

Keysight U2000 Series USB Power Sensors



Programming
Guide

Notices

Copyright Notice

© Keysight Technologies 2007–2016

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Keysight Technologies as governed by United States and international copyright laws.

Manual Part Number

U2000-90411

Edition

Edition 7, April 15, 2016

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.

General 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 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. Duration and conditions of warranty for this product may be superseded when the product is integrated into (becomes a part of) other Keysight products. During the warranty period, Keysight will, at its option, either repair or replace products which prove to be defective. The warranty period begins on the date of delivery or on the date of installation if installed by Keysight.

Restricted Rights Legend

The Software and Documentation have been developed entirely at private expense. They are delivered and licensed as “commercial computer software” as defined in DFARS 252.227-7013 (Oct 1988), DFARS 252.211-7015 (May 1991), or DFARS 252.227-7014 (Jun 1995), as a “commercial item” as defined in FAR 2.101(a), or as “restricted computer software” as defined in FAR 52.227-19 (Jun 1987) (or any equivalent agency regulation or contract clause), whichever is applicable. You have only those rights provided for such Software and Documentation by the applicable FAR or DFARS clause or the Keysight standard software agreement for the product involved.

Equipment Operation

Warnings and Cautions

This guide uses warnings and cautions to denote hazards.

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 loss of life. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

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.

Personal Safety Considerations

This is a Safety Class I product (provided with a protective earthing ground incorporated in the power cord). The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. Any interruption of the protective conductor, inside or outside the instrument, is likely to make the instrument dangerous. Intentional interruption is prohibited. If this instrument is not used as specified, the protection provided by the equipment could be impaired. This instrument must be used in a normal condition (in which all means of protection are intact) only.

No operator serviceable parts inside. Refer servicing to qualified personnel. To prevent electrical shock, do not remove covers. For continued protection against fire hazard, replace the line fuse(s) only with fuses of the same type and rating (for example, normal blow, time delay, etc.). The use of other fuses or material is prohibited.

General Safety Considerations

The following general safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument. Keysight Technologies, Inc. assumes no liability for the customer's failure to comply with these requirements.

WARNING

BEFORE CONNECTING THE POWER SENSOR TO OTHER INSTRUMENTS ensure that all instruments are connected to the protective (earth) ground. Any interruption of the protective earth grounding will cause a potential shock hazard that could result in personal injury.

CAUTION

- Use the device with the cables provided.
 - Repair or service that is not covered in this manual should only be performed by qualified personnels.
-

User Environment

This instrument is designed for indoor use only.

In This Guide...

- 1 U2000 Series USB Power Sensors Remote Operation** describes the configuration and settings of the U2000 Series power sensors.
- 2 Measurement Commands** explains how to use the MEASure group of instructions to acquire data using a set of high level instructions.
- 3 CALibration Subsystem** explains how the CALibration subsystem is used to perform zeroing on the U2000 Series power sensors.
- 4 CALCulate Subsystem** explains how the CALCulate subsystem is used to perform post acquisition data processing.
- 5 FORMat Subsystem** explains how the FORMat subsystem is used to set the data format for transferring numeric information.
- 6 MEMory Subsystem** explains how the MEMory subsystem is used to configure sensor frequency-dependent offset tables and save/recall registers.
- 7 SENSE Subsystem** explains how the SENSE subsystem directly affects device-specific settings that are used to make measurements.
- 8 SERVICE Subsystem** explains how the SERVICE subsystem is used to obtain and set information useful for servicing the U2000 Series power sensors.
- 9 STATus Subsystem** explains how the STATus subsystem enables you to examine the status of the U2000 Series power sensors by monitoring the “Device Status Register”, “Operation Status Register”, and the “Questionable Status Register”.
- 10 SYSTEM Subsystem** explains how the SYSTEM subsystem is used to return error numbers and messages from the U2000 Series power sensors, preset the U2000 Series power sensors, and query the SCPI version.
- 11 TRACe Subsystem** explains how the TRACe subsystem is used to configure and capture the measured power trace.
- 12 TRIGger Subsystem** explains how the TRIGger subsystem is used to synchronize device actions with events.

- 13 **UNIT Subsystem** explains how the UNIT subsystem is used to set the U2000 Series power sensor measurement units to Watts or dBm.
- 14 **IEEE 488.2 Command Reference** describes IEEE 488.2 common commands that are supported by the U2000 Series power sensors.
- 15 **Programming Examples** explains some typical programming examples for the U2000 Series power sensors.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

Table of Contents

General Warranty	3
Restricted Rights Legend	3
Equipment Operation	4
General Safety Considerations	5
User Environment	5
In This Guide...	6
1 U2000 Series USB Power Sensors Remote Operation	
Introduction	22
USB Configuration	23
Zeroing and Calibrating the U2000 Series USB Power Sensors	24
Making Measurements	26
Using Frequency-dependent Offset Tables	36
Setting the Range and Averaging	43
Setting Offsets	47
Setting Measurement Limits	48
Getting the Best Speed Performance	52
How Measurements are Calculated	56
Status Reporting	57
Saving and Recalling U2000 Series Power Sensor Configurations	71
Using Device Clear to Halt Measurements	72
An Introduction to the SCPI Language	73
SCPI Compliance Information	82
Summary of Commands	83
2 Measurement Commands	
Measurement Commands	86
CONFigure[1]?	89
CONFigure[1] Command	90

CONFigure[1][:SCALar][:POWer:AC] [<expected_value>[,<resolution>[,<source list>]]]	91
FETCh[1]? Query	93
FETCh[1][:SCALar][:POWer:AC]? [<expected_value>[,<resolution>[,<source list>]]]	94
READ[1] Query	96
READ[1][:SCALar][:POWer:AC]? [<expected_value>[,<resolution>[,<source list>]]]	97
MEASure[1] Query	100
MEASure[1][:SCALar][:POWer:AC]? [<expected_value>[,<resolution>[,<source list>]]]	101
3 CALibration Subsystem	
CALibration Subsystem	104
CALibration[1][:ALL]	105
CALibration[1][:ALL]?	106
CALibration[1]:ZERO:AUTO [ONCE ON OFF 0 1]	107
CALibration[1]:ZERO:TYPE <EXTernal INTernal>	109
4 CALCulate Subsystem	
CALCulate Subsystem	112
CALCulate[1]:FEED[1]<"string">	114
CALCulate[1]:LIMit Commands	116
CALCulate[1]:LIMit:CLEar:AUTO <boolean> ONCE	117
CALCulate[1]:LIMit:CLEar[:IMMediate]	119
CALCulate[1]:LIMit:FAIL?	120
CALCulate[1]:LIMit:FCOunt?	121
CALCulate[1]:LIMit:LOWer[:DATA] <numeric_value>	123
CALCulate[1]:LIMit:UPPer[:DATA] <numeric_value>	125
CALCulate[1]:LIMit:STATe <boolean>	127
CALCulate[1]:MATH Commands	129
CALCulate[1]:MATH[:EXPRession] <"string">	130

CALCulate[1]:MATH[:EXPRession]:CATalog?	132
5 FORMat Subsystem	
FORMat Subsystem	134
FORMat[:READings]:BORDER <character_data>	135
FORMat[:READings][:DATA] <character_data>	137
6 MEMory Subsystem	
MEMory Subsystem	140
MEMory:CATalog Queries	142
MEMory:CATalog[:ALL]?	143
MEMory:CATalog:STATe?	145
MEMory:CATalog:TABLE?	146
MEMory:CLEar Commands	148
MEMory:CLEar[:NAME] <"character_data">	149
MEMory:CLEar:TABLE	151
MEMory:FREE Queries	152
MEMory:FREE[:ALL]?	153
MEMory:FREE:STATe?	154
MEMory:FREE:TABLE?	155
MEMory:NStates?	156
MEMory:STATe Commands	157
MEMory:STATe:CATalog?	158
MEMory:STATe:DEFine <"character_data">,<numeric_value>	159
MEMory:TABLE Commands	161
MEMory:TABLE:FREQuency <numeric_value>{,<numeric_value>}	162
MEMory:TABLE:FREQuency:POINts?	165
MEMory:TABLE:GAIN[:MAGNitude] <numeric_value>{,<numeric_value>}	166
MEMory:TABLE:GAIN[:MAGNitude]:POINts?	168
MEMory:TABLE:MOVE <"character_data">,<"character_data">	169

MEMory:TABLE:SElect <“character_data”> 171

7 SENSE Subsystem

[SENSe] Subsystem	175
[SENSe[1]:]AVERage Commands	177
[SENSe[1]:]AVERage:COUNT <numeric_value>	178
[SENSe[1]:]AVERage:COUNT:AUTO <boolean>	180
Example 1: Query Example	182
[SENSe[1]:]AVERage:SDETect <boolean>	183
[SENSe[1]:]AVERage[:STATe] <boolean>	185
[SENSe[1]:]BUFFer:COUNT <numeric_value>	187
[SENSe[1]:]CORRection:CSET2 Commands	189
[SENSe[1]:]CORRection:CSET2[:SElect] <“string”>	190
[SENSe[1]:]CORRection:CSET2:STATe <boolean>	192
[SENSe[1]:]CORRection:DCYClE GAIN3[:INPut][:MAGNitude] <numeric_value>	194
[SENSe[1]:]CORRection:DCYClE GAIN3:STATe <boolean>	197
[SENSe[1]:]CORRection:FDOFFset GAIN4[:INPut][:MAGNitude]?	199
[SENSe[1]:]CORRection:GAIN2 Commands	200
[SENSe[1]:]CORRection:GAIN2:STATe <boolean>	201
[SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude] <numeric_value>	203
[SENSe[1]:]DETector:FUNCTion <character_data>	206
[SENSe[1]:]FREQuency[:CW]:FIXed <numeric_value>	209
[SENSe[1]:]FREQuency[:CW]:FIXed:STARt <numeric_value> <unit>	211
[SENSe[1]:]FREQuency[:CW]:FIXed:STEP <numeric_value>	214
[SENSe[1]:]FREQuency[:CW]:FIXed:STOP <numeric_value> <unit>	217
[SENSe[1]:]MRATe <character_data>	220
[SENSe[1]:]POWer:AC:RANGe <numeric_value>	223
[SENSe[1]:]POWer:AC:RANGe:AUTO <boolean>	225
[SENSe[1]:]SPEEd <numeric_value>	227

[SENSe[1]:]SWEep[1] Commands	229
[SENSe[1]:]SWEep[1]:OFFSet:TIME <numeric_value>	231
[SENSe[1]:]SWEep[1]:TIME <numeric_value>	233
[SENSe[1]:]TEMPerature?	235
[SENSe[1]:]TRACe Commands	236
[SENSe[1]:]TRACe:OFFSet:TIME <numeric_value>	237
[SENSe[1]:]TRACe:TIME <numeric_value>	239
[SENSe[1]:]TRACe:UNIT <character_data>	241

8 SERVICE Subsystem

SERVICE Subsystem	244
SERVICE:BIST:TRIGger:LEVel:STATe?	245
SERVICE:OPTion?	246
SERVICE:SENSor[1]:CDATe?	247
SERVICE:SENSor[1]:CDUEdate <“date”>	248
SERVICE:SENSor[1]:CPLace <“place”>	249
SERVICE:SENSor[1]:FREQuency:MAXimum?	251
SERVICE:SENSor[1]:FREQuency:MINimum?	252
SERVICE:SENSor[1]:POWer:AVERage:MAXimum?	253
SERVICE:SENSor[1]:POWer:USABLE:MAXimum?	254
SERVICE:SENSor[1]:POWer:USABLE:MINimum?	255
SERVICE:SENSor[1]:RADc?	256
SERVICE:SENSor[1]:SNUMber?	257
SERVICE:SENSor[1]:TNUMber <“tracking_number”>	258
SERVICE:SENSor[1]:TYPE?	259
SERVICE:SNUMber?	260

9 STATUS Subsystem

STATUS Subsystem	262
Status Register Set Commands	264
Device Status Register Sets	269

Operation Register Sets	270
STATus:OPERation	271
STATus:OPERation:CALibrating[:SUMMARY]	272
STATus:OPERation:LLFail[:SUMMARY]	273
STATus:OPERation:MEASuring[:SUMMARY]	274
STATus:OPERation:SENSe[:SUMMARY]	275
STATus:OPERation:TRIGger[:SUMMARY]	276
STATus:OPERation:ULFail[:SUMMARY]	277
STATus:PRESet	278
Questionable Register Sets	279
STATus:QUEStionable	280
STATus:QUEStionable:CALibration[:SUMMARY]	281
STATus:QUEStionable:POWEr[:SUMMARY]	282
10 SYSTEM Subsystem	
SYSTEM Subsystem	286
SYSTEM:ERRor?	287
SYSTEM:HELP:HEADers?	294
SYSTEM:PERSONa:MANufacturer <"string">	295
SYSTEM:PERSONa:MANufacturer:DEFault	297
SYSTEM:PRESet <character_data>	298
SYSTEM:VERSIon?	302
11 TRACe Subsystem	
TRACe Subsystem	304
TRACe[1][:DATA]? <character_data>	305
TRACe[1]:STATe <boolean>	307
TRACe[1]:UNIT <character_data>	309
12 TRIGger Subsystem	
TRIGger Subsystem	312
ABORT[1]	314

