is enabled and passes.

Syntax



STATus:OPERation:MEASuring[:SUMMary]

The operation status measuring summary register set contains information on the measuring status of the U8480 Series.

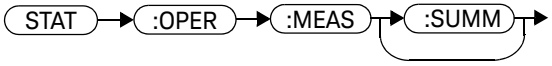
The following bit in these registers is used by the U8480 Series:

Bit number	Decimal weight	Definition
0	-	Not used
1	2	MEASuring status
2 to 15	-	Not used (bit 15 is always 0)

Bit 1 is set when the U8480 Series is taking a measurement.

This bit is cleared when the measurement has completed.

Syntax



STATus:OPERation:SENSe[:SUMMary]

The operation status sense summary register set contains information on the status of the U8480 Series.

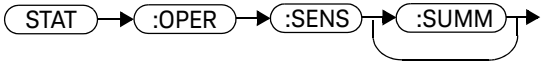
The following bit in these registers is used by the U8480 Series:

Bit number	Decimal weight	Definition
0	-	Not used
1	2	SENSe status
2 to 15	-	Not used (bit 15 is always 0)

Bit 1 is set when the U8480 Series is reading data from the non-volatile memory.

This bit is cleared when the U8480 Series is not reading data from the non-volatile memory.

Syntax



STATus:OPERation:TRIGger[:SUMMary]

The operation status trigger summary register set contains information on the trigger status of the U8480 Series.

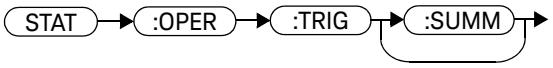
The following bit in these registers is used by the U8480 Series:

Bit number	Decimal weight	Definition
0	-	Not used
1	2	TRIGger status
2 to 15	-	Not used (bit 15 is always 0)

Bit 1 is set when the U8480 Series enters the “wait for trigger” state.

This bit is cleared when the U8480 Series enters the “idle” state.

Syntax



STATus:OPERation:ULFail[:SUMMary]

The operation status upper limit fail summary register set contains information on the upper limit fail status of the U8480 Series.

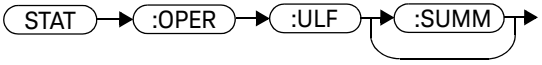
The following bit in these registers is used by the U8480 Series:

Bit number	Decimal weight	Definition
0	-	Not used
1	2	ULFail status
2 to 15	-	Not used (bit 15 is always 0)

Bit 1 is set if the upper limit test fails.

This bit is cleared if a measurement is made and the test is enabled and passes.

Syntax



STATus:PRESet

PRESet sets a number of the status registers to their preset values as shown below – all other registers are unaffected. Bit 15 is always 0.

Syntax

Register	Filter/Enable	PRESet value
OPERation	ENABle	all zeros
	PTR	all ones
	NTR	all zeros
QUEStionable	ENABle	all zeros
	PTR	all ones
	NTR	all zeros
All others	ENABle	all ones
	PTR	all ones
	NTR	all zeros

Questionable Register Sets

The questionable register sets contain information which gives an indication of the quality of the data produced by the U8480 Series. The contents of the individual registers in these register sets may be accessed by appending the commands listed in "Status Register Set Commands" on page 262.

The following command descriptions detail the SCPI register you require to control but do not detail the register set commands.

The three questionable register sets are:

```
STATus:QUEStionable
```

```
STATus:QUEStionable:CALibration[:SUMMary]
```

```
STATus:QUEStionable:POWer[:SUMMary]
```

STATus:QUEStionable

The questionable register set contains bits that indicate the quality of various aspects of signals processed by the U8480 Series.

The following bits in these registers are used by the U8480 Series:

Bit number	Decimal weight	Definition
0 to 2	-	Not used
3	8	POWer summary
4 to 7	-	Not used
8	256	CALibration summary
9	512	Power-on self-test
10 to 15	-	Not used (bit 15 is always 0)

Bit 3 is set by the logical OR outputs of the `STATus:QUEStionable:POWer:SUMMary` register set.

Bit 8 is set by the logical OR outputs of the `STATus:QUEStionable:CALibration:SUMMary` register set.

Bit 9 is set if the power-on self-test fails, and cleared if it passes.

Syntax

STATus:QUEStionable:CALibration[:SUMMary]

The questionable calibration summary register set contains bits which give an indication of the quality of the data produced by the U8480 Series due to its calibration status.

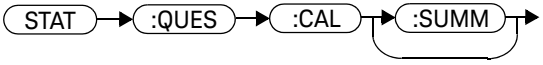
The following bit in these registers is used by the U8480 Series:

Bit number	Decimal weight	Definition
0	-	Not used
1	2	Summary of CALibration
2 to 15	-	Not used (bit 15 is always 0)

Bit 1 is set when:

- `CALibration[1]:ZERO:AUTO ONCE, CALibration[1]:AUTO ONCE`, `CALibration[1][:ALL], or CALibration[1][:ALL]?` is executed
- error –231, “Data questionable;ZERO ERROR” or –231, “Data questionable;CAL ERROR” occurs
- This bit is cleared when any of the commands listed above succeed and no errors are placed on the error queue.

Syntax



STATus:QUEStionable:POWer[:SUMMary]

The questionable power summary register set contain bits that indicate the quality of the power data being acquired by the U8480 Series.

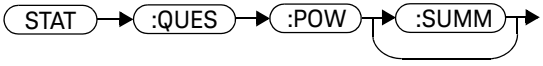
The following bit in these registers is used by the U8480 Series:

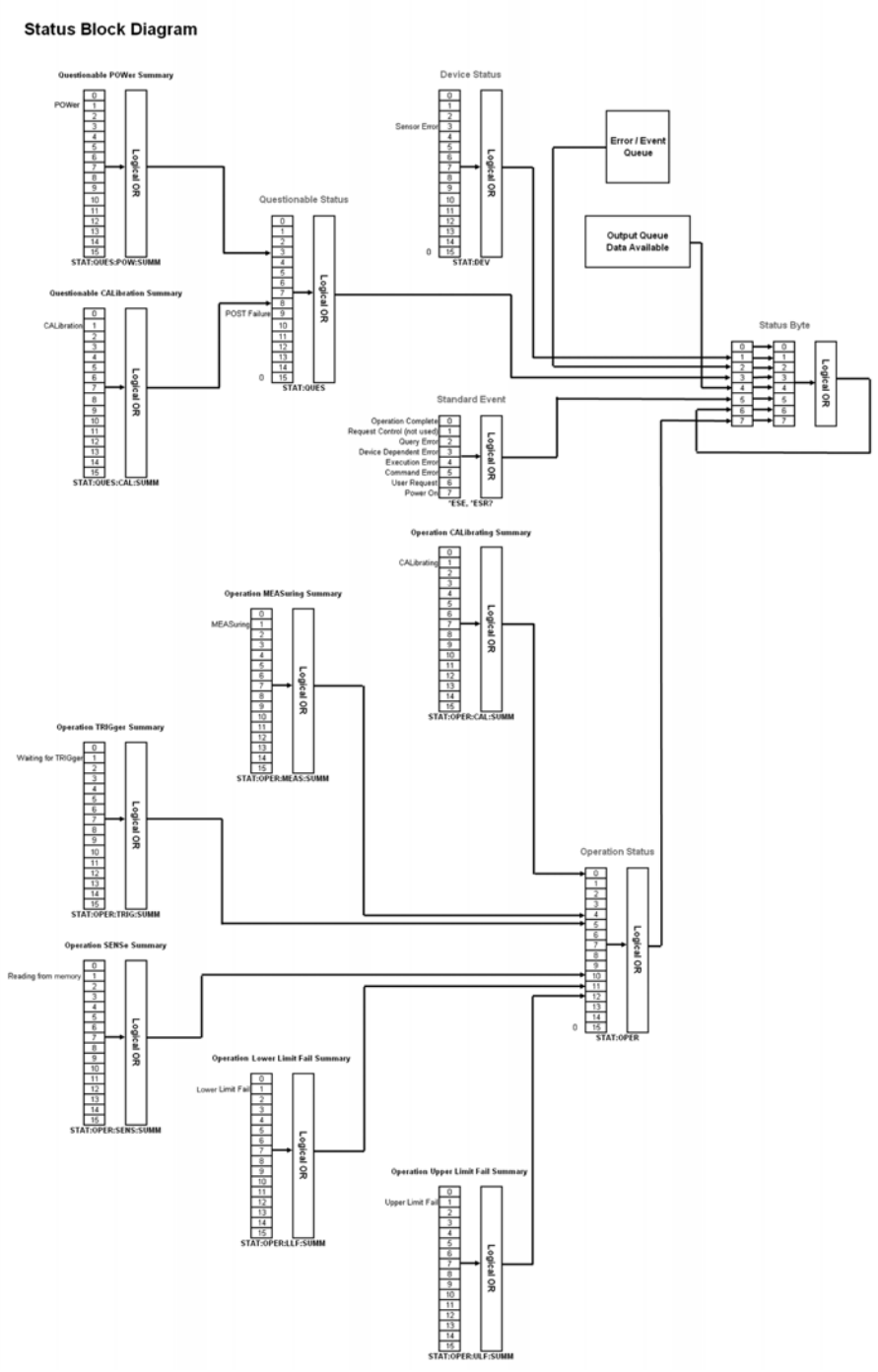
Bit number	Decimal weight	Definition
0	-	Not used
1	2	Summary of POWer
2 to 15	-	Not used (bit 15 is always 0)

Bit 1 is set when error –230, “Data corrupt or stale”, error –231, “Data questionable;Input Overload”, or error –231, “Data questionable;ZERO ERROR” occurs.

This bit is cleared when no errors or events are detected by the U8480 Series during a measurement covering the causes given for it to set.