INITiate Commands	315
INITiate[1]:CONTinuous <boolean>	316
INITiate[1][:IMMEDIATE]	318
INITiate[1]:CONTinuous:ALL <boolean>	319
INITiate[1]:CONTinuous:SEQuence[1] <boolean>	321
INITiate[1][:IMMEDIATE]:ALL	323
INITiate[1][:IMMEDIATE]:SEQuence[1]	324
TRIGger Commands	325
TRIGger[1]:DELay:AUTO <boolean>	326
TRIGger[1][:IMMEDIATE]	328
TRIGger[1]:SOURce BUS INTernal[1] EXTernal HOLD IMMEDIATE	329
TRIGger[:SEQuence]:DELay <numeric_value>	332
TRIGger[:SEQuence]:HOLDoff <numeric_value>	334
TRIGger[:SEQuence]:HYSTeresis <numeric_value>	336
TRIGger[:SEQuence]:LEVel <numeric_value>	338
TRIGger[:SEQuence]:SLOPe <character_data>	340
TRIGger[:SEQuence[1]]:COUNT <numeric_value>	342
TRIGger[:SEQuence[1]]:DELay:AUTO <boolean>	344
TRIGger[:SEQuence[1]]:IMMEDIATE	346
TRIGger[:SEQuence[1]]:SOURce BUS INTernal[1] EXTernal HOLD IMMEDIATE	347

13 UNIT Subsystem

UNIT Subsystem	352
UNIT[1]:POWER <amplitude_unit>	353

14 IEEE 488.2 Command Reference

SCPI Compliance Information	356
*CLS	357
*ESE <NRf>	358
*ESR?	360

*IDN?	361
*OPC	362
*OPT?	363
*RCL <NRf>	364
*RST	365
*SAV <NRf>	366
*SRE <NRf>	367
*STB?	369
*TRG	371
*TST?	372
*WAI	373
USBTMC/USB488 Universal Commands	374

15 Programming Examples

Example 1: Identifying the U2000 Series Power Sensor In Use	376
Example 2: Internal/External Zeroing	377
Example 3: FETCh, MEASure, and READ Queries	378
Example 4: CW Power Measurement from +44 dBm to -60 dBm	380
Example 5: Average Modulated Power Measurement	383
Example 6: Average Pulse Power Measurement with Duty Cycle	384
Example 7: Time-Gated Burst Power Measurement via Internal/External Trigger	385
Example 8: Creating a Trace Graph Display	387
Example 9: Acquiring 100, 250, and 1000 Readings/s	388
Example 10: Frequency-Dependent Offset	390
Example 11: Power Sweep Operation	391
Example 12: Frequency Sweep Operation	394

A Appendix A

Auto-Averaging Settings	398
-------------------------	-----

List of Figures

Figure 1-1	Frequency-dependent offset tables	37
Figure 1-2	U2000/1/2/4A averaging range hysteresis	44
Figure 1-3	U2000/1/2H averaging range hysteresis	44
Figure 1-4	U2000/1B averaging range hysteresis	45
Figure 1-5	Limits checking results	48
Figure 1-6	How measurements are calculated	56
Figure 1-7	Generalized status register model	57
Figure 1-8	Typical status register bit changes	59
Figure 1-9	Status system	62
Figure 1-10	Hierarchical structure of SCPI	73
Figure 1-11	Format of <character_data>	76
Figure 1-12	Format of <non-decimal numeric>	77
Figure 1-13	Format of <NR1>	78
Figure 1-14	Format of <NR2>	78
Figure 1-15	Format of <NR3>	79
Figure 1-16	Format of <string>	80
Figure 2-1	Measurement display CALCulate block channel	86
Figure 4-1	Measurement display CALCulate block channel	112
Figure 4-2	CALCulate block	112
Figure 7-1	Example of averaged readings	180
Figure 10-1	IEEE 488.2 arbitrary block program data format	294

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

List of Tables

Table 1-1	MEASure? and CONFigure preset states	26
Table 1-2	Settling time for Normal Speed and x2 Speed	46
Table 1-3	Range of values for measurement limits	49
Table 1-4	Bit definitions - Status byte register	63
Table 1-5	Bit definitions - Standard event register	64
Table 1-6	Bit definitions - Questionable status registers	65
Table 1-7	Bit change conditions for Questionable status register	66
Table 1-8	Bit definitions - Operation status	67
Table 1-9	Bit change conditions for Operation status	68
Table 1-10	Bit definitions - Device status register	69
Table 1-11	Bit change conditions for Device status register	69
Table 6-1	Frequency and offset factor list	162
Table 6-2	Frequency and offset factor list	166
Table 9-1	Commands and events affecting Status register	262
Table 10-1	DEFault: U2000 Series power sensor presets	299
Table 14-1	*ESE mapping	358
Table 14-2	*ESR? mapping	360
Table 14-3	*SRE mapping	367
Table 14-4	*STB? mapping	369

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

1 U2000 Series USB Power Sensors Remote Operation

Introduction	22
USB Configuration	23
Zeroing and Calibrating the U2000 Series USB Power Sensors	24
Making Measurements	26
Using Frequency-dependent Offset Tables	36
Setting the Range and Averaging	43
Range	47
Setting Offsets	47
Setting Measurement Limits	48
Getting the Best Speed Performance	52
How Measurements are Calculated	56
Status Reporting	57
Saving and Recalling U2000 Series Power Sensor Configurations	71
Using Device Clear to Halt Measurements	72
An Introduction to the SCPI Language	73
SCPI Compliance Information	82
Summary of Commands	83

This chapter describes the configuration and settings of the U2000 Series USB power sensors.

Introduction

This chapter contains the following sections:

- “USB Configuration” on page 23.
- “Zeroing and Calibrating the U2000 Series USB Power Sensors” on page 24.
- “Making Measurements” on page 26.
- “Using Frequency-dependent Offset Tables” on page 36.
- “Setting the Range and Averaging” on page 43.
- “Setting Offsets” on page 47.
- “Setting Measurement Limits” on page 48.
- “Getting the Best Speed Performance” on page 52.
- “How Measurements are Calculated” on page 56.
- “Status Reporting” on page 57.
- “Saving and Recalling U2000 Series Power Sensor Configurations” on page 71.
- “Using Device Clear to Halt Measurements” on page 72.
- “An Introduction to the SCPI Language” on page 73.
- “SCPI Compliance Information” on page 82.
- “Summary of Commands” on page 83.

USB Configuration

The Keysight U2000 Series power sensor comes with a USB interface and it 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.

You are recommended to verify the connection of the U2000 Series power sensor to the PC by sending the ***IDN?** default command via the Keysight Connection Expert.

For more information on configuring the USB remote interface connectivity, refer to the *Keysight Technologies USB/LAN/GPIB Interfaces Connectivity Guide*. If you have installed the IO Libraries Suite, you can access the Connectivity Guide via the Keysight IO Libraries Control icon. Alternatively, you can access the Connectivity Guide 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.

Zeroing and Calibrating the U2000 Series USB Power Sensors

The U2000 Series power sensors do not require calibration with a 50 MHz reference source as was commonly done with Keysight power meters and power sensors in the past. However, zeroing is necessary before measurements are made to achieve the best accuracy possible. There are two zeroing types available to the user. Internal zeroing uses an electronic switch to isolate the U2000 Series power sensor RF input port from the internal measurement circuitry during the procedure, thereby allowing the sensor to be physically connected to an active RF source. Therefore, you do not need to disconnect the RF source or power it off during an internal zeroing. This feature makes zeroing more convenient, but one may only use internal zeroing if zero set (internal) is within the user's application requirement. In the second method, external zeroing, the sensor does not utilize the electronic switch to isolate the measurement circuitry from the RF input port. In this case, the RF power must be removed from the RF input port either by turning the source off or physically removing it from the sensor during the external zeroing procedure. External zeroing generally has better zero set performance. The **INTernal** or **EXTernal** zeroing selection should be made based on the measurement needs. It is very important to note that after setting the zeroing type to either **INTernal** or **EXTernal**, the CALibration subsystem is used to perform zeroing on the U2000 Series power sensor.

Zeroing

Zeroing adjusts the U2000 Series power sensor for a zero power reading with no power applied to the RF input port.

The command **CALibration[1]:ZERO:AUTO [ONCE|ON|OFF|0|1]** causes the U2000 Series power sensor to perform its zeroing routine when enabled.

The **1|ON** parameter can only be used for internal zeroing. When **1** or **ON** is specified, zeroing of the U2000 Series power sensor happens automatically:

- when a 5 °C change in temperature occurs
- when the connection to the U2000 Series power sensor is established
- every 24 hours

To perform external zeroing, the command **CALibration[1]:ZERO:TYPE EXTernal** is used, followed by **CALibration[1]:ZERO:AUTO ONCE** to start the zeroing process.

Calibration

This feature allows you to perform calibration with a single command. The command is:

CALibration[1][:ALL]

The calibration consists of zeroing the U2000 Series power sensors.

You can query the calibration status by issuing a query. The query is

CALibration[1][:ALL]?

If the result is 0, the calibration is successful. If the result is 1, the calibration has failed. Refer to “**CALibration[1][:ALL]**” on page 105 for further information.

NOTE

The **CALibration[1][:ALL]** command is identical to the **CALibration[1][:ALL]?** query except that no number is returned to indicate the outcome of the calibration sequence. You can examine the **Questionable Status Register** or the error queue to determine if the calibration sequence has passed or failed. Refer to “**Status Reporting**” on page 57 for further information.

Making Measurements

The **MEASure?** query and **CONFigure** command provide a straightforward method to program the U2000 Series power sensor for measurements. You can select the measurement expected power level and resolution in one command. The U2000 Series power sensor automatically presets other measurement parameters to default values as shown in [Table 1-1](#) below. Refer to [“Auto-Averaging Settings”](#) on page 398 on how to select the corresponding parameters.

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 U2000 Series power sensor 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 U2000 Series power sensor. As shown in [Table 1-1](#), the **CONFigure** command presets various states in the U2000 Series power sensor. It may be likely that you do not want to preset these states. Refer to [“Using the Lower-Level Commands”](#) on page 34 for further information.

Using MEASure?

The simplest way to program the U2000 Series power sensor for measurements is by using the **MEASure?** query. However, this query does not offer much flexibility. When you execute the query, the U2000 Series power sensor 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 shows a few examples of how to use the **MEASure?** query to make a measurement. These examples configure the U2000 Series power sensor for a measurement (as described in each individual example), automatically place the U2000 Series power sensor in the “wait-for-trigger” state, internally trigger the U2000 Series power sensor to take one reading, and then sends the reading to the output buffer.

The following examples give an overview of the **MEASure?** query. For further information on the **MEASure?** query, refer to the “[Measurement Commands](#)” on page 86.

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 place holder.

The source list parameter is used to specify a measurement channel. The U2000 Series power sensors support 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.

specifies source list

↓
MEAS? DEF,DEF,(@1)

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 value entered determines which of the two ranges is used for the measurement. If the current setting of the U2000 Series power sensor range is no longer valid for the new measurement, specifying the expected power value decreases the time taken to obtain a result.

The following example uses the expected value parameter to specify a value of -50 dBm. This selects the U2000 Series power sensor lower range (refer to “**Range**” on page 47 for details of the range breaks). The resolution parameter is set to its default value, leaving it at its current setting.

specifies expected power value


↓
MEAS? -50,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 resolution. This parameter does not affect the resolution of the data, however it does affect the auto-averaging setting (refer to “**Auto-Averaging Settings**” on page 398).

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 85, 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**.

specifies resolution setting



MEAS? DEF, 3

Using the CONFigure Command

When you execute this command, the U2000 Series power sensor 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 U2000 Series power sensor configuration from the preset conditions. The U2000 Series power sensor offers a variety of low-level commands in the **SENSe**, **CALCulate**, and **TRIGger** subsystems. For example, if you want to change the measurement filter length, use the **[SENSe[1]:]AVERage:COUNT** command.

*Use the **INITiate** command or **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 show 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	<i>Resets the sensor.</i>
CONF	<i>Configures the measurement -sets to a single measurement by default.</i>
READ?	<i>Initiates and retrieves the measurement.</i>

Using INITiate and FETCh?

*RST	<i>Resets the sensor.</i>
CONF	<i>Configures the measurement -sets to a single measurement by default.</i>
INIT	<i>Sets it to wait for a trigger state.</i>
FETC?	<i>Triggers a measurement, then retrieves the measurement reading.</i>

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 place holder.

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	<i>Aborts the measurement.</i>
CONF DEF,DEF,(@1)	<i>Configures the measurement to make a measurement using the current expected power and resolution settings.</i>
READ?	<i>Initiates and retrieves the measurement.</i>

Using INITiate and FETCh?

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

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 value entered determines which two ranges are used for the measurement. If the current setting of the U2000 Series power sensor range is no longer valid for the new measurement, specifying the expected power value decreases the time taken to obtain a result.

The following example uses the expected value parameter to specify an expected power of -50 dBm. This selects the U2000 Series power sensor lower range (refer to “**Range**” on page 47 for details of the range breaks). The resolution parameter is set to its default value, leaving it at its current setting.

Using READ?

ABOR	<i>Aborts the measurement.</i>
CONF -50,DEF, (@1)	<i>Configures the measurement to use an expected power of -50 dBm and the current resolution setting.</i>
READ?	<i>Initiates and retrieves the measurement.</i>

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

```

1 ABOR
2 CONF -50,DEF, (@1)
3 SENS:AVER:COUN 1024
4 TRIG:DEL:AUTO OFF
5 READ?

```

Using INITiate and FETCh?

ABOR	<i>Aborts the measurement.</i>
CONF -50,DEF, (@1)	<i>Configures the measurement to use an expected power of -50 dBm and the current resolution setting.</i>
INIT	<i>Sets it to wait for a trigger state.</i>
FETC? -50,DEF, (@1)	<i>Triggers a measurement, then retrieves the measurement reading.</i>

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 -50,DEF, (@1)

```



```

3 SENS: AVER: COUN 1024
4 TRIG: DEL: AUTO OFF
5 INIT
6 FETC? -50, 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 (refer to [Figure 1-2](#) on page 44).

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	<i>Aborts the measurement.</i>
CONF DEF, 3	<i>Configures the measurement to use the current setting of the expected power and source list and a resolution setting of 3.</i>
READ?	<i>Initiates and retrieves the measurement.</i>

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

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:IMM
5 FETC? DEF,3

```

Using the Lower-Level Commands

An alternative method of making measurements is to use the lower-level commands to set up the expected range and resolution. This can be done using the following command:

```
[SENSe[1]:]POWer:AC:RANGe
```

The measurement type can be set using the following command in the **CALCulate** subsystem:

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

The advantage of using the lower-level commands over the **CONFigure** command is that they give you more precise control of the U2000 Series power sensor. As shown in [Table 1-1](#), the **CONFigure** command presets various states in the U2000 Series power sensor. It may be likely that you do not want to preset these states.

Example

The following example sets the expected power value to -50 dBm and the resolution setting to 3 using the lower-level commands.

ABOR	<i>Aborts the measurement.</i>
CALC:MATH:EXPR "(SENS1)"	<i>Sets to single measurement.</i>
POW:AC:RANG -50	<i>Sets lower range.</i>
INIT	<i>Sets it to wait for a trigger state.</i>
FETC?	<i>Triggers a measurement, then retrieves the measurement reading.</i>

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 U2000 Series power sensor is capable of storing 10 frequency-dependent offset tables of 80 frequency points each.

To use 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** Zeroing and calibrate the U2000 Series power sensor.
- 5** Specify the frequency of the signal you want to measure. The required offset is automatically set by the U2000 Series power sensor from the frequency-dependent offset table.
- 6** Make the measurement.

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

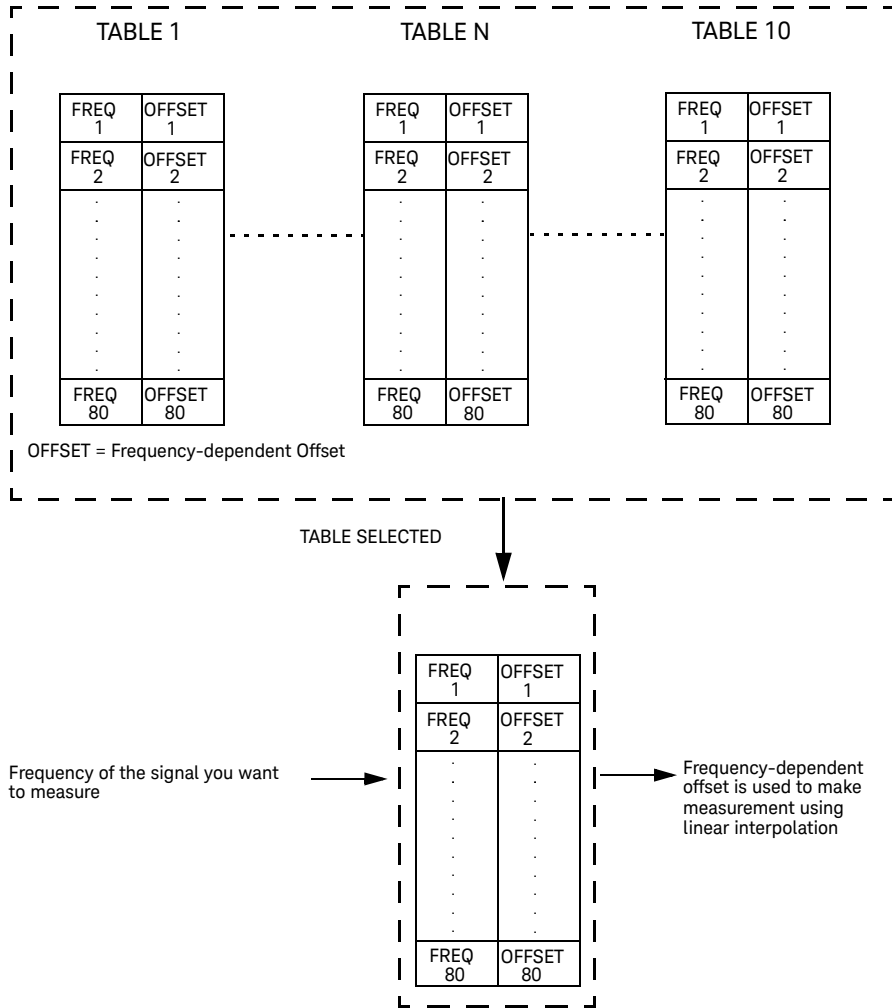


Figure 1-1 Frequency-dependent offset tables

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** subsystem. To do this:

- 1 Select one of the existing tables using:
MEMory:TABLE:SElect <string>
 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 <string>,<string>. 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 U2000 Series power sensor 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 3](#) for their specified ranges.

Any offset values entered into the table should exclude the effect of the sensor. Characterization of the test setup independently of the sensor allows the same table to be used with any sensor.

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 U2000 Series power sensor 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 U2000 Series power sensor, use the following query:

```
MEMory:CATalog:TABLE?
```

The U2000 Series power sensor 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:

- 1 Table names use a maximum of 12 characters.
- 2 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"
```

Select 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 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 U2000 Series power sensor for the frequency of the signal you want to measure. The U2000 Series power sensor 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:POWe:AC DEF,1,(@1)
CORR:CSET2:SEL "Offset1"
CORR:CSET2:STAT ON
FREQ 500KHZ
INIT:IMM
FETC?
```

READ? Example

```
ABOR
CONF:POW:AC DEF,2,(@1)
CORR:CSET2:SEL "Offset1"
CORR:CSET2:STAT ON
FREQ 500KHZ
READ?
```

NOTE

If the measurement frequency does not correspond directly to a frequency in the frequency-dependent offset table, the U2000 Series power sensor calculates the offset using linear interpolation.

If you enter a frequency outside the frequency range defined in the frequency-dependent offset table, then the U2000 Series power sensor 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 U2000 Series power sensor to make a measurement, use the query:

```
[SENSe[1]:]CORRection:GAIN4|FDOFfset[:INPut][:MAGNITUDE]?
```

The response may be an interpolated value.

Setting the Range and Averaging

This section provides an overview of setting the range, resolution, and averaging. For more detailed information about these features refer to the individual commands in [Chapter 7, “SENSe Subsystem”](#).

Averaging

The U2000 Series power sensor 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 U2000 Series power sensor to 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 U2000 Series power sensor 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 398 for more information.

[Figure 1-2](#), [Figure 1-3](#), and [Figure 1-4](#) illustrate the U2000 Series power sensor dynamic range hysteresis respectively.

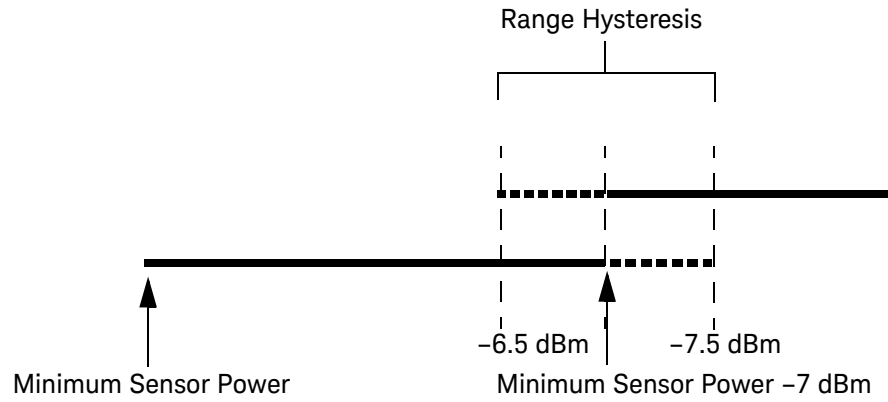


Figure 1-2 U2000/1/2/4A averaging range hysteresis

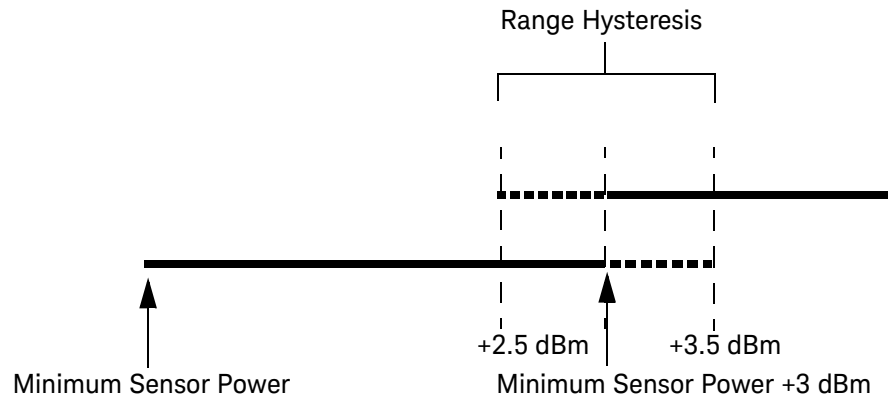


Figure 1-3 U2000/1/2H averaging range hysteresis

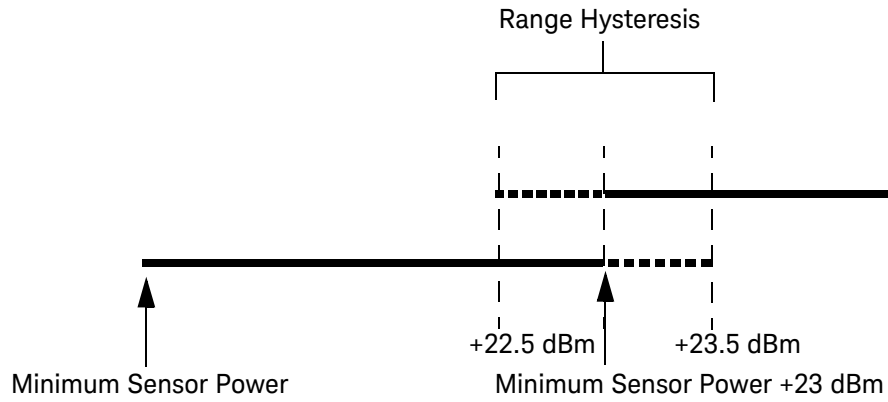


Figure 1-4 U2000/1B averaging range hysteresis

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 and x2 Speed

Number of Averages	Settling Time ^[a] (s) (Normal Speed)	Settling Time ^[a] (s) (x2 Speed)
1	0.045	0.042
2	0.09	0.05
4	0.17	0.09
8	0.34	0.17
16	0.66	0.34
32	1.3	0.66
64	2.6	1.3
128	5.2	2.6
256	10.4	5.2
512	20.9	10.4
1024	41.9	20.9

[a] Manual filter, 10 dB decreasing power step (not across the switching point)

Range

With the U2000 Series power sensors, the range can be set either automatically or manually. Use auto-ranging when you are not sure of the power level you will be measuring.

Setting the Range

To set the range manually use the following command:

```
[SENSE[1]:]POWER:AC:RANGE <numeric_value>
```

If the `<numeric_value>` is set to:

- 0, the sensor lower range is selected.
- 1, the sensor upper range is selected.

For details on the range limits of the U2000 Series power sensors, refer to the *U2000 Series USB Power Sensors Operating and Service Guide*.

For further information on this command refer to [page 223](#).

To enable auto-ranging use the following command:

```
[SENSE[1]:]POWER:AC:RANGE:AUTO ON
```

Use auto-ranging when you are not sure of the power level you will be measuring.

Setting Offsets

Channel Offsets

The U2000 Series power sensor 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 U2000 Series power sensor. 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 U2000 Series power sensor to detect any measurement that is outside the predefined upper and/or lower limit value.

Setting Limits

The U2000 Series power sensor 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.

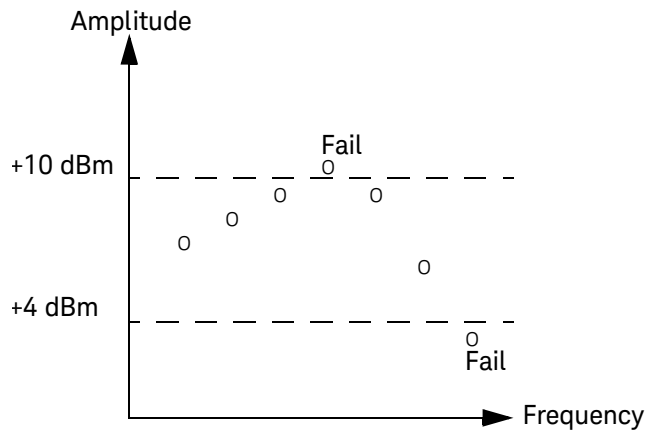


Figure 1-5 Limits checking results

The U2000 Series power sensor 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 depend on the measurement unit that is currently selected, refer to [Table 1-3](#).

Table 1-3 Range of values for measurement limits

Units			Default	
	Maximum	Minimum	Maximum	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 U2000 Series power sensor.
-> 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 U2000 Series power sensor 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 U2000 Series power sensor to log an error.

NOTE

“->” indicates the commands that you send to the U2000 Series power sensor.

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 U2000 Series power sensor.
-> 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 FCO counter.
-> CALC:LIM:CLE           // Clears the FCO counter of any limit
                          // failure.
```

Provides 5 dBm of RF power to the U2000 Series power sensor, followed by sending an external trigger signal to the U2000 Series power sensor.

```
-> CALC:LIM:FAIL?         // Checks for limit failures.
<- 0                      // No limit failure, where the measured
                          // power is within the range of >4dBm 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 U2000 Series power sensor, followed by sending an external trigger signal to the U2000 Series power sensor.