Syntax




Status Block Diagram
Questionable POWer Summary
POWer
Logical OR
STAT:QUES:POW:SUMM
Device Status
Sensor Error
Logical OR
STAT:DEV
Error / Event Queue
Output Queue Data Available
Questionable Status
Logical OR
STAT:QUES
Questionable CALibration Summary
CALibration
POST Failure
Logical OR
STAT:QUES:CAL:SUMM
Status Byte
Logical OR
Standard Event
Operation Complete
Request Control (not used)
Query Error
Device Dependent Error
Execution Error
Command Error
User Request
Power On
Logical OR
*ESE, *ESR?
Operation CALibrating Summary
CALibrating
Logical OR
STAT:OPER:CAL:SUMM
Operation MEASuring Summary
MEASuring
Logical OR
STAT:OPER:MEAS:SUMM
Operation TRIGger Summary
Waiting for TRIGger
Logical OR
STAT:OPER:TRIG:SUMM
Operation Status
Logical OR
STAT:OPER
Operation SENSe Summary
Reading from memory
Logical OR
STAT:OPER:SENS:SUMM
Operation Lower Limit Fail Summary
Lower Limit Fail
Logical OR
STAT:OPER:LLF:SUMM
Operation Upper Limit Fail Summary
Upper Limit Fail
Logical OR
STAT:OPER:ULF:SUMM

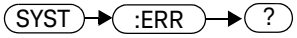
11 SYSTem Subsystem

This chapter explains how to use the **SYSTem** command subsystem to return error numbers and messages from the U8480 Series, preset the U8480 Series, and query the SCPI version.

SYSTem:ERRor?

This query returns error numbers and messages from the U8480 Series error queue. When an error is generated by the U8480 Series, it stores an error number and corresponding message in the error queue. One error is removed from the error queue each time this query is executed. The errors are cleared in the order of first-in first-out, which means that the oldest errors are cleared out first. To clear all the errors from the error queue, execute the **`*CLS`** command. When the error queue is empty, subsequent **`SYSTem:ERRor?`** queries return a +0, "No error" message. The error queue has a maximum capacity of 50 errors.

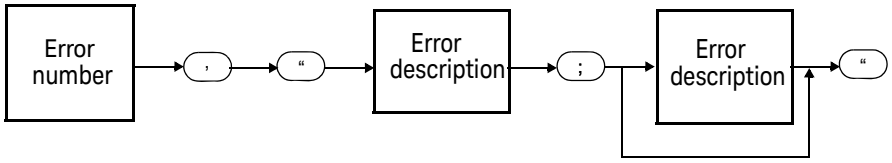
Syntax



Example

`SYST:ERR?` *Queries the system error.*

Error queue messages have the following format:



For example, –330, "Self-test Failed;ROM Check Failed"

Errors are retrieved in a first-in first-out (FIFO) order. If more than 50 errors occur, the error queue overflows and the last error in the queue is replaced with the error –350, "Queue overflow". Each time the queue overflows, the most recent error is discarded.

Example

`SYST:ERR?`	*Queries the oldest error message stored in the U8480 Series error queue.*

Reset condition

On reset, the error queue is unaffected.

Error messages

If the error queue overflows, the last error is replaced with –350, “Queue overflow”. No additional errors are accepted by the queue until space becomes available.

Error message list

–101	Invalid character An invalid character was found in the command string. You may have inserted a character such as #, $, or % in the command header or within a parameter. For example, `CALC:LIM:LOW 0#`.
–102	Syntax error Invalid syntax was found in the command string. For example, `CALC:LIM:CLE:AUTO 1` or `CALC:LIM:CLE: AUTO 1`.
–103	Invalid separator An invalid separator was found in the command string. You may have used a comma instead of a colon, semicolon, or blank space; or you may have used a blank space instead of a comma. For example, `CALC:LIM:CLE:AUTO,1`.
–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 for the command. You may have entered an extra parameter, or added a parameter to a command that does not accept a parameter. For example, `CAL 10`.

–109	Missing parameter Fewer parameters were received than expected for the command. You omitted one or more parameters that are required for this command. For example, `AVER:COUN`.
–112	Program mnemonic too long A command header was received which contained more than the maximum 12 characters allowed. For example, `SENSe:AVERage:COUNt ABCDEFGHIJKLMN`.
–113	Undefined header A command was received that is not valid for the U8480 Series. You may have misspelled the command, it may not be a valid command, or you may have the wrong interface selected. If you are using the short form of the command, remember that it may contain up to four letters. For example, `TRIG:SOUR0 IMM`.
–121	Invalid character in number An invalid character was found in the number specified for a parameter value. For example, `SENS:AVER:COUN 128`#H.
–123	Exponent too large A numeric parameter was found whose exponent was larger than 32000. For example, `SENS:AVER:COUN 1E34000`.
–124	Too many digits A numeric parameter was found whose mantissa contained more than 255 digits, excluding leading zeros.
–128	Numeric data not allowed A numeric value was received within a command which does not accept a numeric value. For example, `MEM:CLE 24`.
–131	Invalid suffix A unit was incorrectly specified for a numeric parameter. You may have misspelled the unit. For example, `SENS:FREQ 200MZ`.
–134	Suffix too long A unit used contained more than 12 characters. For example, `SENS:FREQ 20MHZZZZZZZZZZZZ`.
–138	Suffix not allowed A unit was received following a numeric parameter which does not accept a unit. For example, `INIT:CONT 0Hz`.

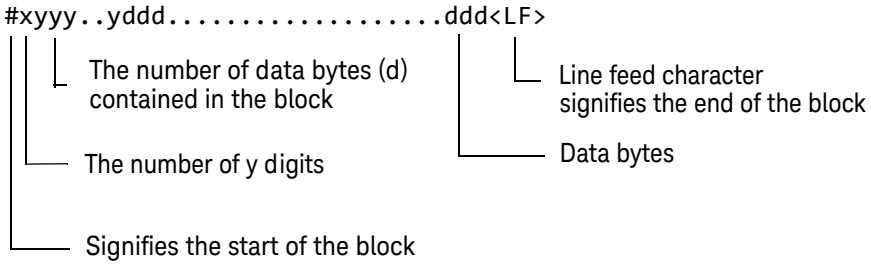
–148	Character data not allowed A discrete parameter was received but a character string or a numeric parameter was expected. Check the list of parameters to verify that you have used a valid parameter type. For example, `MEM:CLE CUSTOM_1`.
–151	Invalid string data An invalid string was received. Check to see if you have enclosed the character string in single or double quotes. For example, `MEM:CLE “CUSTOM_1`.
–158	String data not allowed A character string was received but is not allowed for the command. Check the list of parameters to verify that you have used a valid parameter type. For example, `AVER:STAT 'ON'`.
–161	Invalid block data A block data element was expected but was invalid for some reason. For example, `*DDT #15FET`. The 5 in the string indicates that 5 characters should follow, whereas in this example there are only 3.
–168	Block data not allowed A legal block data element was encountered but not allowed by the U8480 Series at this point. For example, `AVER #0` or `SENS:AVER:STAT #0`.
–178	Expression data not allowed A legal expression data was encountered but not allowed by the U8480 Series at this point. For example, `SENS:AVER:COUN (32+2)`.
–211	Trigger ignored Indicates that `*TRG` or `TRIG:IMM` was received and recognized by the device but was ignored because the U8480 Series was not in the wait-for-trigger state.
–213	Init ignored Indicates that a request for a measurement initiation was ignored as the U8480 Series was already initiated. For example, `INIT:CONT ON INIT`.
–214	Trigger deadlock `TRIG:SOUR` was set to `HOLD` or `BUS`, and a `READ?` or `MEASure?` was attempted, expecting `TRIG:SOUR` to be set to `IMMediate`.

–220	Parameter error;Frequency list must be in ascending order. Indicates that the frequencies entered using the **MEMory:TABLe:FREQuency** command are not in ascending order.
–221	Settings conflict This message occurs under a variety of conflicting conditions. The following list gives a few examples of where this error may occur: – If the **READ?** parameters do not match the current settings. – If you are in the fast mode and attempting to switch on for example, averaging, duty cycle, or limits. – Trying to clear a frequency-dependent offset table when none is selected.
–222	Data out of range A numeric parameter value is outside the valid range for the command. For example, **SENS:FREQ 1HZ**.
–224	Illegal parameter value A discrete parameter was received which was not a valid choice for the command. You may have used an invalid parameter choice. For example, **CORR:CSET2:SEL "CUSTOM_wahaha_A"**.
–226	Lists not same length This occurs when **SENSe:CORRection:CSET2:STATe** is set to **ON** and the frequency and offset lists do not correspond in length.
–230	Data corrupt or stale This occurs when a **FETC?** is attempted and either a reset has been received or the U8480 Series state has changed such that the measurement is invalidated (for example, a change of frequency setting or triggering conditions).
–231	Data questionable;**CAL ERROR** The U8480 Series calibration failed.
–231	Data questionable;**ZERO ERROR** The U8480 Series zeroing failed. The most likely cause is attempting to zero when some power signal is being applied to the U8480 Series.
–231	Data questionable;Input Overload The power input to the U8480 Series exceeds the maximum range.
–310	System error;Sensor non-volatile memory Read Failed - critical data not found or unreadable This indicates a failure with the U8480 Series. Refer to the U8480 Series service manual for details on returning it for repair.

–321	Out of memory The U8480 Series required more memory than was available to run an internal operation.
–330	Self-test Failed; The –330,"Self-test Failed" errors indicate that you have a problem with the U8480 Series. Refer to the U8480 Series service manual for details of what to do with the faulty U8480 Series.
–330	Self-test Failed;Measurement Channel Fault
–330	Self-test Failed;ROM Check Failed
–330	Self-test Failed;RAM Check Failed
–350	Queue overflow The error queue is full and another error has occurred which could not be recorded.
–410	Query **INTERRUPTED** A command was received which sends data to the output buffer, but the output buffer contained data from a previous command (the previous data is not overwritten). The output buffer is cleared when power has been turned off, or after the ***RST** (reset) command has been executed.
–420	Query **UNTERMINATED** The U8480 Series was addressed to talk (that is, to send data over the interface) but a command has not been received which sends data to the output buffer. For example, you may have executed a **CONFigure** command (which does not generate data) and then attempted to read data from the remote interface.
–430	Query **DEADLOCKED** A command was received which generates too much data to fit in the output buffer and the input buffer is also full. Command execution continues but data is lost.
–440	Query **UNTERMINATED** after indefinite response The ***IDN?** command must be the last query command within a command string.

SYSTem:HELP:HEADers?

This query returns a list of all SCPI commands supported by the U8480 Series. Data is returned in the IEEE-488.2 arbitrary block program data format as shown in the figure below.



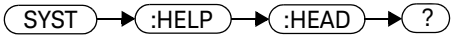
Example: if there are 12435 data bytes, y = 12435 and x = 5

Figure 11-1 IEEE 488.2 arbitrary block program data format

Each point in the trace is represented as an IEEE-754 32-bit floating point number, made up of four bytes in the data block. The MSB is transmitted first. Each complete block is terminated by a line feed.

Commands are listed in the alphabetical order.

Syntax



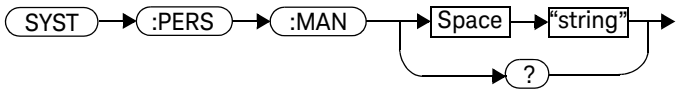
Example

`SYST:HELP:HEAD?`	*Returns the SCPI commands supported by the U8480 Series.*

SYSTem:PERSona:MANufacturer <"string">

This command only accepts two non-case sensitive strings “Agilent Technologies” and “Keysight Technologies”. A power cycle or reboot is required for the changes in the instrument’s manufacturer string to take effect and to be recognized in the Keysight Connection Expert. The string will remain for the subsequent power cycle or reboot.

Syntax



Example

`SYST:PERS:MAN "Agilent Technologies"`	*This command sets the instrument’s manufacturer to “Agilent Technologies”.*
`SYST:PERS:MAN "Keysight Technologies"`	*This command sets the instrument’s manufacturer to “Keysight Technologies”.*

Reset condition

On reset, the manufacturer string is not affected.

Query

SYSTem:PERSona:MANufacturer?

The query returns the manufacturer string that was set.

Query example

SYST:PERS:MAN? *Queries the manufacturer string that was set.*

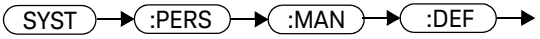
Error message

If <"string"> is not "Agilent Technologies" or "Keysight Technologies", error -158, "String data not allowed" occurs.

SYSTem:PERSona:MANufacturer:DEFault

This command sets the instrument's manufacturer to "Keysight Technologies" which is the default manufacturer setting.

Syntax



Example

```
SYST:PERS:MAN:DEF
```

This command sets the instrument's manufacturer to "Keysight Technologies" which is the default manufacturer setting.

Query

```
SYSTem:PERSona:MANufacturer:DEFault?
```

The query returns the default manufacturer string.

Query example

```
SYST:PERS:MAN:DEF?
```

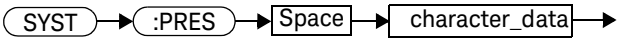
Queries the default manufacturer string.

SYSTem:PRESet <character_data>

This command presets the U8480 Series to the values appropriate for measuring the communications format specified by <character_data>. The U8480 Series is preset to the default values if no value or the value `DEFault` is supplied.

NOTE `DEFault` settings apply to both `*RST` and to `SYSTem:PRESet DEFault` unless stated otherwise.

Syntax



Parameters

Item	Description	Range of values
character_data	A communications format which determines the preset values.	`DEFault`

Example

```
SYST:PRES DEF
```

This command presets the U8480 Series to the default values. The same default values are set when the parameter is omitted.

Preset values

DEFault

Table 11-1 shows the U8480 Series presets when `<character_data>` is set to `DEFault` or omitted.

Table 11-1 DEFault: U8480 Series presets

Command	Setting	Comment
`CALC[1]:FEED[1]`	"POW:AVER"	Select average measurement type
`CALC[1]:LIM:CLE:AUTO`	ON	Clear limit data at `INIT`
`CALC[1]:LIM:LOW[:DATA]`	–90 dBm	Lower limit
`CALC[1]:LIM:STAT`	OFF	Limits checking disabled
`CALC[1]:LIM:UPP[:DATA]`	+90 dBm	Upper limit
`CALC[1]:MATH[:EXPR]`	"(SENS1)"	Math expression
`FORM[:READ]:BORD`	NORMal	Binary order
`FORM[:READ][:DATA]`	ASCii	Data format
`INIT[1]:CONT`	*RST: OFF SYST:PRES: ON	U8480 Series in idle state U8480 Series in wait-for-trigger state
`MEM:TABL:SEL`	not affected	Active frequency-dependent offset table
`[SENS[1]:]AVER:COUN`	4	Filter length
`[SENS[1]:]AVER:COUN:AUTO`	ON	Auto-filtering enabled
`[SENS[1]:]AVER:SDET`	1	Step detection enabled
`[SENS[1]:]AVER[:STAT]`	ON	Averaging enabled
`[SENS[1]:]CORR:CSET2[:SEL]`	not affected	Selected frequency-dependent offset table
`[SENS[1]:]CORR:CSET2:STAT`	not affected	Frequency-dependent offset table disabled
`[SENS[1]:]CORR:FDOF|GAIN4[:INP][:MAGN]`	not affected	Return frequency-dependent offset
`[SENS[1]:]CORR:GAIN2:STAT`	OFF	Channel offset disabled
`[SENS[1]:]CORR:GAIN2[:INP][:MAGN]`	0.0 dB	Enter channel offset value

Command	Setting	Comment
[SENS[1]:]FREQ[:CW|:FIX]	+50.000 MHz	Frequency setting
SENSe[1]:]CORRection:CSET3:STATe	OFF	Table-based Gamma Correction disabled
SENSe[1]:]CORRection:SGAMma:STATe	OFF	Single Point Gamma Correction disabled
SENSe[1]:]CORRection:CSET4:STATe	OFF	S-Parameters Correction disabled
SENSe[1]:]CORRection:SGAMma:MAGNitude	0.0	Single Point Gamma magnitude
SENSe[1]:]CORRection:SGAMma:PHASe	0.0	Single Point Gamma phase
SENSe[1]:]MUNC:STATe	OFF	Real-Time MU disabled
SENSe[1]:]MUNC:SGAMma:TYPE	SINGle	Gamma type for Real-Time MU
[SENS[1]:]MRAT	NORMal	Measurement speed
TRIG[1]:DEL:AUTO	ON	Enable settling time delay
TRIG[:SEQ]:SLOP	POSitive	Trigger event recognized on rising edge
TRIG[:SEQ[1]]:COUN	1	Trigger events for measurement cycle
TRIG[:SEQ[1]]:DEL:AUTO	ON	Enable settling time delay
TRIG[:SEQ[1]]:SOUR	IMMediate	Trigger source setup
UNIT:POW	dBm	Power units

SYSTem:VERSion?

This query returns the version of SCPI used in the U8480 Series. The response is in the form of XXXX.Y, where XXXX is the year and Y is the version number.

Syntax

Example

```
SYST:VERS?
```

Queries which version of SCPI is used in the U8480 Series.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

12 TRIGger Subsystem

This chapter explains how the **TRIGger** command subsystem is used to synchronize device actions with events.

TRIGger Command Subsystem

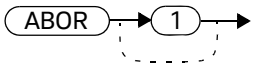
The **TRIGger** command subsystem is used to synchronize device actions with events. It includes the **ABORt**,**INITiate**, and **TRIGger** commands. These are all at the root level in the command hierarchy, but they are grouped here because of their close functional relationship.

ABORt[1]

This command removes the U8480 Series from the wait-for-trigger state and places it in the idle state. It does not affect any other settings of the trigger system. When the **INITiate** command is sent, the trigger system responds as it did before **ABORt[1]** was executed.

If **INITiate[1]:CONTinuous** is **ON**, then after **ABORt[1]**, the measurement immediately goes into the wait-for-trigger state.

Syntax



Example

ABOR *This command places the U8480 Series in the idle state.*

INITiate Commands

INITiate commands allow you to place the U8480 Series in the wait-for-trigger state.

The **INITiate** commands are overlapped, that is, the U8480 Series can continue parsing and executing subsequent commands while initiated. Note that the pending operation flag is set when the U8480 Series moves out of the idle state, and the flag is cleared when it re-enters the idle state.

The following commands are described in this section:

```
INITiate[1]:CONTinuous <boolean>
INITiate[1][:IMMediate]
INITiate[1]:CONTinuous:ALL <boolean>
INITiate[1]:CONTinuous:SEQuence[1] <boolean>
INITiate[1][:IMMediate]:ALL
INITiate[1][:IMMediate]:SEQuence[1]
```

INITiate[1]:CONTinuous <boolean>

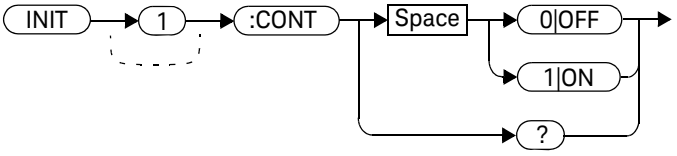
This command sets the U8480 Series for either a single trigger cycle or continuous trigger cycles. A trigger cycle means that the U8480 Series exits the wait-for-trigger state and starts a measurement.

If `INITiate[1]:CONTinuous` is set to:

- `OFF`, the trigger system remains in the idle state until it is set to `ON` or `INITiate[1][:IMMediate]` is received. Once this trigger cycle is complete, the trigger system returns to the idle state.
- `ON`, the trigger system is initiated and exits the idle state. On completion of each trigger cycle, the trigger system immediately commences another trigger cycle without entering the idle state.

NOTE This command performs the same function as `INITiate[1]:CONTinuous:SEQuence[1] <boolean>`.

Syntax



Example

```
INIT:CONT ON
```

This command places the U8480 Series in the wait-for-trigger state.

Reset condition

On reset (***RST**), this command is set to **OFF**.

On preset (**SYSTem:PRESet**) and U8480 Series power-up, **INITiate[1]:CONTinuous** is set to **ON**.

Query

INITiate[1]:CONTinuous?

The query enters a 1 or 0 into the output buffer.

- 1 is returned when there is continuous triggering
- 0 is returned when there is only a single trigger

Query example

INIT:CONT?	*Queries whether the U8480 Series is set for single or continuous triggering.*

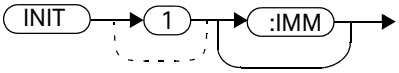
INITiate[1][:IMMediate]

This command sets the U8480 Series in the wait-for-trigger state. When a trigger is received, the measurement is taken and the result is placed in the U8480 Series memory. If **TRIGger[1]:SOURce** is set to **IMMediate**, the measurement begins as soon as **INITiate[1][:IMMediate]** is executed.

Use **FETCh?** to transfer a measurement from memory to the output buffer. Refer to "FETCh[1]? Query" on page 78 for further details.

NOTE This command performs the same function as **INITiate[1][:IMMediate]:SEQuence[1].**

Syntax



Example

INIT *This command places the U8480 Series in the wait-for-trigger state.*

Error message

If the U8480 Series is not in the idle state or **INITiate[1]:CONTinuous** is **ON**, error –213, "INIT ignored" occurs.