```
-> 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 U2000 Series power sensor, followed by
sending an external trigger signal to the U2000 Series power sensor.
-> 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 U2000 Series power sensor, followed by
sending an external trigger signal to the U2000 Series power sensor.
-> 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 U2000 Series power sensor.
 “<-” indicates the response from the U2000 Series power sensor.

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/s) of the U2000 Series power sensor.

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

- The selected measurement rate of either **NORMa1**, **DOUB1e**, 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, in **FAST** mode, there are other influences which are described in “Fast Mode” on page 55.

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

Measurement Rate

There are three possible speed settings: **NORMa1**, **DOUB1e**, and **FAST**. These are set using the **[SENSE[1]:]MRATE** command.

In **NORMa1** and **DOUB1e** modes, full sensor functionality is available. In **FAST** mode averaging, limits are disabled.

Refer to “Specifications” in the *U2000 Series USB Power Sensors Operating and Service Guide* to see the influence of these speed settings on the accuracy and noise performance of the U2000 Series power sensors.

Trigger Mode

The U2000 Series power sensor has a very flexible triggering system. For simplicity, it can be described as having three modes:

- Free Run: When the U2000 Series power sensor is in Free Run mode, it continuously takes measurements. A measurement is in free run when **INITiate[1]:CONTinuous** is set to **ON** and **TRIGger[1]:SOURce** is set to **IMMediate**.
- Triggered Free Run: When the U2000 Series power sensor is in Triggered Free Run or Continuous Trigger mode, it takes a new measurement each time a trigger event is detected. The U2000 Series power sensor is in Triggered Free Run/Continuous Trigger mode when **INITiate[1]:CONTinuous** is set to **ON** and **TRIGger[1]:SOURce** is not set to **IMMediate**.
- Single Shot: When the U2000 Series power sensor is in Single Shot mode, it takes a new measurement when a trigger event is detected and then returns to the idle state. The U2000 Series power sensor is in Single Shot mode when **INITiate[1]: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 326 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 **TRIGger[1]:DELay:AUTO** set to **ON**. Also, the **MEASure?** query operates in “trigger with delay” mode.

In “trigger with delay” mode, a measurement is not completed until the U2000 Series power sensor 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 approximately using the following equation:

$$\text{readings/s} = \text{speed (as set by [SENSE[1]:]MRATe)} / \text{filter length}$$

For example, with a filter length of 4 and **[SENSE[1]:]MRATe set to NORMa1**, approximately 5 readings/s is calculated by the U2000 Series power sensor.

Output Format

The U2000 Series power sensor 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 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 **FAST** mode as it reduces the amount of bus traffic.

Units

The U2000 Series power sensor can output results in either linear or log units. The internal units are linear, therefore optimal performance is achieved when the results are also in linear units (since the overhead of performing a log function is removed).

Command Used

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

In other trigger modes, there are a number of queries which can be used, for example, **MEASure?**, **READ?**, **FETCh?** Note that the **MEASure?** and **READ?** queries are compound commands as 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 **TRIGger[:SEquence[1]]:COUNT** must be set to return multiple measurements for each **FETCh** command. For average only measurements a count of four 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 U2000 Series power sensor, and to a certain extent, the volume of remote traffic. The latter can be reduced using the **FORMat REAL** command to return results in 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 1-6 shows how measurements are calculated. It shows the order in which the various U2000 Series power sensor functions are implemented in the measurement calculation.

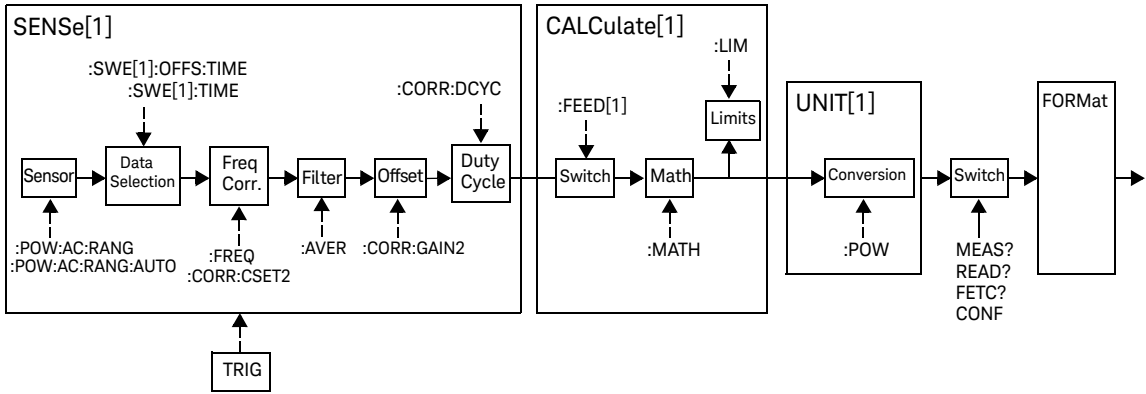


Figure 1-6 How measurements are calculated

Status Reporting

Status reporting is used to monitor the U2000 Series power sensor to determine when events have occurred. Status reporting is accomplished by configuring and reading status registers.

The U2000 Series power sensor 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 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 1-7](#) 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.

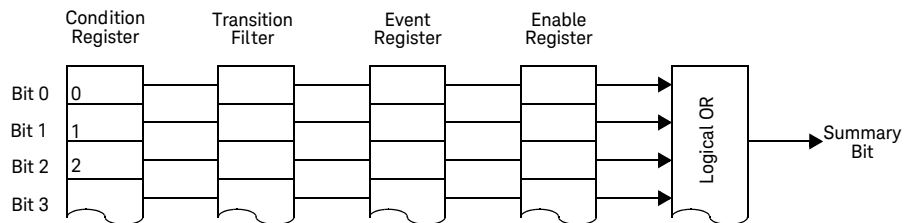


Figure 1-7 Generalized status register model

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 U2000 Series power sensor. 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, subsequent 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 1-8 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.

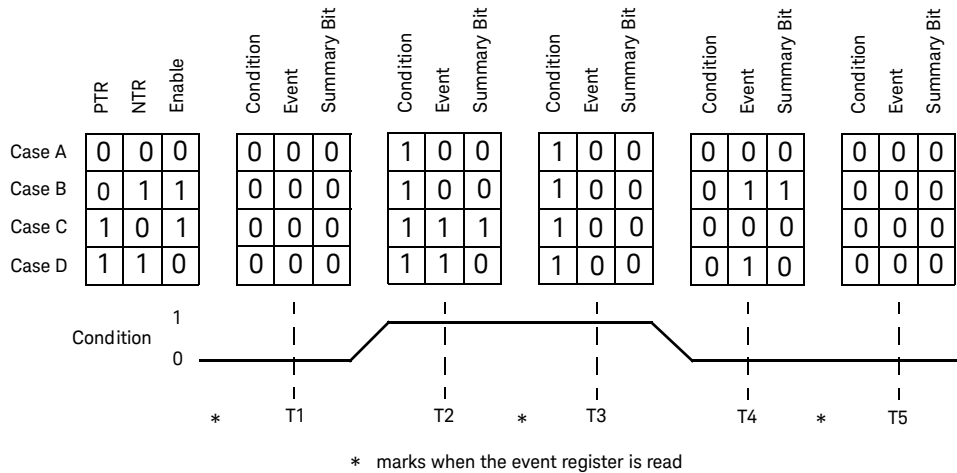


Figure 1-8 Typical status register bit changes

Consider **Case C**, where the positive transition filter is set to 1 and negative transition filter to 0. This configures the U2000 Series power sensor 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 U2000 Series power sensor 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 U2000 Series power sensor 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 U2000 Series power sensor is shown in [Figure 1-9](#). 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.

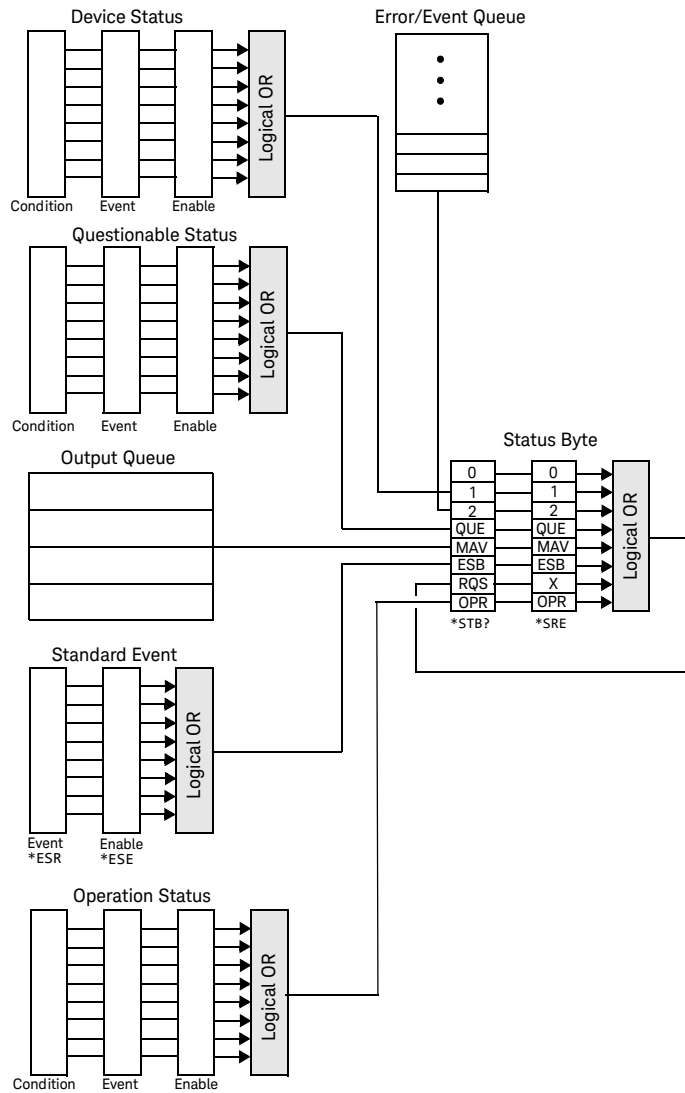


Figure 1-9 Status system

The Status Byte Summary Register

The status byte summary register reports conditions from other status registers. Query data waiting in the U2000 Series power sensor 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 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 enable register)
4	16	Data Available Data is available in the U2000 Series power sensor output buffer
5	32	Standard Event One or more bits are set in the Standard Event register (bits must be “enabled” in enable register)
6	64	Request Service The U2000 Series power sensor 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 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 sensor 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 U2000 Series power sensor 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.

The 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 a ***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 the Error Message List in Chapter SYSTem Subsystem
3	8	Device Error A device error occurred, refer to error numbers 310 to 350 in the Error Message List in Chapter SYSTem Subsystem
4	16	Execution Error An execution error occurred, refer to error numbers 211 to 231 in the Error Message List in Chapter SYSTem Subsystem

Bit Number	Decimal Value	Definition
5	32	Command Error A command syntax error occurred, refer to error numbers 101 to 161 in the Error Message List in Chapter SYSTem Subsystem
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 sensor power.
- execute a ***ESE 0** command.

Questionable Status Register

The questionable status register provides information about the quality of the U2000 Series power sensor 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 questionable status model is shown in the pullout at the end of this chapter.

The following bits in these registers are used by the U2000 Series power sensor.

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

Bit Number	Decimal Weight	Definition
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	<p>This is a summary bit for the Questionable POWER Register</p> <ul style="list-style-type: none"> - SET: Error -230, "Data corrupt or stale" - CLEARED: When no errors are detected by the U2000 Series power sensor during a measurement covering the causes given for it to set
8	CALibration Summary	<p>This is a summary bit for the Questionable CALibration Register</p> <ul style="list-style-type: none"> - SET: These may be caused by CALibration[1]:ZERO: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	<ul style="list-style-type: none"> - SET: This bit is set when the power on self-test fails - CLEARED: When the power on self-test passes

Operation Status

The Operation Status group monitors conditions in the U2000 Series power sensor measurement process.

The Operation status model is shown in the pullout at the end of this chapter.