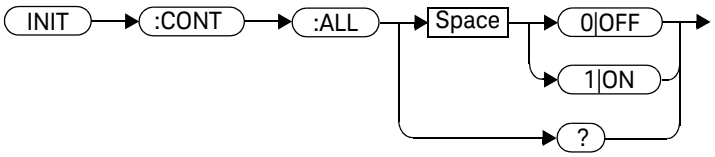
INITiate[1]:CONTinuous:ALL <boolean>

Sets all trigger sequences to be continuously initiated.

If **INITiate[1]:CONTinuous:ALL** is set to:

- **ON**, trigger sequences are set to be continuously initiated
- **OFF**, trigger sequences are not set to be continuously initiated

Syntax



Example

```
INIT:CONT:ALL ON
```

This command sets all trigger sequences to be continuously initiated.

Reset condition

On reset (***RST**), this command is set to **OFF**.

On preset (**SYSTem:PRESet**) and U8480 Series power-up, this command is set to **ON**.

Query

`INITiate[1]:CONTinuous:ALL?`

The query enters a 1 or 0 into the output buffer.

- 1 is returned when trigger sequences are set to be continuous
- 0 is returned when trigger sequences are not set to be continuous

Query example

`INIT:CONT:ALL?`	*Queries whether the U8480 Series is in a wait-for-trigger state.*

INITiate[1]:CONTinuous:SEQuence[1] <boolean>

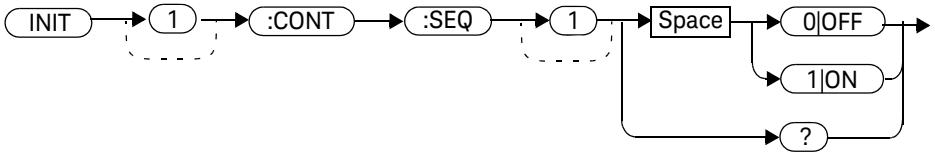
This command sets the U8480 Series for either a single trigger cycle or continuous trigger cycles. A trigger cycle means that the U8480 Series exits the wait-for-trigger state and starts a measurement.

If `INITiate[1]:CONTinuous:SEQuence[1] <boolean>` is set to:

- `OFF`, the trigger system remains in the idle state until it is set to `ON` or `INITiate[1]:IMMediate` is received. Once this trigger cycle is complete, the trigger system returns to the idle state.
- `ON`, the trigger system is initiated and exits the idle state. On completion of each trigger cycle, the trigger system immediately commences another trigger cycle without entering the idle state.

NOTE This command performs the same functions as `INITiate[1]:CONTinuous <boolean>`.

Syntax



Example

```
INIT:CONT:SEQ ON
```

This command places the U8480 Series in a wait-for-trigger state.

Reset condition

On reset (`*RST`), this command is disabled.

On preset (`SYSTem:PRESet`) and U8480 Series power-up, this command is enabled.

Query

`INITiate[1]:CONTinuous:SEQuence[1]?`

The query enters a 1 or 0 into the output buffer.

- 1 is returned when there is continuous triggering
- 0 is returned when there is only a single trigger

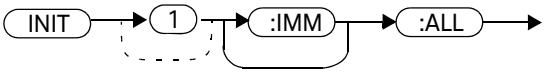
Query example

`INIT:CONT:SEQ?`	*Queries whether the U8480 Series is set for single or continuous triggering.*

INITiate[1][:IMMediate]:ALL

This command initiates all trigger sequences.

Syntax



Example

`INIT:ALL`	*This command initiates all trigger sequences.*

Error messages

If the U8480 Series is not in the idle state or **`INITiate[1]:CONTinuous`** is **`ON`**, error –213, “INIT ignored” occurs.

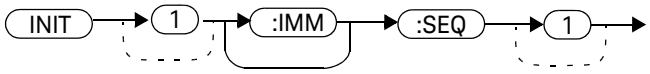
INITiate[1][:IMMediate]:SEQuence[1]

This command sets the U8480 Series in the wait-for-trigger state. When a trigger is received, the measurement is taken and the result is placed in the U8480 Series memory. If **TRIGger[1]:SOURce** is set to **IMMediate**, the measurement begins as soon as **INITiate[1][:IMMediate]** is executed.

Use **FETCh?** to transfer a measurement from memory to the output buffer. Refer to "FETCh[1]? Query" on page 78 for further information.

NOTE **This command performs the same function as INITiate[1][:IMMediate].**

Syntax



Example

`INIT:SEQ`	*This command places the U8480 Series in the wait-for-trigger state.*

Error messages

If the U8480 Series is not in the "idle" state or **INITiate[1]:CONTinuous** is **ON**, error –213, "INIT ignored" occurs.

TRIGger Commands

TRIGger commands control the behavior of the trigger system.

The following commands are described in this section:

```
TRIGger[1]:DELay:AUTO <boolean>
TRIGger[1][:IMMediate]
TRIGger[1]:SOURce BUS|EXTernal|HOLD|IMMediate
TRIGger[:SEQuence]:DELay <numeric_value>
TRIGger[:SEQuence]:SLOPe <character_data>
TRIGger[:SEQuence[1]]:COUNt <numeric_value>
TRIGger[:SEQuence[1]]:DELay:AUTO <boolean>
TRIGger[:SEQuence[1]]:IMMediate
TRIGger[:SEQuence[1]]:SOURce BUS|EXTernal|HOLD|IMMediate
```

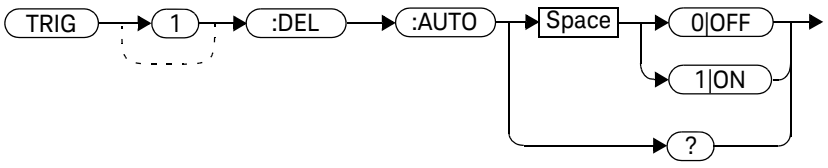
TRIGger[1]:DELay:AUTO <boolean>

This command is used to determine whether or not there is a settling-time delay before a measurement is made.

When this command is set to:

- **ON**, the U8480 Series inserts a settling-time delay before taking the requested measurement. This settling time allows the internal digital filter to be updated with new values to produce valid and accurate measurement results. The trigger with delay command allows settling time for the internal amplifiers and filters.
- In cases of large power changes, the delay may not be sufficient for a complete settling. Accurate readings can be assured by taking two successive measurements for comparison.
- **OFF**, the U8480 Series makes the measurement immediately after a trigger is received.

Syntax



Example

`TRIG:DEL:AUTO ON` *This command enables a delay on the U8480 Series.*

Reset condition

On reset, **TRIGger[1]:DELay:AUTO** is set to **ON**.

Query

TRIGger[1]:DELay:AUTO?

The query enters a 1 or 0 into the output buffer indicating the status of **TRIGger[1]:DELay:AUTO**.

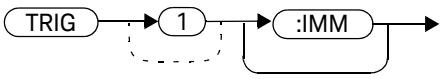
- 1 is returned when it is **ON**
- 0 is returned when it is **OFF**

TRIGger[1][:IMMediate]

This command causes a trigger to occur immediately, provided the U8480 Series is in the wait-for-trigger state. When this command is executed, the measurement result is stored in the U8480 Series memory. Use **FETCh?** to place the measurement result in the output buffer.

NOTE This command performs the same function as **INITiate[1][:IMMediate]**.

Syntax



Example

TRIG *This command causes a U8480 Series trigger to occur immediately.*

Error message

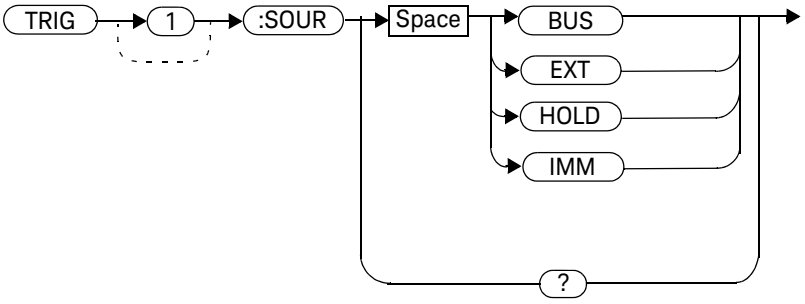
If the U8480 Series is not in the wait-for-trigger state, then **TRIGger[1][:IMMediate]** causes error –211, “Trigger ignored” to occur.

TRIGger[1]:SOURce BUS|EXTernal|HOLD|IMMediate

This command configures the trigger system to respond to the specified source. This command only selects the trigger source. Use the `INITiate[1][:IMMediate]` command to place the U8480 Series in the wait-for-trigger state.

NOTE This command has been included for compatibility purposes. It has the same purpose as `TRIGger[:SEQuence[1]]:SOURce BUS|EXTernal|HOLD|IMMediate` which should be used in preference.

Syntax



Parameters

Item	Description/Default	Range of values
source	Available trigger sources: – BUS : The trigger source is a *TRG common command or the TRIGger[1][:IMMediate] SCPI command – EXTernal : The trigger source is the trigger input in the U8480 Series – HOLD: Triggering is suspended. The only way to trigger the U8480 Series is to use TRIGger[1][:IMMediate]. – IMMediate : The trigger system is always true. If INITiate[1]:CONTinuous is ON, the U8480 Series is continually triggering free (free run mode). If an INITiate[1][:IMMediate] command is sent, a measurement is triggered, and then the U8480 Series returns to the idle state.	BUS EXTernal HOLD IMMediate

NOTE

The trigger source is set to IMMediate on U8480 Series power-up.

The MEASure and CONFigure commands automatically set the trigger source to IMMediate.

The READ? query or MEASure command should not be used if the trigger source is set to HOLD.

Example

```
TRIG:SOUR IMM
```

This command configures the U8480 Series for immediate triggering.

Reset condition

On reset, the trigger source is set to **IMMediate**.

Query

```
TRIGger[1]:SOURce?
```

The query returns the current trigger source of either **IMM**, **BUS**, **EXT**, or **HOLD**.

Query example

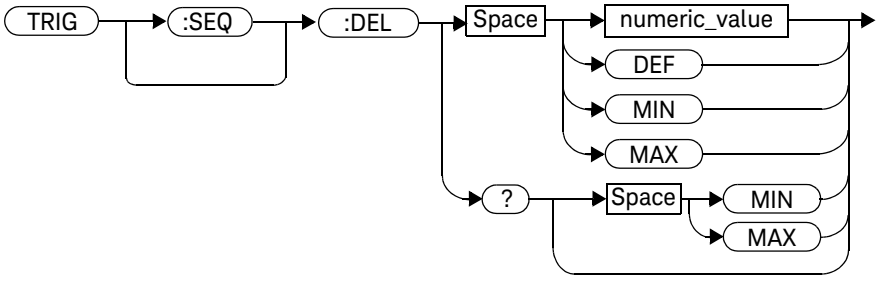
```
TRIG:SOUR?
```

Queries the U8480 Series trigger source.

TRIGger[:SEQuence]:DELay <numeric_value>

This command sets the delay between the recognition of a trigger event and the start of a measurement.

Syntax



Parameters

Item	Description/Default	Range of Values
numeric_value	The delay between the recognition of a trigger event and the start of the measurement: – **DEF**: the default value is 0 s – **MIN**: 0 s – **MAX**: 1 s Units are resolved to 10 µs.	0 to 1 s **DEF** **MIN** **MAX**

Example

```
TRIG:SEQ:DEL 0.001
```

This command sets a delay of 1 ms for the U8480 Series.

Reset condition

On reset, the trigger delay is set to 0 seconds.

Query

`TRIGger[:SEQuence]:DELay? [MIN|MAX]`

The query returns the current setting of the trigger delay or the values associated with `MIN` or `MAX`.

Query example

`TRIG:DEL?` *Queries the trigger delay.*

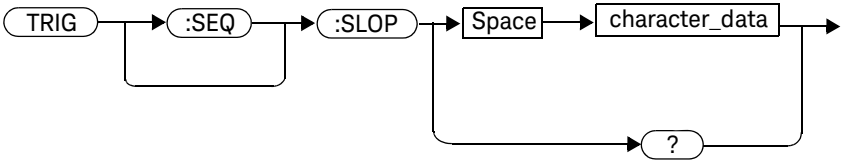
Error message

If the trigger source is not set to `EXT` while setting `TRIGger[:SEQuence]:DELay`, error –221, “Settings conflict” occurs.

TRIGger[:SEQuence]:SLOPe <character_data>

This command specifies whether a trigger event is recognized on the rising or falling edge of a signal.

Syntax



Parameters

Item	Description/Default	Range of values
character_data	How a trigger event is recognized: – `POSitive`: A trigger event is recognized on the rising edge of a signal – `NEGative`: A trigger event is recognized on the falling edge of a signal	`POSitive` `NEGative`

Reset condition

On reset, the value is set to `POSitive`.

Query

```
TRIGger[:SEQuence]:SLOPe?
```

The query returns the current value of `<character_data>`.

Query example

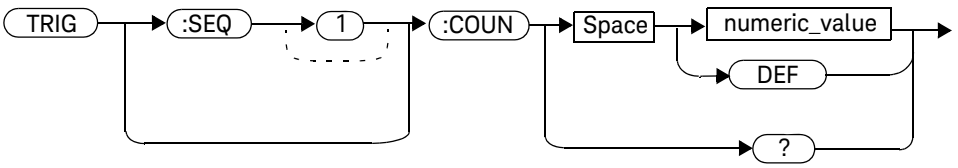
`TRIG:SLOP?`	*Queries the current value of <character_data> for the U8480 Series.*

TRIGger[:SEQuence[1]]:COUNt <numeric_value>

This command controls the path of the trigger subsystem in the upward traverse of the wait-for-trigger state. **COUNt** loops through the event detection/ measurement cycle performed. That is, **COUNt** measurements are performed in response to **COUNt** trigger events.

COUNt can be set to a value of >1 only when **[SENSe[1]:]MRATe <character_data>** is set to **FAST**.

Syntax



Parameters

Item	Description/Default	Range of values
numeric_value	The number of triggered events for the measurement cycle. – **DEF**: The default value is 1	1 to 100 **DEF**

Example

```
TRIG:COUN 10
```

This command sets the number of triggered events to 10 for the U8480 Series measurement cycle.

Reset condition

On reset, the value is set to 1.

Query

`TRIGger[:SEQuence[1]]:COUNt?`

The query returns the current setting of trigger events for the U8480 Series.

Query example

`TRIG:COUN?` *Queries the number of triggered events for the U8480 Series measurement cycle.*

Error message

If **COUNt** is >**1** when **[SENSe[1]:]MRATe <character_data>** is set to **NORMal** or **DOUBle**, error –221, "Settings conflict" occurs.

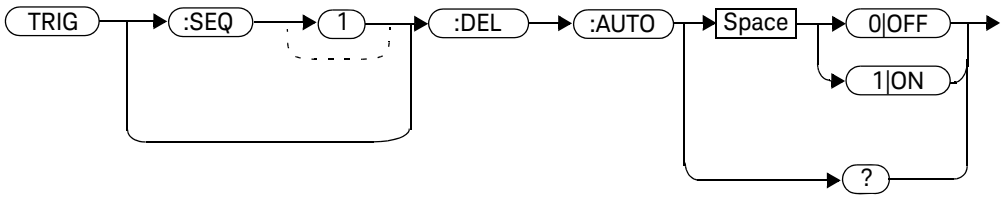
TRIGger[:SEQuence[1]]:DELay:AUTO <boolean>

This command is used to determine whether or not there is a settling-time delay before a measurement is made.

When this command is set to:

- **ON**, the U8480 Series inserts a settling-time delay before taking the requested measurement and for subsequent measurements. This settling time allows the internal digital filter to be updated with new values to produce valid, accurate measurement results. The trigger with delay command allows settling time for the internal amplifiers and filters.
- In cases of large power changes, the delay may not be sufficient for complete settling. Accurate readings can be assured by taking two successive measurements for comparison.
- **OFF**, no settling-time delay is inserted and the U8480 Series makes the measurement immediately once a trigger is received.

Syntax



Example

```
TRIG:DEL:AUTO ON
```

This command enables a delay on the U8480 Series.

Reset condition

On reset, **TRIGger[:SEQuence[1]]:DELay:AUTO** is set to **ON**.

Query

TRIGger[:SEQuence[1]]:DELay:AUTO?

The query enters a 1 or 0 into the output buffer indicating the status of **TRIGger[:SEQuence[1]]:DELay:AUTO.**

- 1 is returned when it is **ON**
- 0 is returned when it is **OFF**

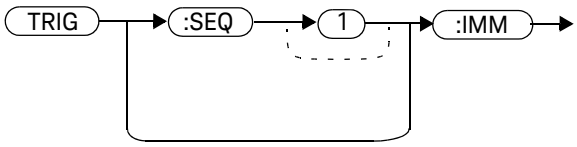
Query example

TRIG:DEL:AUTO? *Queries the settling-time delay of the U8480 Series.*

TRIGger[:SEQuence[1]]:IMMediate

This command provides a one-time override of the normal process of the downward path through the wait-for-trigger state. It causes the immediate exit of the event detection layer if the trigger system is in this layer when the command is received. In other words, the U8480 Series stops waiting for a trigger and takes a measurement ignoring any delay set.

Syntax



Example

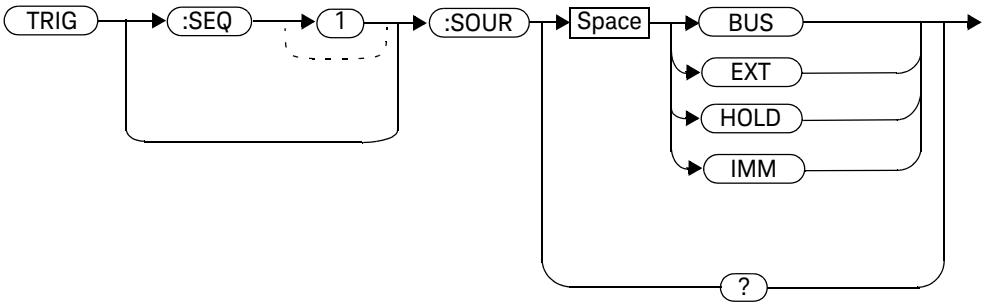
`TRIG:IMM` *This command initiates a measurement on the U8480 Series.*

TRIGger[:SEQuence[1]]:SOURce BUS|EXTernal|HOLD|IMMediate

This command configures the trigger system to respond to the specified source. This command only selects the trigger source. Use the **INITiate** command to place the U8480 Series in the wait-for-trigger state.

NOTE This command has the same purpose as `TRIGger[1]:SOURce BUS|EXTernal|HOLD|IMMediate`.

Syntax



Parameters

Item	Description/Default	Range of values

source	Available trigger sources:
	– **BUS:** The trigger source is a ***TRG** common command or the **TRIGger[1][:IMMediate]** SCPI command.
	– **EXTernal**: The trigger source is the trigger input in the U8480 Series
	– **HOLD**: Triggering is suspended. The only way to trigger the U8480 Series is to use **TRIGger[1][:IMMediate]**.
	– **IMMediate**: The trigger system is always true. If **INITiate[1]:CONTinuous** is **ON**, the U8480 Series is continually triggering free (free run mode). If an **INITiate[1][:IMMediate]** command is sent, a measurement is triggered and then the U8480 Series returns to the idle state.
		`BUS`
		`EXTernal`
		`HOLD`
		`IMMediate`

NOTE

The trigger source is set to `IMMediate` on U8480 Series power-up.

The `MEASure` and `CONFigure` commands automatically set the trigger source to `IMMediate`.

The `READ?` query or `MEASure` command should not be used if the trigger source is set to `HOLD`.

Example

`TRIG:SOUR IMM`	*This command configures the U8480 Series for immediate triggering.*

Reset condition

On reset, the trigger source is set to **`IMMediate`**.

Query

`TRIGger:SEQuence[1]:SOURce?`

The query returns the current trigger source.

Query example

`TRIG:SOUR?`	*Queries the current trigger source for the U8480 Series.*

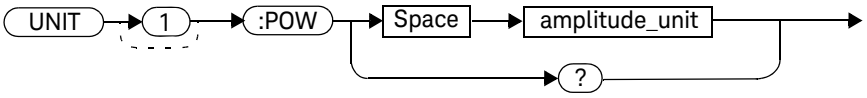
13 UNIT Subsystem

This chapter explains how the **UNIT** command subsystem is used to set the U8480 Series measurement units to Watts or dBm.

UNIT[1]:POWer <amplitude_unit>

This command sets the power measurement unit which is used for any command that accepts a numeric value in more than one unit.

Syntax



Parameters

Item	Description/Default	Range of values
amplitude_unit	The measurement unit. The default unit is dBm.	W DBM

Example

```
UNIT:POW DBM
```

This command sets the power measurement unit to dBm.