The following bits in these registers are used by the U2000 Series power sensor:

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
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 <ul style="list-style-type: none"> – SET: At beginning of zeroing (CALibration[1]:ZERO:AUTO ONCE). Also for the compound command/query CALibration[1][:ALL]?, this bit is set when sensor zeroing begins. – CLEARED: At the end of zeroing or calibration
4	MEASuring	This is a summary bit for the Operation MEASuring Register <ul style="list-style-type: none"> – SET: When the U2000 Series power sensor is taking a measurement – CLEARED: When the measurement is finished
5	Waiting for TRIGger	This is a summary bit for the Operation TRIGger Register <ul style="list-style-type: none"> – SET: When the U2000 Series power sensor enters the “wait for trigger” state – CLEARED: When the U2000 Series power sensor enters the “idle” state
10	SENSe	This is a summary bit for the Operation SENSe Register <ul style="list-style-type: none"> – SET: When the U2000 Series power sensor is reading data from non-volatile memory – CLEARED: When the U2000 Series power sensor is not reading data from non-volatile memory
11	Lower Limit Fail	This is a summary bit for the Lower Limit Fail Register <ul style="list-style-type: none"> – SET: If a measurement is made and 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 <ul style="list-style-type: none"> – SET: If a measurement is made and 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 U2000 Series power sensor:

Table 1-10 Bit definitions - Device status register

Bit Number	Decimal Weight	Definition
0	-	Not used
1	2	Not used
2	4	Not used
3	8	U2000 Series power sensor error
4	16	Not used
5	32	Not used
6	64	Not used
14	16384	Not used

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

Table 1-11 Bit change conditions for Device status register

Bit Number	Definition	EVENTs Causing Bit Changes
3	U2000 Series power sensor error	<ul style="list-style-type: none"> - SET: If the U2000 Series power sensor non-volatile memory has failed or other hardwares have failed - CLEARED: In every other condition

Using the Operation Complete Commands

***OPC?** and ***OPC** allow you to maintain synchronization between the PC and the U2000 Series power sensor. ***OPC?** places a 1 into the U2000 Series power sensor output queue when all pending U2000 Series power sensor commands are complete. 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 U2000 Series power sensor operations are complete.

Procedure

- Send a Device Clear message to clear the U2000 Series power sensor output buffer.
- Clear the event registers with the ***CLS** (clear status) command.
- Enable operation complete using the ***ESE 1** command (standard event register).
- Send ***OPC?** (operation complete query) to assure synchronization.
- Send your programming command string, and place the ***OPC** (operation complete) command as the last command.
- 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 U2000 Series power sensor has finished calibrating.

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

Saving and Recalling U2000 Series Power Sensor Configurations

To reduce repeated programming, up to 10 U2000 Series power sensor configurations can be stored in the non-volatile memory. The error list, zeroing and calibration information are not stored.

How to Save and Recall a Configuration

The U2000 Series power sensor 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 U2000 Series power sensor returns to the trigger “idle state”.
- The U2000 Series power sensor input and output buffers are cleared.
- The U2000 Series power sensor is prepared to accept a new command string.

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.

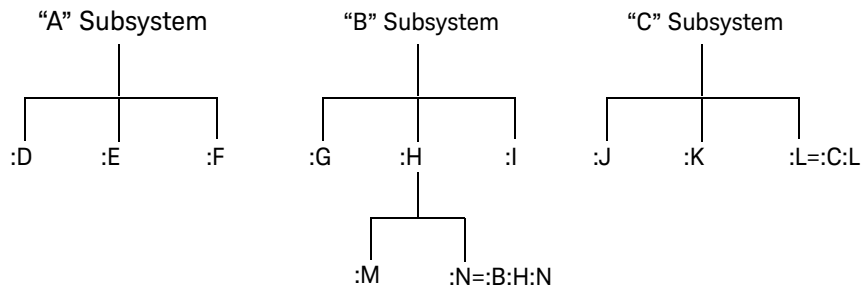


Figure 1-10 Hierarchical structure of SCPI

Mnemonic Forms

Each keyword has both a long 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 uppercase characters, and the rest of the keyword shown in lowercase 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.

```
CALibration[1]:ZERO:TYPE EXTerna1;CALibration[1]:ZERO:AUTO ONCE  
CALibration[1]:ZERO:TYPE EXTerna1;ZERO:AUTO ONCE
```

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 queries (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 queries 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 without reading the response. When you cannot avoid this situation, send a Device Clear before sending the second query.

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 U2000 Series power sensor.

Syntax Conventions

Throughout this guide, the following conventions are used for 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 by passing 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). START and R6_5F are examples of character data. The first character must be an alphanumeric, followed by either alphanumeric or underscore characters up to a maximum of 12 characters.

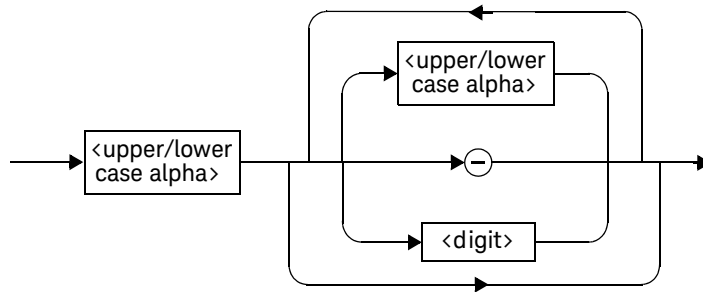


Figure 1-11 Format of **<character_data>**

<NAN> Definition

Not a number (NAN) is represented as 9.91 E37. 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 10 (that is, hexadecimal, octal, and binary). Examples of non-decimal numeric include #HA2F, #ha4e, #Q62, #q15, and #B01011. [Figure 1-12](#) shows the non-decimal numeric standard data structures.

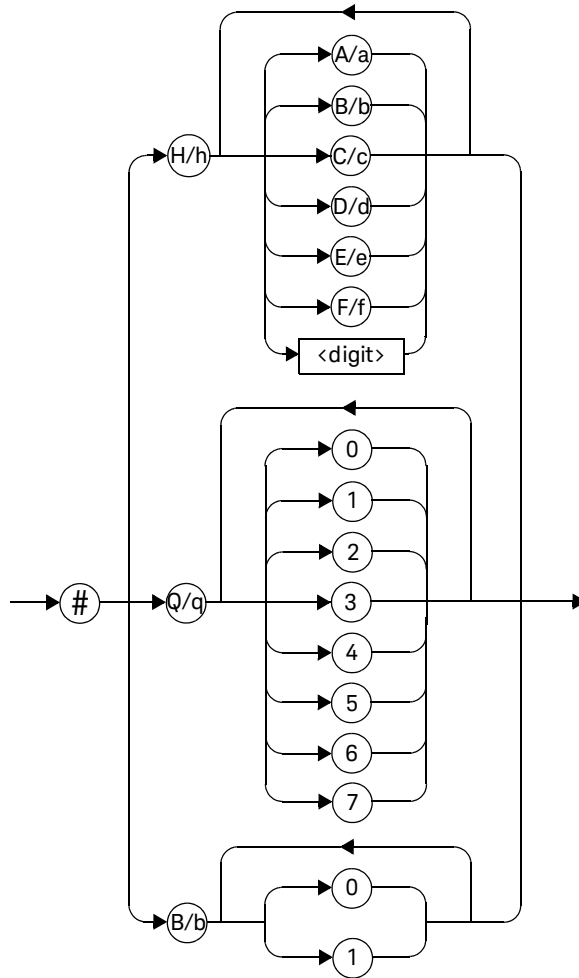


Figure 1-12 Format of <non-decimal numeric>

<NRf> Definition

Throughout this document, <NRf> is used to denote a flexible numeric representation, for example, +200; -56; +9.9E36.

<NR1> Definition

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

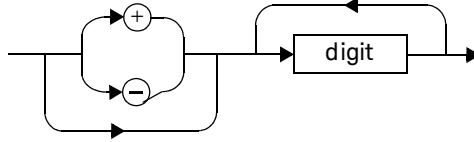


Figure 1-13 Format of <NR1>

The following shows the examples of <NR1>:

- 146
- +146
- -12345

<NR2> Definition

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

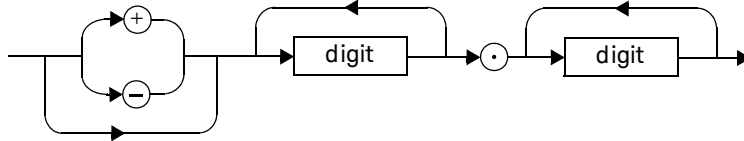


Figure 1-14 Format of <NR2>

The following shows the examples of <NR2>:

- 12.3
- +1.2345
- -0.123

<NR3> Definition

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

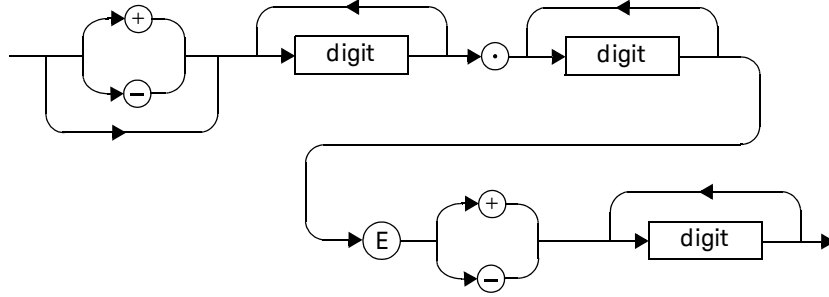


Figure 1-15 Format of <NR3>

The following shows the examples of <NR3>:

- 1.23E+6
- 123.4E-54
- -1234.567E+90

<numeric_value> Definition

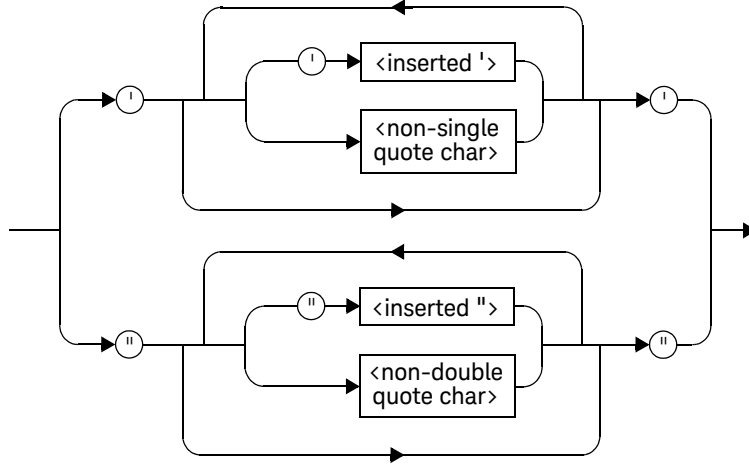
Throughout this document, the decimal numeric element is abbreviated to <numeric value>, for example, <NRf>, MINimum, MAXimum, DEFault, or Not A Number (NAN).

<string> Definition

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

The format is defined as:

Program Data



Response Data

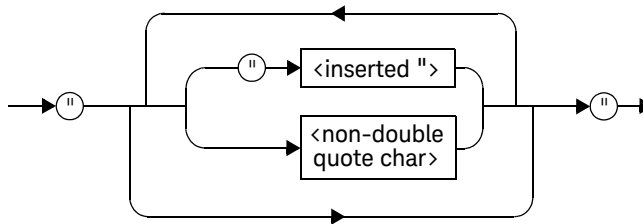


Figure 1-16 Format of <string>

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.

SCPI Compliance Information

The U2000 Series power sensor complies with the rules and regulations of the present version of SCPI (Standard Commands for Programmable Instruments). You can determine the SCPI version which the U2000 Series power sensor complies with by sending the **SYSTem:VERSion?** query from the remote interface.

The following commands are device-specific to the U2000 Series power sensor. They are not included in the 1999.0 version of the SCPI standard. However, these commands are designed with the SCPI format in mind and they follow all of the syntax rules of the standard.

```

MEMory:CLEar[:NAME]
MEMory:TABLE:SElect
MEMory:STATE:DEFine
MEMory:TABLE:GAIN[:MAGNitude]
MEMory:TABLE:GAIN:POINTs?
MEMory:TABLE:MOVE
[SENSe[1]:]AVERage:SDETect
[SENSe[1]:]CORRection:FDOFfset
[SENSe[1]:]MRATe
[SENSe[1]:]POWer:AC:RANGe
SERVice:SENSor[1]:CDATe?
SERVice:SENSor[1]:CPLace?
SERVice:SENSor[1]:SNUMber?
SERVice:SENSor[1]:TYPE?

```

Summary of Commands

For the details of each SCPI (Standard Commands for Programmable Instruments) command for the U2000 Series power sensor, refer to later chapters for more details on each command.

All the commands listed have queries unless otherwise stated in the “Notes” column.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

2 Measurement Commands

Measurement Commands	86
CONFigure[1]?	89
CONFigure[1] Command	90
CONFigure[1][:SCALar][:POWer:AC] [<expected_value>[,<resolution>[,<source list>]]]	91
FETCh[1]? Query	93
FETCh[1][:SCALar][:POWer:AC]? [<expected_value>[,<resolution>[,<source list>]]]	94
READ[1] Query	96
READ[1][:SCALar][:POWer:AC]? [<expected_value>[,<resolution>[,<source list>]]]	97
MEASure[1] Query	100
MEASure[1][:SCALar][:POWer:AC]? [<expected_value>[,<resolution>[,<source list>]]]	101

This chapter explains how to use the **MEASure** group of commands 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	Descriptions
MEASure?	Provides the simplest way to program a U2000 Series power sensor 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 U2000 Series power sensor 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 311.

CONFigure, **FETCh?**, **READ?**, and **MEASure?** all have a numeric suffix which refers to a specific measurement window. The U2000 Series power sensors do not have the measurement window feature, so this suffix is always 1. [Figure 2-1](#) shows an example of the configuration returned measurement result.