Reset condition

On reset, all measurement units are set to **DBM**.

Query

```
UNIT[1]:POWer?
```

The query returns the current setting of the power measurement unit.

Query example

```
UNIT:POW?
```

Queries which measurement unit is being used on the current measurement.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

14 IEEE-488.2 Command Reference

This chapter contains information on the IEEE-488.2 common commands supported by the U8480 Series.

SCPI Compliance Information

This chapter contains information on the IEEE-488.2 common commands that the U8480 Series supports. It also describes the USBTMC/USB488 Universal Command statements which form the nucleus of USB programming; they are understood by all instruments in the network. When combined with programming language codes, they provide all management and data communication instructions for the system.

The IEEE-488.2 common command descriptions are listed below in the alphabetical order.

*CLS

The ***CLS** (CLear Status) command clears the status data structures. The SCPI registers (Questionable Status, Operation Status, and all the other SCPI registers), the Standard Event Status Register, the Status Byte, and the Error/Event Queue are all cleared.

Syntax

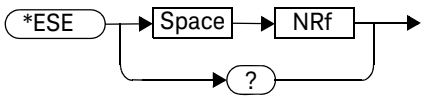
*ESE <NRf>

The ***ESE** (Event Status Enable) **<NRf>** command sets the Standard Event Status Enable Register. This register contains a mask value for the bits to be enabled in the Standard Event Status Register. A **1** in the Enable Register enables the corresponding bit in the Status Register, a **0** disables the bit. The parameter value, when rounded to an integer and expressed in base 2, represents the bit values of the Standard Event Status Enable Register. Table 14-1 shows the contents of this register.

Table 14-1 *ESE mapping

Bit	Weight	Meaning
0	1	Operation Complete
1	2	Request Control (not used)
2	4	Query Error
3	8	Device-Dependent Error
4	16	Execution Error
5	32	Command Error
6	64	User Request
7	128	Power On

Syntax



Parameters

Type	Description/Default	Range of values
NRf	A value used to set the Standard Event Status Enable Register.	0 to 255

Query

*ESE?

The query returns the current contents of the Standard Event Status Enable Register. The format of the return is <NR1> in the range of 0 to 255.

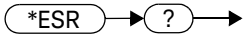
*ESR?

The ***ESR?** query returns the contents of the Standard Event Status Register and then clears it. The format of the return is **<NR1>** in the range of 0 to 255. Table 14-2 shows the contents of this register.

Table 14-2 *ESR? mapping

Bit	Weight	Meaning
0	1	Operation Complete
1	2	Request Control (not used)
2	4	Query Error
3	8	Device-Dependent Error
4	16	Execution Error
5	32	Command Error
6	64	User Request
7	128	Power On

Syntax



*IDN?

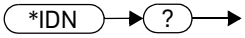
The **`*IDN?`** query allows the U8480 Series to identify itself. The string returned is:

`Keysight Technologies,U848XA,<serial number>,A1.XX.YY`

where:

- **`<serial number>`** uniquely identifies each U8480 Series.
- **`A1.XX.YY`** represents the firmware revision with **`XX`** and **`YY`** representing the major and minor revisions respectively.

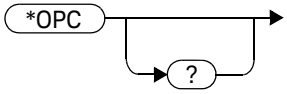
Syntax



*OPC

The ***OPC** (OPeration Complete) command causes the U8480 Series to set the operation complete bit in the Standard Event Status Register when all pending device operations have completed.

Syntax



Query

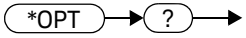
***OPC?**

The query places a 1 in the output queue when all pending device operations have completed.

*OPT?

The ***OPT?** query reports the options installed in the U8480 Series and returns a " " empty string for a standard sensor.

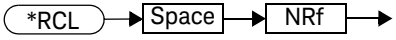
Syntax



*RCL <NRf>

The ***RCL <NRf>** (ReCaLl) command restores the state of the U8480 Series from the specified save or recall register. A sensor setup must have been stored previously in the specified register.

Syntax



Parameters

Type	Description/Default	Range of values
NRf	The number of the register to be recalled.	1 to 10

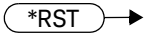
Error message

If the register does not contain a saved state, error –224, “Illegal parameter value” occurs.

*RST

The ***RST** (ReSeT) command places the U8480 Series in a known state. Refer to "SYSTem:PRESet <character_data>" on page 292 for information on reset values.

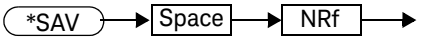
Syntax



*SAV <NRf>

The ***SAV <NRf>** (SAVe) command stores the current state of the U8480 Series in the specified register.

Syntax



Parameters

Item	Description/Default	Range of values
NRf	The number of the register that the current state of the U8480 Series is to be saved to.	1 to 10

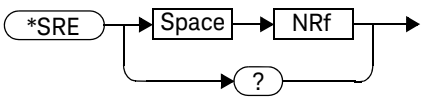
*SRE <NRf>

The **`*SRE <NRf>`** command sets the Service Request Enable register bits. This register contains a mask value for the bits to be enabled in the Status Byte Register. A **`1`** in the Enable Register enables the corresponding bit in the Status Byte Register; a **`0`** disables the bit. The parameter value, when rounded to an integer and expressed in base 2, represents bit 0 to bit 7 of the Service Request Enable Register. Table 14-3 shows the contents of this register. Refer to the Status Block Diagram at the end of Chapter 10, "STATus Subsystem" for further information.

Table 14-3 *SRE mapping

Bit	Weight	Meaning
0	1	Not used
1	2	Device Dependent
2	4	Error/Event Queue
3	8	Questionable Status Summary
4	16	Message Available
5	32	Event Status Bit
6	64	Master Summary Status
7	128	Operation Status Summary

Syntax



Parameters

Type	Description/Default	Range of values
NRf	A value used to set the Service Request Enable Register.	0 to 255

Query

`*SRE?`

The query returns the contents of the bits of the Service Request Enable Register. The format of the return is **<NR1>**.

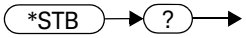
*STB?

The ***STB?** (STatus Byte) query returns bit 0 to 5 and bit 7 of the U8480 Series status byte and returns the Master Summary Status (MSS) as bit 6. The MSS is the inclusive OR of the bitwise combination (excluding bit 6) of the Status Byte and the Service Request Enable registers. The format of the return is **<NR1>** in the range of 0 to 255. Table 14-4 shows the contents of this register. Refer to the Status Block Diagram at the end of Chapter 10, "STATus Subsystem" for further information.

Table 14-4 *STB? mapping

Bit	Weight	Meaning
0	1	Not used
1	2	Device Dependent 0 - No device status conditions have occurred 1 - A device status condition has occurred
2	4	Error/Event Queue 0 - Queue empty 1 - Queue not empty
3	8	Questionable Status Summary 0 - No QUEStionable status conditions have occurred 1 - A QUEStionable status condition has occurred
4	16	Message Available 0 - No output messages are ready 1 - An output message is ready
5	32	Event Status Bit 0 - No event status conditions have occurred 1 - An event status condition has occurred
6	64	Master Summary Status 0 - The U8480 Series not requesting service 1 - There is at least one reason for requesting service
7	128	Operation Status Summary 0 - No OPERation status conditions have occurred 1 - An OPERation status condition has occurred

Syntax



*TRG

The ***TRG** (TRiGger) command triggers the U8480 Series that is in the wait-for-trigger state.

Syntax

Error message

- If **TRIGger[1]:SOURce** is not set to **BUS**, error –211, “Trigger ignored” occurs.
- If the U8480 Series is not in the wait-for-trigger state, error –211, “Trigger ignored” occurs.

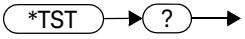
*TST?

The ***TST?** (TeST) query causes the U8480 Series to perform a self-test. The test takes approximately 25 s.

The result of the test is placed in the output queue.

- 0 is returned if the test passes
- 1 if the test fails

Syntax



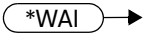
*WAI

The ***WAI** (WAIt) command causes the U8480 Series to wait until either:

- all pending operations are complete
- the device clear command is received
- power is cycled

before executing any subsequent commands or queries.

Syntax



USBTMC/USB488 Universal Commands

DCL

The **DCL** (Device Clear) command causes all USB instruments to assume a cleared condition. The definition of Device Clear is unique for each instrument. For the U8480 Series:

- All pending operations are halted.
- The parser (the software that interprets the programming codes) is reset and now expects to receive the first character of a programming code.
- The output buffer is cleared.

15 Programming Examples

This chapter provides the programming sequences or examples to remotely control the U8480 Series using SCPI commands.

Identifying the U8480 Series In Use

You can verify whether you are communicating with the right U8480 Series. Refer to "Error messages" on page 283 for more information.

```
-> *IDN?                    // Queries the identification of the U8480 Series and
                               checks whether you are communicating with the
                               right U8480 Series.
-> SYST:ERR?                // Checks the U8480 Series system error queue.
```

NOTE

"->" indicates the commands that you send to the U8480 Series.

"<-" indicates the response from the U8480 Series.

FETCh, MEASure, and READ Queries

There are three different ways to query the power measurement results using `FETC?`, `MEAS?`, and `READ?`.

In the **Free Run** or **Continuous** mode, you can use either `FETC?` or `MEAS?` to query the power measurement results.

```
-> INIT:CONT ON          // Sets the U8480 Series to the Free Run mode.
-> FETC?                 // Queries the measurement results from the
                            buffer.
```

or

```
-> INIT:CONT ON          // Sets the U8480 Series to the Free Run mode.
-> MEAS?                 // Reads the measurement results. MEAS? is
                            equivalent to CONF followed by a READ?.
```

In the **Single Trigger** mode, you can use `FETC?`, `MEAS?`, or `READ?` to query the power measurement.

```
-> INIT:CONT OFF         // Sets the U8480 Series to the Single Trigger
                            mode.
-> CONF                  // Configures the measurement.
-> INIT                  // Initializes the measurement.
-> FETC?                 // Queries the measurement results. The sequence
                            above must be followed.
```

or

```
-> INIT:CONT OFF         // Sets the U8480 Series to the Single Trigger
                            mode.
-> MEAS?                 // Reads the measurement results. MEAS? is
                            equivalent to CONF followed by a READ?.
```

or

```
-> INIT:CONT OFF        // Sets the U8480 Series to the Single Trigger
                           mode.
-> CONF                 // Configures the measurement.
-> READ?                // Reads the measurement results. READ? is
                           equivalent to INIT followed by a FETC?
                           (Assuming that TRIG:SOUR is set to
                           IMMediate).
```

NOTE

"->" indicates the commands that you send to the U8480 Series.