CONFigure1?
current measurement

Figure 2-1 Measurement display CALCulate block channel

Optional Parameters

CONFigure, **FETCh**?, **READ**?, and **MEAS**ure? have the following three optional parameters:

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

Refer to “[Auto-Averaging Settings](#)” on page 398 to configure the correct parameters for expected power and resolution.

Expected Power Value

An **<expected_value>** parameter is only required if you are using a U2000 Series power sensor. The value entered determines which of the U2000 Series power sensor two ranges is used for the measurement. If the current setting of the U2000 Series power sensor range is no longer valid for the new measurement, specifying the expected power value decreases the time taken to obtain a result.

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.

Summary of CONFigure, FETCh?, READ?, and MEASure?

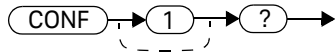
The following commands are described in this chapter:

Keyword	Parameter Form	Notes	Page
CONFigure[1]?		[query only]	page 89
CONFigure[1]			
[:SCALar]			
[:POWer:AC]	[<expected_value> [,<resolution>[,<source list>]]]	[no query]	page 91
FETCh[1]			
[:SCALar]			
[:POWer:AC]?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	page 94
READ[1]			
[:SCALar]			
[:POWer:AC]?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	page 97
MEASure[1]			
[:SCALar]			
[:POWer:AC]?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	page 101

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 U2000 Series power sensor 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 U2000 Series power sensor to make one trigger cycle when INITiate is sent
TRIGger[1]:SOURce IMMEDIATE	When TRIG:SOUR is set to BUS or HOLD , sets the U2000 Series power sensor 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

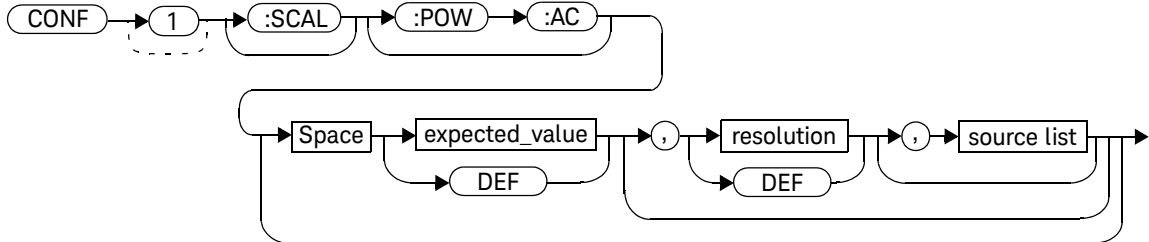
```
CONFigure[1][:SCALar][:POWer:AC]
[<expected_value>[,<resolution>[,<source list>]]]
```

This command is used to set:

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

Refer to “[Auto-Averaging Settings](#)” on page 398 to configure the correct parameters for expected power and resolution.

Syntax



Parameters

Refer to “Optional Parameters” on page 87 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 U2000 Series power sensors support 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 place holder. 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 of 1, 0.1, 0.01 and 0.001 respectively.

Example

```
CONF:POW:AC DEF,2,(@1)
```

This command configures the measurement to measure power using the current sensor 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 133 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 U2000 Series power sensor is in the idle state and the data is invalid, or the U2000 Series power sensor 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 U2000 Series power sensor has hung.

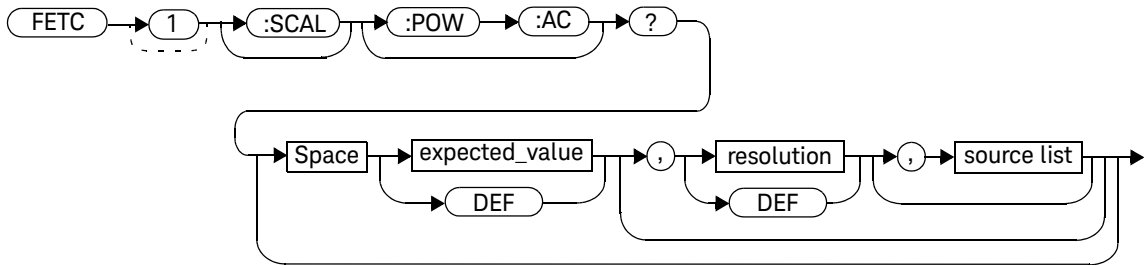
To unlock the U2000 Series power sensor and adjust trigger settings, a **Device Clear** command must be performed.

`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 398 to configure the correct parameters for expected power and resolution.

Syntax



Parameters

Refer to “Optional Parameters” on page 87 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] DEF ^[a] 1.0, 0.1, 0.01, 0.001
source list	The measurement channel which the command is implemented on. The U2000 Series power sensors support 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 place holder. 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 of 1, 0.1, 0.01 and 0.001 respectively.

Example

FETC:POW:AC?

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 setting on the measurement, 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 133 for further information.

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

```
INITiate  
FETCh?
```



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

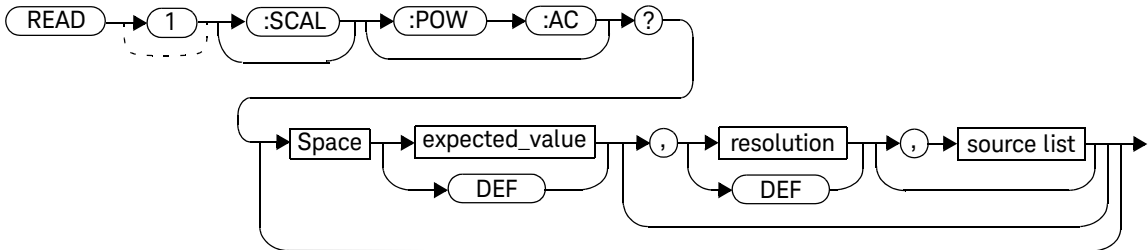
This query sets the measurement function, aborts then initiates the measurement, calculates the measurement result, and places the result on the bus. The result is a power based measurement and is expressed in the units defined by **UNIT[1]:POWer**.

Refer to “**Auto-Averaging Settings**” on page 398 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 87 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 CONF i gure 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 U2000 Series power sensors support 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 place holder. 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 of 1, 0.1, 0.01 and 0.001 respectively.

Example

READ:POW:AC?

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.

MEASure[1] Query

The **MEASure?** query configures the U2000 Series power sensor to perform a power measurement with the given measurement function, range and resolution then makes the measurement. The format of the result is set by **FORMat[:READing][:DATA]**. Refer to [Chapter 5, “FORMat Subsystem,”](#) on page 133 for further information.

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

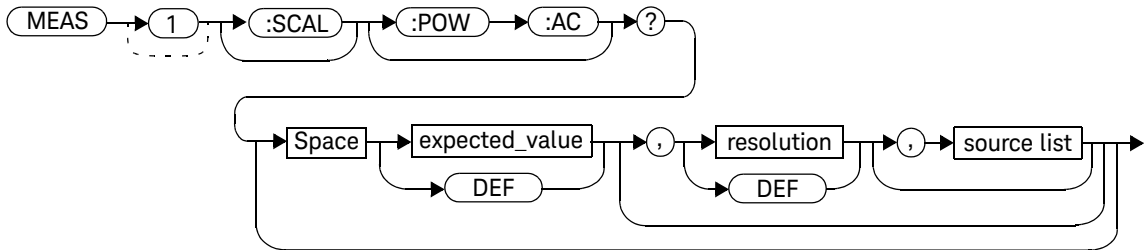
```
CONFigure  
READ?
```

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

This query aborts any measurement in progress, configures the U2000 Series power sensor, calculates the measurement result, and places the result on the bus.

Refer to **“Auto-Averaging Settings”** on page 398 to configure the correct parameters for expected power and resolution.

Syntax



Parameters

Refer to “Optional Parameters” on page 87 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 U2000 Series power sensors support 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 place holder. 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 of 1, 0.1, 0.01 and 0.001 respectively.

Example

```
MEAS:POW:AC? -70DBM,1,(@1)
```

Queries the measurement of the sensor, using an expected power level of -70 dBm and a resolution setting of 1.

3 CALibration Subsystem

CALibration Subsystem	104
CALibration[1][:ALL]	105
CALibration[1][:ALL]?	106
CALibration[1]:ZERO:AUTO [ONCE ON OFF 0 1]	107
CALibration[1]:ZERO:TYPE <EXTernal INTernal>	109

This chapter explains how the **CALibration** subsystem is used to perform zeroing on the U2000 Series power sensors.

CALibration Subsystem

The **CALibration** command subsystem is used to perform zeroing on the U2000 Series power sensor.

Zeroing of the U2000 Series power sensor is automatically performed:

- when a 5 °C change in temperature occurs
- when the connection to the U2000 Series power sensor is established
- every 24 hours

Refer to [Chapter 1, “Zeroing and Calibrating the U2000 Series USB Power Sensors”](#) on page 24 for more zeroing information.

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

- **CAL:ALL**
- **CAL:ZERO:AUTO**

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

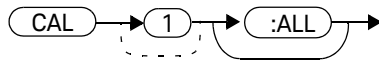
Keyword	Parameter Form	Notes	Page
CALibration[1]			
[:ALL]		[event; no query]	page 105
[:ALL]?		[event; query]	page 106
:ZERO			
:AUTO	<boolean> ONCE		page 107
:TYPE	<EXTernal INTernal>		page 109

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.

Syntax



Example

CAL:ALL

This command causes the U2000 Series power sensor to perform a zeroing.

Error Messages

If zeroing was not carried out successfully, error -231, "Data Questionable; ZERO ERROR" occurs.

CALibration[1][:ALL]?

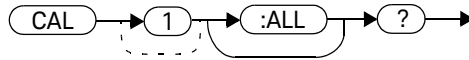
NOTE

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

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

Syntax



Query Example

CAL?

This query causes the U2000 Series power sensor to perform a zeroing and return a result.

Error Messages

If zeroing was not carried out successfully, error -231, "Data Questionable; ZERO ERROR" occurs.

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

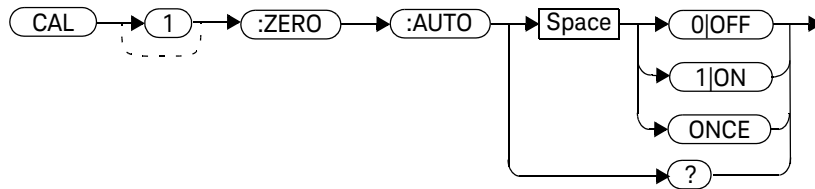
This command performs zeroing on the U2000 Series power sensor.

When **ONCE** is enabled, the zeroing will be performed only once for both internal and external zeroing types.

When **1|ON** is set, zeroing is performed if the sensor temperature changes by ± 5 °C or the time since zeroing was last performed is greater than 24 hours.

The **0|OFF** parameter disables the auto-zeroing feature.

Syntax



Example

CAL:ZERO:AUTO ONCE

This command causes the U2000 Series power sensor to perform a zeroing routine.

Reset Condition

Automatic zeroing is enabled on reset.

3 CALibration Subsystem

Query

CALibration[1]:ZERO:AUTO?

The query always returns a value of 0 when the event is successful.

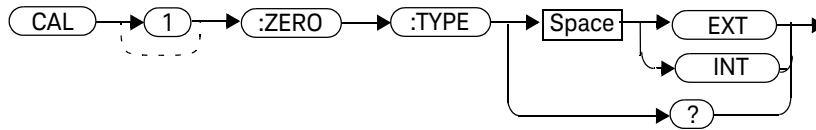
Error Messages

If zeroing was not carried out successfully, error -231, "Data Questionable; ZERO ERROR" occurs.

CALibration[1]:ZERO:TYPE <EXTernal||INTernal>

This command is used to configure the U2000 Series power sensor either for external zeroing or internal zeroing.

Syntax



Example

CAL:ZERO:TYPE EXT

This command changes the type of zeroing to external.

Reset Condition

The zeroing type is not affected by a reset.

Query

CALibration[1]:ZERO:TYPE?

This query returns the current zeroing type, either “INT” or “EXT”.

Query Example

CAL:ZERO:TYPE?

Queries the type of zeroing for the sensor.

Error Messages

This command is able to configure the zeroing type to “EXT” and “INT” only. Error -224, "Illegal parameter value" occurs for any other value.

4 CALCulate Subsystem

CALCulate Subsystem	112
CALCulate[1]:FEED[1]<"string">	114
CALCulate[1]:LIMit Commands	116
CALCulate[1]:LIMit:CLEar:AUTO <boolean> ONCE	117
CALCulate[1]:LIMit:CLEar[:IMMediate]	119
CALCulate[1]:LIMit:FAIL?	120
CALCulate[1]:LIMit:FCOunt?	121
CALCulate[1]:LIMit:LOWer[:DATA] <numeric_value>	123
CALCulate[1]:LIMit:UPPer[:DATA] <numeric_value>	125
CALCulate[1]:LIMit:STATe <boolean>	127
CALCulate[1]:MATH Commands	129
CALCulate[1]:MATH[:EXPRession] <"string">	130
CALCulate[1]:MATH[:EXPRession]:CATalog?	132

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

CALCulate Subsystem

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

There is an independent **CALCulate** block in the U2000 Series power sensor, as shown in [Figure 4-1](#).

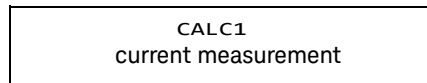


Figure 4-1 Measurement display CALCulate block channel

[Figure 4-2](#) details where the commands are applied within the **CALCulate** block.