"<-" indicates the response from the U8480 Series.

CW Power Measurement from +20 dBm to –35 dBm

The following programming sequence examples measure continuous wave (CW) power from +20 dBm to –35 dBm with Free Run and Single Trigger modes.

Free run

```
-> SYST:PRES DEF          // Presets the U8480 Series.
-> FREQ 1000MHz           // Sets the frequency to 1000 MHz.
-> CAL:ZERO:AUTO ONCE     // Performs zeroing.
-> *OPC?                  // Waits for the operation to complete.
<- 1                      // Returns a 1 when zeroing has completed.
-> INIT:CONT ON           // Sets the U8480 Series to the Free Run
                             mode.
```

NOTE It is strongly advisable to perform zeroing on the U8480 Series for power measurement levels below –30 dBm for accurate measurements. During the zeroing process, the RF input signal must be switched off or the device-under-test disconnected from the U8480 Series.

```
-> FETC?                  // Queries the measurement results.
```

NOTE It is advisable to use `FETC?` to query the measurement results. `MEAS?` can also be used but it will trade off the measurement speed.

```
-> MEAS?                  // Reads the measurement results.
```

Single trigger

```
-> SYST:PRES DEF              // Presets the U8480 Series.
-> FREQ 1000MHz               // Sets the frequency to 1000 MHz.
-> CAL:ZERO:AUTO ONCE         // Performs zeroing.
-> *OPC?                      // Waits for the operation to complete.
<- 1                          // Returns a 1 when zeroing has completed.
-> INIT:CONT OFF              // Sets the U8480 Series to the Single Trigger
                                 mode.
```

NOTE It is strongly advisable to perform zeroing on the U8480 Series for power measurement levels below –30 dBm for accurate measurements. During the zeroing process, the RF input signal must be switched off or the device-under-test disconnected from the U8480 Series.

```
-> INIT                       // Initializes the measurement.
-> FETC?                      // Queries the measurement results.
```

NOTE In the Single Trigger mode, `INIT` must be executed before `FETC?`.

or

```
-> MEAS?                      // Reads the measurement results.
```

NOTE In the Single Trigger mode, `MEAS?` can be used without executing `INIT`.

or

```
-> CONF                    // Configures the measurement.
-> READ?                   // Reads the measurement results.
```

NOTE In the Single Trigger mode, `READ?` can be used without executing `INIT`.

NOTE "->" indicates the commands that you send to the U8480 Series.

"<-" indicates the response from the U8480 Series.

Acquiring 400 Readings/s with Buffer Mode

To acquire 400 readings/s, set the trigger mode to Free Run, and it will continuously take measurements. Set the measurement rate to the `FAST` mode and trigger count to `100` before querying the measurement.

```
-> INIT:CONT ON          // Sets to the Free Run mode.
-> MRAT FAST             // Sets the measurement rate to FAST.
-> TRIG:COUN 100         // Automatically sets to 100 in the NORMal or
                            DOUBle mode.
-> FETC?                 // Retrieves the data in the Free Run mode.
```

NOTE

"->" indicates the commands that you send to the U8480 Series.

"<-" indicates the response from the U8480 Series.

Frequency-Dependent Offset

The frequency-dependent offset (FDO) feature provides you with a convenient way to store the offset values in a handy table. Multiple FDO tables can be created to compensate different external test setups with different frequency responses. By turning on the FDO table during measurement, the pre-entered offset values will be retrieved to compensate your external test setup over a range of frequencies.

The following programming sequence example describes the simplest commands used to create a FDO table, assign an offset at a frequency value, store the table under your preferred name, and turn on the table for measurement use.

```
-> MEM:TABL:SEL “CUSTOM_A”      // Selects an FDO table named “CUSTOM_A”.
-> MEM:TABL:FREQ 50MHz          // Enters the frequency of 50 MHz into the selected table.
-> MEM:TABL:GAIN 50             // Enters the reference offset factor of 50% into the selected table.
-> CORR:CSET2 “CUSTOM_A”        // Enters the name of the selected table.
-> CORR:CSET2:STAT 1            // Enables the selected table.
-> FREQ 1000MHz                 // Sets the frequency to 1000 MHz.
-> FETC?                        // Queries the measurement results.
```

NOTE

“->” indicates the commands that you send to the U8480 Series.

“<-” indicates the response from the U8480 Series.

Frequency Sweep Operation

The frequency sweep feature is used to perform measurements in which the input signal varies in frequency. You will be required to specify the start frequency, stop frequency, and step frequency. The start frequency and stop frequency represent the start and stop sweep frequencies respectively. The step frequency represents the number of triggers of equally-spaced frequency intervals between the start and stop frequencies, inclusive of the start and stop frequencies.

In frequency sweep, the algorithm automatically selects the frequency table as reference according to the current signal and calculated frequency. The frequency sweep feature is to be used with an external trigger. When you send a trigger to the U8480 Series, it will then acquire a measurement. There is a maximum buffer of 250, which means that you can capture a maximum of 250 readings in the frequency sweep mode. You can set the filter length for each measurement.

For example, if the filter length is set to 128, the U8480 Series will take 128 readings and perform averaging to acquire one reading for the frequency sweep.

To check if the frequency sweep operation has completed, first of all, you need to enable the OPC feature by issuing the `*OPC` command. When the `*ESR?` query is issued for the first time, the returned value will not be 0. However, when the `*ESR?` query is issued for the second time, its returned value will be cleared to 0. If the subsequent `*ESR?` query returns a 1, this indicates that the frequency sweep operation has completed.

NOTE **Enabling the OPC feature will cause the OPC bit in the ESR to be set when the frequency sweep operation has completed. If the OPC feature is not enabled, the OPC bit in the ESR will not be set when the frequency sweep operation has completed.**

The following programming sequence example performs a frequency sweep.

```
-> TRIG:SOUR EXT          // Sets the external trigger source required for
                             the frequency sweep operation.
-> TRIG:SLOP POS          // Sets the U8480 Series to accept an external
                             positive-edge trigger.
```

```
-> AVER:COUN 64          // Sets the filter length to 64.
-> *OPC                  // Enables the OPC feature.
-> *ESR?                 // *ESR? is issued for the first time.
<- 129                   // Some non-zero value (any value ranging from
                         0 to 255) will be returned when *ESR? is
                         issued for the first time.
-> *ESR?                 // *ESR? is issued for the second time.
<- 0                     // The returned value will be cleared to 0 when
                         *ESR? is issued for the second time.
-> FREQ:STAR 10MHz       // Sets the start frequency to 10 MHz.
-> FREQ:STOP 100MHz      // Sets the stop frequency to 100 MHz.
-> FREQ:STEP 10          // Sets the frequency sweep to capture 10
                         triggers in equally-spaced frequency intervals
                         between 10 MHz to 100 MHz.
-> INIT:CONT ON          // Sets the U8480 Series to accept continuous
                         trigger cycles.
```

Sends a positive-edged trigger to the U8480 Series through the external trigger port.

```
-> *ESR?                 // Checks the OPC bit to confirm that the
                         frequency sweep operation has completed.
<- 0                     // Returns a 0 if the frequency sweep operation
                         has not completed.
```

Sends nine positive-edged triggers to the U8480 Series through the external trigger port.

```
-> *ESR?                 // Checks the OPC bit to confirm that the
                         frequency sweep operation has completed.
<- 1                     // Returns a 1 if the frequency sweep operation
                         has completed.
-> FETC?                 // Reads back the 10 data points captured.
```

```
-> TRIG:SOUR EXT          // Sets the external trigger source required for
                             the frequency sweep operation.
-> TRIG:SLOP POS          // Sets the U8480 Series to accept an external
                             positive-edge trigger.
```

NOTE

- "->" indicates the commands that you send to the U8480 Series. "<-" indicates the response from the U8480 Series.
- To switch to power sweep, the SENS:FREQ:STEP command must be set to 0. The SENS:BUFF:COUN command will only take effect if SENS:FREQ:STEP is set to 0.

Power Sweep Operation

The power sweep feature is used when you make a measurement in which the signal varies in amplitude. In power sweep, the U8480 Series will use the 50 MHz frequency table as default for all measurements.

The `SENS:FREQ?` query can be used to check for the current frequency table in use. The power sweep feature is to be used in conjunction with an external trigger.

When a trigger is sent to the U8480 Series, the U8480 Series will acquire a measurement. With a maximum buffer size of 250, you can capture up to 250 readings in the power sweep mode.

For each measurement, you can opt to set the filter length.

For example, if the filter length is set to 128, the U8480 Series will take up to 128 readings and perform averaging, to acquire one reading for the power sweep.

To check if the power sweep operation has completed, first of all, you will have to enable the OPC feature by issuing the `*OPC` command. When the `*ESR?` query is issued for the first time, the returned value will not be 0. However, when the `*ESR?` query is issued for the second time, its returned value will be cleared to 0. If the subsequent `*ESR?` returns a 1, this indicates that the power sweep operation has completed.

NOTE Enabling the OPC feature will cause the OPC bit in the ESR to be set when the power sweep operation has completed. If the OPC feature is not enabled, the OPC bit in the ESR will not be set when the power sweep operation has completed.

The following programming sequence example performs a power sweep.

```
-> TRIG:SOUR EXT          // Sets the external trigger source required for
                             the power sweep operation.
-> TRIG:SLOP POS          // Sets the U8480 Series to accept an external
                             positive-edge trigger.
-> AVER:COUN 64           // Sets the filter length to 64.
-> *OPC                   // Enables the OPC feature.
-> *ESR?                  // *ESR? is issued for the first time.
<- 129                    // Some non-zero value (any value ranging from
                             0 to 255) will be returned when *ESR? is
                             issued for the first time.
-> *ESR?                  // *ESR? is issued for the second time.
<- 0                      // The returned value will be cleared to 0 when
                             *ESR? is issued for the second time.
-> BUFF:COUN 2            // Sets the power sweep mode to capture two
                             triggers.
-> INIT:CONT ON           // Sets the U8480 Series to accept continuous
                             trigger cycles.
```

Sends a positive-edged trigger to the U8480 Series through the external trigger port.

```
-> *ESR?                  // Checks the OPC bit to confirm that the power
                             sweep operation has completed.
<- 0                      // Returns a 0 if the power sweep operation has
                             not completed.
```

Sends another positive-edged trigger to the U8480 Series through the external trigger port.

```
-> *ESR?                    // Checks the OPC bit to confirm that the power
                            sweep operation has completed.
<- 1                        // Returns a 1 if the power sweep operation has
                            completed.
-> FETC?                    // Reads back the two data points captured.
```

NOTE

"->" indicates the commands that you send to the U8480 Series.

"<-" indicates the response from the U8480 Series.

Gamma Correction

The Gamma Correction feature enables you to correct for the impedance mismatch between the Device-Under-Test (DUT) and the power sensor via Single Point Gamma Correction or Table-Based Gamma Correction. For Single Point Gamma Correction, the DUT Gamma you provide will be applied to measurement correction across all frequencies in the sensor operating range. For Table-Based Gamma Correction, you may provide a list of Gamma values for the desired measurement frequencies. The programming sequence examples below describe some simple commands to accomplish the following:

- Enter the Single Point Gamma
- Turn on Single Point Gamma Correction

```
-> SENS:CORR:SGAM:MAGN 0.2      // Sets DUT Gamma magnitude to 0.2
-> SENS:CORR:SGAM:PHAS -45      // Sets DUT Gamma phase to –45°
-> SENS:CORR:SGAM:STAT ON       // Turns on Single Point Gamma
                                   Correction
-> SENS:CORR:SGAM?              // Queries the DUT Gamma currently
                                   being used for measurement correction
```

The examples below describe some simple commands to accomplish the following:

- Create a Gamma table
- Assign a Gamma magnitude-phase pair of values for a frequency value
- Turn on Table-Based Gamma Correction

```
-> MEM:TABL:SEL "Gamma1"        // Selects the Gamma table named
                                   "Gamma1"
-> MEM:TABL:FREQ 50MHz          // Inputs a frequency of 50 MHz into the
                                   selected table
-> MEM:TABL:SGAM 0.1,150        // Inputs the DUT Gamma
                                   magnitude-phase pair values; 0.1 for
                                   magnitude, and 150° for phase
```

```
-> SENS:CORR:CSET3:SEL “Gamma1”   // Selects “Gamma1” as the table to be used for Table-based Gamma Correction
-> SENS:CORR:CSET3:STAT ON        // Turns on Table-Based Point Gamma Correction
-> FREQ 50MHz                     // Sets the measurement frequency to 50 MHz
```

S-Parameter Correction

The S-Parameter Correction feature enables you to correct for the effect of 2-port devices, for ideal and non-ideal DUTs in your test setup. For a non-ideal DUT, the S-Parameter Correction is enabled in tandem with Gamma Correction (refer to "Gamma Correction" on page 366). The following programming sequence examples describe some simple commands to accomplish the following:

- Create a S-Parameter table
- Assign S-Parameter data as a magnitude-phase pair of values for a frequency value
- Turn on S-Parameter Correction

```
-> MEM:TABL:SEL "SParam1"          // Selects the S-Parameter table named "SParam1"
-> MEM:TABL:FREQ 50MHz             // Inputs a frequency of 50 MHz into the selected table
-> MEM:TABL:SPAR S11,0.1,150       // Inputs the S11 magnitude-phase pair values; 0.1 for magnitude, and 150° for phase
-> MEM:TABL:SPAR S12,0.9,45        // Inputs the S12 magnitude-phase pair values; 0.9 for magnitude, and 45° for phase
-> MEM:TABL:SPAR S21,0.3,90        // Inputs the S21 magnitude-phase pair values; 0.3 for magnitude, and 90° for phase
-> MEM:TABL:SPAR S22,0.7,-135      // Inputs the S22 magnitude-phase pair values; 0.7 for magnitude, and –135° for phase
-> SENS:CORR:CSET4:SEL "SParam1"   // Selects "SParam1" as the table to be used for S-Parameter Correction
-> SENS:CORR:CSET4:STAT ON         // Turns on S-Parameter Correction
-> FREQ 50MHz                      // Sets the measurement frequency to 50 MHz
```

Real-Time Measurement Uncertainty

The Real-Time Measurement Uncertainty (RTMU) feature enables you to obtain the Measurement Uncertainty (MU) for your test setup by dynamically calculating the MU as measurements are being made. The DUT Gamma, which is used to determine the mismatch between the DUT and the sensor, may be selected as a Single Point Gamma, Table-Based Gamma, or S-Parameter.

- For Single Point Gamma, the DUT Gamma you provide will be applied to the MU calculations across all frequencies in the sensor operating range.
- For Table-Based Gamma, you may provide a list of Gamma values for the desired measurement frequencies.

NOTE The values for Single Point Gamma and Table-Based Gamma are the same as those for the Gamma Correction feature.

- For test setups that include a 2-port device, select S-Parameter as the DUT Gamma. The Gamma values will then be taken from the selected S-Parameter table.

If you want the mismatch to be included as a source of uncertainty in the MU calculation, follow the steps below:

- Disable the Gamma Correction feature
- Disable the S-Parameter Correction feature

The following programming sequence examples describe some simple commands to accomplish the steps as outlined above.

Example 1

- Enter the Single Point Gamma
- Turn on the RTMU feature

```
-> SENS:CORR:SGAM:STAT OFF          // Turns off Single Point Gamma
                                    Correction in order to include the
                                    mismatch between the DUT and the
                                    sensor in the MU calculation
-> SENS:CORR:CSET3:STAT OFF         // Turns off Table-Based Gamma
                                    Correction in order to include the
                                    mismatch between the DUT and the
                                    sensor in the MU calculation
-> SENS:MUNC:SGAM:TYPE SINGle       // Selects Single Point Gamma for the
                                    DUT
-> SENS:CORR:SGAM:MAGN 0.2          // Sets the DUT Gamma magnitude to
                                    0.2
-> SENS:MUNC:STAT ON                // Turns on the RTMU function
-> FETCH:MUNC?                      // Queries the measured power and
                                    calculated MU
```

Example 2

- Create a Gamma table
- Assign a Gamma magnitude-phase pair of values for a frequency value
- Turn on the RTMU feature

```
-> MEM:TABL:SEL "Gamma1"           // Selects the Gamma table named
                                   "Gamma1"
-> MEM:TABL:FREQ 50MHz             // Inputs a frequency of 50 MHz into the
                                   selected table
-> MEM:TABL:SGAM 0.1,150           // Inputs the DUT Gamma
                                   magnitude-phase pair values; 0.1 for
                                   magnitude, and 150° for phase
-> SENS:CORR:CSET3:SEL "Gamma1"    // Selects "Gamma1" as the table to be
                                   used for Table-Based Gamma
                                   Correction
```

```
-> SENS:CORR:CSET3:STAT OFF      // Turns off Table-Based Gamma Correction in order to include the mismatch between the DUT and the sensor in the MU calculation
-> SENS:CORR:SGAM:STAT OFF       // Turns off Single Point Gamma Correction in order to include the mismatch between the DUT and the sensor in the MU calculation
-> FREQ 50MHz                    // Sets the measurement frequency to 50 MHz
-> SENS:MUNC:SGAM:TYPE TABLe     // Selects Table-Based Gamma for the DUT
-> SENS:MUNC:STAT ON             // Turns on the RTMU function
-> FETCH:MUNC?                   // Queries the measured power and calculated MU
```

Example 3

- Create an S-Parameter table
- Assign S-Parameter data in the magnitude-phase format for a frequency value
- Select S-Parameter as the DUT Gamma
- Turn on the RTMU feature

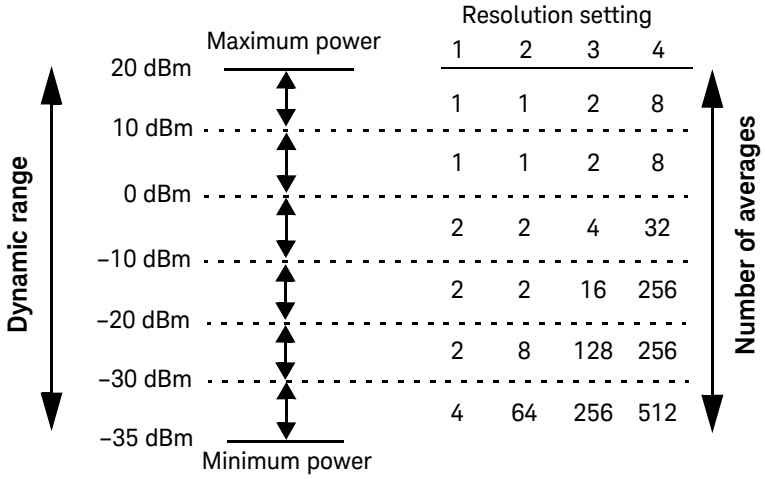
```
-> MEM:TABL:SEL “SParam1”        // Selects the S-Parameter table named “SParam1”
-> MEM:TABL:FREQ 50MHz           // Inputs a frequency of 50 MHz into the selected table
-> MEM:TABL:SPAR S11,0.1,150     // Inputs the S11 magnitude-phase pair values; 0.1 for magnitude, and 150° for phase
-> MEM:TABL:SPAR S12,0.9,45      // Inputs the S12 magnitude-phase pair values; 0.9 for magnitude, and 45° for phase
```

```
-> MEM:TABL:SPAR S21,0.3,90          // Inputs the S21 magnitude-phase pair values; 0.3 for magnitude, and 90° for phase
-> MEM:TABL:SPAR S22,0.7,-135        // Inputs the S22 magnitude-phase pair values; 0.7 for magnitude, and –135° for phase
-> SENS:CORR:CSET4:STAT OFF          // Turns off S-Parameter Correction in order to include the mismatch between the DUT and the sensor in the MU calculation
-> SENS:CORR:CSET4:SEL "SParam1"     // Selects "SParam1" as the table to be used to obtain the DUT Gamma
-> SENS:MUNC:SGAM:TYPE SPAR          // Selects S-Parameter as the DUT Gamma
-> FREQ 50MHz                        // Sets the measurement frequency to 50 MHz
-> SENS:MUNC:STAT ON                 // Turns on the RTMU function
-> FETCH:MUNC?                       // Queries the measured power and calculated MU
```

A Appendix

Auto-Averaging Settings

The figure below shows the averaged number of readings for each range and resolution when the U8480 Series is in the auto-measurement average mode.



Edition 9, April 15, 2019

Printed in Malaysia

U8481-90003
www.keysight.com