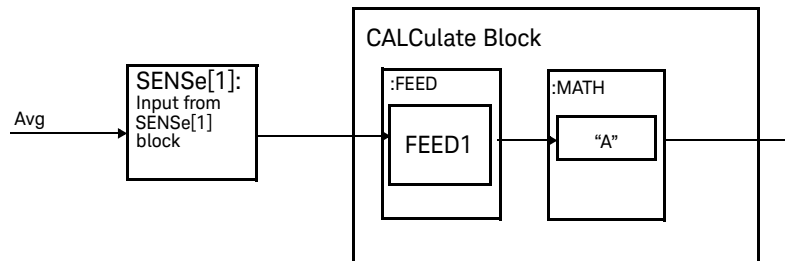


Figure 4-2 CALCulate block

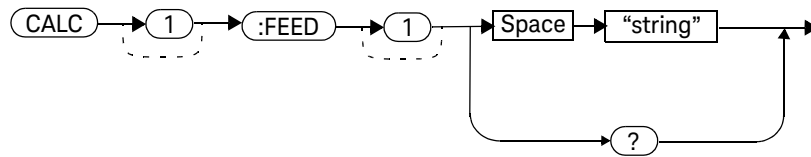
Keyword	Parameter Form	Notes	Page
CALCulate[1]			
:FEED[1]	<"string">		page 114
:LIMit			
:CLEAr			
:AUTO	<boolean> ONCE		page 117
[:IMMediate]			page 119
:FAIL?		[query only]	page 120
:FCOunt?		[query only]	page 121
:LOWer			
[:DATA]	<numeric_value>		page 123
:UPPer			
[:DATA]	<numeric_value>		page 125
:STATe	<boolean>		page 127
:MATH			
[:EXPRession]	<"string">		page 130
:CATalog?		[query only]	page 132

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 130 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). Values may be followed by ON SWEEP to specify the gate to be used for the feed, for example, " POW:AVER ON SWEEP ".	POW:AVER

Example

CALC:FEED "POW:AVER"

This command selects the input for FEED1 of CALC block CALC1 to be average power, using gate 1. 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 FEED1.

Error Messages

- If the command is used when no sensor is connected, error –241, “Hardware missing” occurs.
- If the <string> contains “ON SWEEP” and the feed’s **TRIGger:SOURCE** is not set to **INTernal** or **EXTernal**, error –221, “Settings conflict” occurs.
- If the command is used when U2004A model is connected, error –241, “Hardware missing” occurs.

CALCulate[1]:LIMit Commands

These commands set the measurement limits which enable you to:

- Set upper 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>

CALCulate[1]:LIMit:CLEar[IMMediate]

CALCulate[1]:LIMit:FAIL?

CALCulate[1]:LIMit:FCOunt?

CALCulate[1]:LIMit:LOWer[:DATA]

CALCulate[1]:LIMit:UPPer[:DATA]

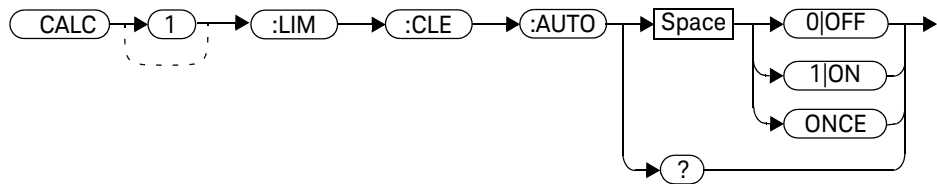
CALCulate[1]:LIMit:STATe <boolean>

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 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

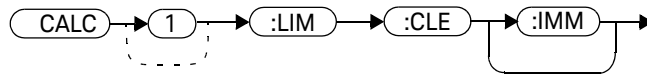
CALC:LIM:CLE:AUTO?

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

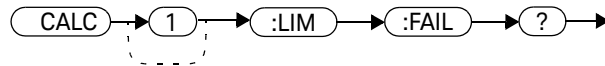
This command clears the FCO.

CALCulate[1]:LIMit:FAIL?

This query enters a 1 or 0 into the output buffer indicating whether there have been any limit failure. 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:

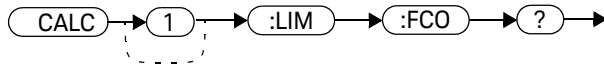
- measured using **MEASure?**
- read using **READ?**
- initiated using:
 - **INITiate[1][:IMMediate]** or,
 - **INITiate[1]:CONTinuous ON**

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?`

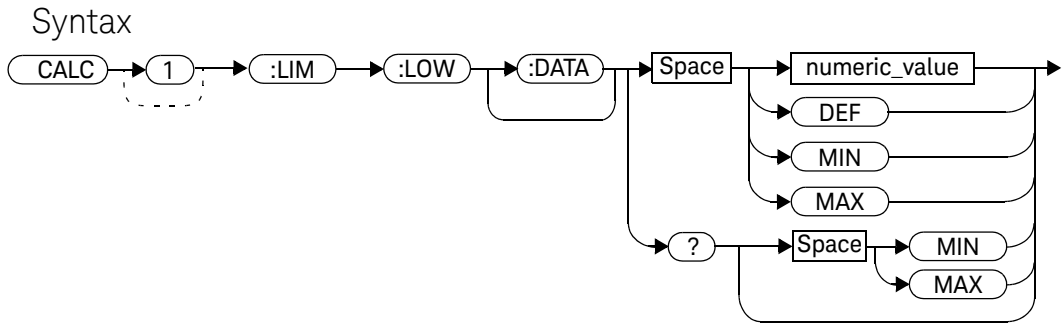
Queries the number of limit failures.

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.



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

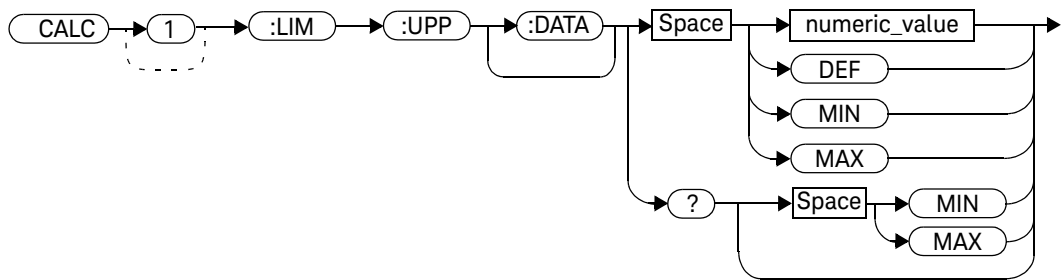
CALC:LIM:LOW?

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]

Query Example

CALC:LIM:UPP?

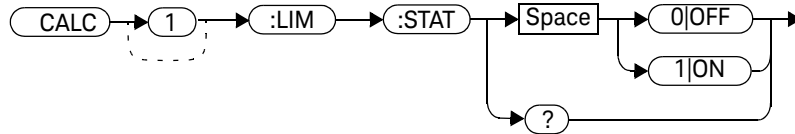
Queries the setting of the upper limit.

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

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

Query Example

CALC:LIM:STAT?

Queries whether the limit checking function is 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 **110**, 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:

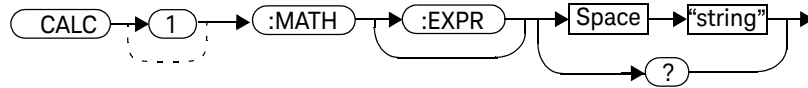
```
CALCulate[1]:MATH[:EXPRession] <"string">
```

```
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]

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

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?
```

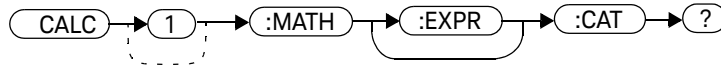
Queries the current setting of the math expression.

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 single measurement, the string is:
"(SENS1)"

Syntax



Example

CALC:MATH:CAT?

This query lists all the defined math expressions.

5 FORMat Subsystem

FORMat Subsystem	134
FORMat[:READings]:BORDER <character_data>	135
FORMat[:READings][:DATA] <character_data>	137

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

FORMat Subsystem

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

The queries affected are:

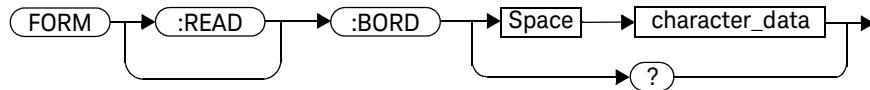
- FETCh?
- READ?
- MEASure?

Keyword	Parameter Form	Notes	Page
FORMat			
[:READings]			
:BORDer	<character_data>		page 135
[:DATA]	<character_data>		page 137

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: - NORMa1 - SWAPped	NORMa1 SWAPped

Example

FORM:BORDER SWAP

This command sets the byte order to swapped.

Reset Condition

On reset, this value is set to **NORMa1**.

Query

FORMat[:READings]:BORDER?

The query returns the current setting of the byte order. The format of the response is **NORMa1** or **SWAPped**.

Query Example

FORM:BORD?

Queries the current byte order setting.

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

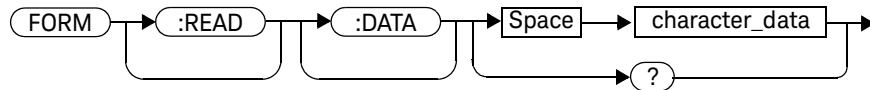
This command sets the data format for transferring numeric information to either **ASCIi** or **REAL**:

- 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.

NOTE

FORMat data formatting is not affected by TRACe subsystem data formatting.

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: either **ASCIi** or **REAL**.

Query Example

FORM?

Queries the current format setting.

6 MEMory Subsystem

MEMory Subsystem	140
MEMory:CATalog Queries	142
MEMory:CATalog[:ALL]?	143
MEMory:CATalog:STATe?	145
MEMory:CATalog:TABLE?	146
MEMory:CLEar Commands	148
MEMory:CLEar[:NAME] <“character_data”>	149
MEMory:CLEar:TABLE	151
MEMory:FREE Queries	152
MEMory:FREE[:ALL]?	153
MEMory:FREE:STATe?	154
MEMory:FREE:TABLE?	155
MEMory:NSTATes?	156
MEMory:STATe Commands	157
MEMory:STATe:CATalog?	158
MEMory:STATe:DEFine <“character_data”>,<numeric_value>	159
MEMory:TABLE Commands	161
MEMory:TABLE:FREQuency <numeric_value>{,<numeric_value>}	162
MEMory:TABLE:FREQuency:POINts?	165
MEMory:TABLE:GAIN[:MAGNitude] <numeric_value>{,<numeric_value>}	166
MEMory:TABLE:GAIN[:MAGNitude]:POINts?	168
MEMory:TABLE:MOVE <“character_data”>,<“character_data”>	169
MEMory:TABLE:SELEct <“character_data”>	171

This chapter explains how the **MEMory** subsystem is used to configure sensor frequency-dependent offset tables and save/recall registers.

MEMory Subsystem

The **MEMory** subsystem is used to:

- Edit and review sensor frequency-dependent offset tables.
- Store sensor frequency-dependent offset tables.
- Edit and review sensor save/recall registers.

Stored tables remain in the U2000 Series power sensor memory during power down. The U2000 Series power sensor is capable of storing 10 frequency-dependent offset tables of 80 frequency points each.

Keyword	Parameter Form	Notes	Page
MEMory			
:CATalog			
[:ALL]?		[query only]	page 143
:STATe?		[query only]	page 145
:TABLe?		[query only]	page 146
:CLEar			
[:NAME]	<"character_data">	[no query], [non-SCPI]	page 149
:TABLe		[no query]	page 151
:FREE			
[:ALL]?		[query only]	page 153
:STATe?		[query only]	page 154
:TABLe?		[query only]	page 155
:NSTates?		[query only]	page 156
:STATE			
:CATalog?		[query only]	page 158
:DEFine	<"character_data"> [, <numeric_value>]	[non-SCPI]	page 159
:TABLe			

Keyword	Parameter Form	Notes	Page
:FREQuency	<numeric_value> [,<numeric_value>]		page 162
:POINts?		[query only]	page 165
:GAIN			
[:MAGNitude]	<numeric_value> [,<numeric_value>]	[non-SCPI]	page 166
:POINts?		[query only], [non-SCPI]	page 168
:MOVE	<"character_data">, <"character_data">	[no query], [non-SCPI]	page 169
:SElect	<"character_data">	[no query], [non-SCPI]	page 171

MEMory:CATalog Queries

This group is used to query information on the current contents of a U2000 Series power sensor:

- Frequency-dependent offset tables
- Save/recall registers

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 and save/recall registers.

The U2000 Series power sensor 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>}

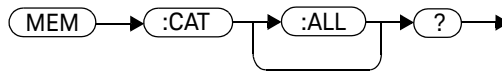
- The first numeric parameter indicates the amount of memory, in bytes, used for the storage of tables and registers.
- The second numeric parameter indicates the memory, in bytes, available for the storage of tables and registers.
- Each string parameter returned indicates the name, type and size of 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, 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" .....
```

There are also 10 frequency-dependent offset tables named **CUSTOM_A** through **CUSTOM_J** which do not contain any data when the U2000 Series power sensor is shipped from the factory.

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 U2000 Series power sensor 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 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.

The U2000 Series power sensor returns the data in the form of two numeric parameters and as many strings as there are stored tables.

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

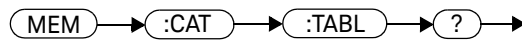
- The first numeric parameter indicates the amount of memory, in bytes, used for the storage of tables.
- The second parameter indicates the memory, in bytes, available for the storage of tables.
- Each string parameter returned indicates the name, type and size of a stored table:
 - **<string>,<type>,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

There are also 10 frequency-dependent offset tables named **CUSTOM_A** through **CUSTOM_J** which do not contain any data when the U2000 Series power sensor is shipped from the factory.

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 and save/recall registers. This 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, or save/recall register.

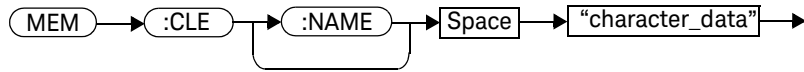
Although the table remains, a **MEMory:TABLE:FREQUENCY|GAIN:POINTS?** query returns a 0 as there are no contents in the table.

For frequency-dependent offset tables, 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



Parameters

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 Messages

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?** 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:TABLE

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 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, and save/recall registers. The format of the response is:

<bytes_available>,<bytes_in_use>

Syntax



Example

MEM:FREE?

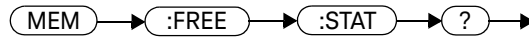
Queries the amount of memory free for frequency-dependent offset tables and save/recall register.

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?`

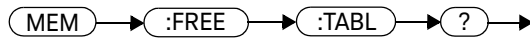
Queries the amount of free memory for save/recall registers.

MEMory:FREE:TABLE?

This query returns the amount of memory free for frequency-dependent offset tables. The format of the response is:

<bytes_available>,<bytes_in_use>

Syntax



Example

MEM:FREE:TABLE?

Queries the amount of free memory for tables.

MEMory:NStates?

This query returns the number of registers that are 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: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?

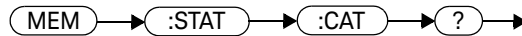
MEMory:STATe:DEFine

MEMory:STATe:CATalog?

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

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

Syntax



Example

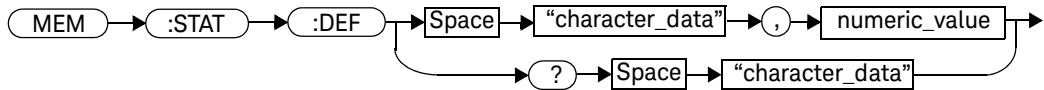
`MEM:STAT:CAT?`

Queries the 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 SETUP1.

Query

MEMory:STATe:DEFine? <"string">

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, 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: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 ascending order. Entries in the corresponding frequency lists as shown in [Table 6-1](#) with entries in the offset factor lists.

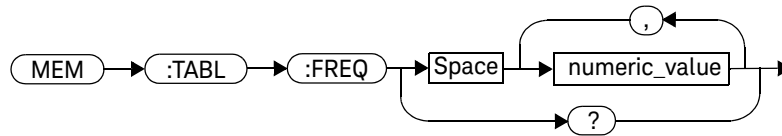
Table 6-1 Frequency and offset factor list

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

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 U2000 Series power sensor uses the highest or lowest point in the table to calculate the calibration factor or offset.

Depending on the available memory, the U2000 Series power sensor is capable of storing up to 10 frequency-dependent offset tables, each containing 80 points.

Syntax



Parameters

Item	Description/Default	Range of Values
numeric_value	A numeric value for the frequency. The default units are Hz.	1 kHz to 1000.0 GHz ^{[a][b]}

[a] The following measurement units can be used:

- Hz
- kHz (103)
- MHz (106)
- GHz (109)

[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 are in the list, error -108, "Parameter not allowed" occurs.
- If the frequencies are not entered in 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 which is 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 0. If no table is selected, this query returns **NAN**.

Syntax



Example

MEM:TABLE: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.

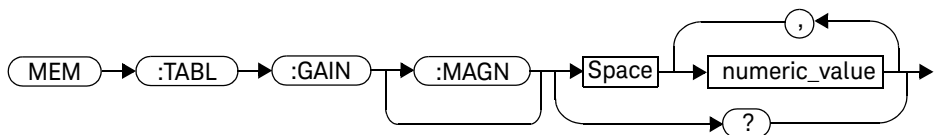
A maximum of 80 parameters for frequency-dependent offset tables can be sent with this command.

Entries in the frequency lists correspond as shown in [Table 6-2](#) with entries in the offset factor lists.

Table 6-2 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 calibration/ offset factors. The units are PCT.	1.0 to 150.0

Example

```
MEM:TABLE:SEL "Sensor_1"
MEM:TABLE:GAIN 97,99.5,97.4
```

This command enters a reference offset factor of 97%, 99.5% and 97.4% into the sensor 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:TABLE:GAIN?
```

Queries the offset in the current table.

Error Messages

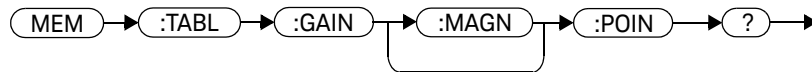
- If more than 80 offsets for the frequency-dependent offset tables are in the list, error -108, "Parameter not allowed" occurs.
- If a table is not specified using the **MEMory:TABLE:SElect** command, the data cannot be entered and error -221, "Settings conflict" occurs.
- If any of the offset factors 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

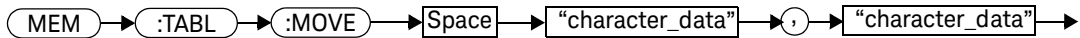
MEM:TABL:GAIN:POIN?

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.

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:TABLE:MOVE "tab1","tab1a" *This command renames a table named tab1 to tab1a.*

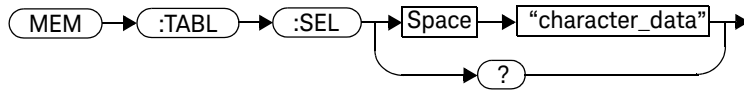
Error Messages

- If either table names are invalid, error -224, "Illegal parameter value" occurs.
- If the first parameter does not match an existing table name, the error -256, "File name not found" occurs.
- If the second parameter matches an existing table name or 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 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:TABLE: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.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

7

SENSe Subsystem

[SENSe] Subsystem	175
[SENSe[1]:]AVERage Commands	177
[SENSe[1]:]AVERage:COUNT:AUTO <boolean>	180
[SENSe[1]:]AVERage:SDETect <boolean>	183
[SENSe[1]:]AVERage[:]STATe <boolean>	185
[SENSe[1]:]BUFFer:COUNT <numeric_value>	187
[SENSe[1]:]CORRection:CSET2 Commands	189
[SENSe[1]:]CORRection:CSET2[:]SElect <"string">	190
[SENSe[1]:]CORRection:CSET2:STATe <boolean>	192
[SENSe[1]:]CORRection:DCYClE GAIN3[:]INPut[:]MAGNitude <numeric_value>	194
[SENSe[1]:]CORRection:DCYClE GAIN3:STATe <boolean>	197
[SENSe[1]:]CORRection:FDOFFset GAIN4[:]INPut[:]MAGNitude?	199
[SENSe[1]:]CORRection:GAIN2 Commands	200
[SENSe[1]:]CORRection:GAIN2:STATe <boolean>	201
[SENSe[1]:]CORRection:GAIN2[:]INPut[:]MAGNitude <numeric_value>	203
[SENSe[1]]:DETEctor:FUNCTion <character_data>	206
[SENSe[1]:]FREQuency[:]CW[:]FIXed <numeric_value>	209
[SENSe[1]:]FREQuency[:]CW[:]FIXed:STARt <numeric_value> <unit>	211
[SENSe[1]:]FREQuency[:]CW[:]FIXed:STEP <numeric_value>	214
[SENSe[1]:]FREQuency[:]CW[:]FIXed:STOP <numeric_value> <unit>	217
[SENSe[1]:]MRATe <character_data>	220
[SENSe[1]:]POWer:AC:RANGe <numeric_value>	223
[SENSe[1]:]POWer:AC:RANGe:AUTO <boolean>	225
[SENSe[1]:]SPEed <numeric_value>	227
[SENSe[1]:]SWEep[1] Commands	229
[SENSe[1]:]SWEep[1]:OFFSet:TIME <numeric_value>	231
[SENSe[1]:]SWEep[1]:TIME <numeric_value>	233

[SENSE[1]:]TEMPerature?	235
[SENSE[1]]:TRACe Commands	236
[SENSE[1]]:TRACe:OFFSet:TIME <numeric_value>	237
[SENSE[1]]:TRACe:TIME <numeric_value>	239
[SENSE[1]]:TRACe:UNIT <character_data>	241

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

[SENSe] Subsystem

The **SENSe** command subsystem directly affects device-specific settings used to make measurements. The **SENSe** subsystem is optional since this is the primary function of the U2000 Series power sensor. The high level command **CONFigure** uses the **SENSe** commands to prepare the U2000 Series power sensor for making measurements. At a lower level **SENSe** enables you to change the following parameters: **RANGe**, **FREQUency**, **LOSS**, **GAIN2** (channel offset), **DCYClE** (duty cycle), and **AVERage** without completely re-configuring the U2000 Series power sensor.

The **SENSe** command subsystem also allows you to select the measurement and a frequency-dependent offset table.

Keyword	Parameter Form	Notes	Page
[SENSe[1]:]			
AVERage			
:COUNT	<numeric_value>		page 178
:AUTO	<boolean>		page 180
:SDEtect	<boolean>	[non-SCPI]	page 183
[:STATe]	<boolean>		page 185
BUFFer			
:COUNT	<numeric>		page 187
CORRection			
:CSET2			
[:SELEct]	<"string">		page 190
:STATe	<boolean>		page 192
:DCYClE GAIN3		[non-SCPI]	
[:INPut]			
[:MAGNitude]	<numeric_value>		page 194
:STATe	<boolean>		page 197
:FDOffset GAIN4			
[:INPut]			
[:MAGNitude]?		[query only]	page 199

Keyword	Parameter Form	Notes	Page
:GAIN2			
:STATe	<boolean>		page 201
[:INPut]			
[:MAGNitude]	<numeric_value>		page 203
DETEctor			
:FUNCTion	AVERAge NORMAl		page 206
FREQUency			
[:CW :FIXed]	<numeric_value>		page 209
:START	<numeric_value>		page 211
:STEP	<numeric_value>		page 214
:STOP	<numeric_value>		page 217
MRATe	<character_data>		page 220
POWer			
:AC			
:RANGe	<numeric_value>	[non-SCPI]	page 223
:AUTO	<boolean>		page 225
SPEed	<numeric_value>		page 227
SWEEp[1]			page 229
:OFFSet			
:TIME	<numeric_value>		page 231
:TIME	<numeric_value>		page 233
TEMPerature?		[query only]	page 235
SENSE[1]			
:TRACe			
:OFFSet			
:TIME	<numeric_value>		page 237
::TIME	<numeric_value>		page 237
:UNIT	<character_data>		page 237

[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>

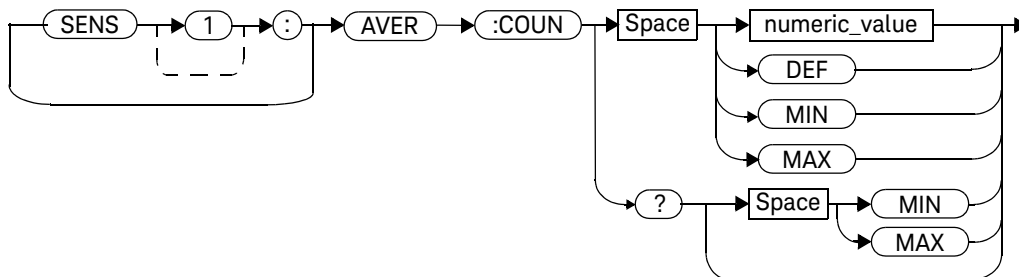
This command is used to enter a value for the filter length. If [SENSe[1]:]AVERAge:COUNT:AUTO is set to **ON**, then entering a value for the filter length automatically sets it to **OFF**. Increasing the value of filter length increases measurement accuracy but also increases the time taken to make 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: <ul style="list-style-type: none"> - DEF: the default value is 4 - MIN: 1 - MAX: 1024 	1 to 1024 DEF MIN MAX

Example

AVER:COUN 400

This command enters a filter length of 400 for the U2000 Series power sensor.

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 for the U2000 Series power sensor.

Error Messages

If a filter length value is entered using **[SENSe[1]:]AVERage:COUNT** while **[SENSe[1]:]MRATe** is set to **FAST**, error -221, "Settings Conflict" occurs. However, the filter length value is set but the **[SENSe[1]:]AVERage:STATe** command is not automatically set **ON**.

[SENSe[1]:]AVERAge:COUNT:AUTO <boolean>

This command enables and disables automatic averaging.

When the auto-filter mode is enabled, the U2000 Series power sensor 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 U2000 Series power sensor is currently operating. Figure 7-1 is an example of the averaged number of readings for each range and resolution when the U2000 Series power sensor is in auto measurement average mode.

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

U2000 Series power sensor Dynamic Range	Maximum Sensor Power	Resolution Setting				Number of Averages
		1	2	3	4	
		1	1	1	8	
10 dB		1	1	1	16	
10 dB		1	1	2	32	
10 dB		1	1	16	256	
10 dB	Minimum Sensor Power	1	8	128	128	

Figure 7-1 Example of averaged readings

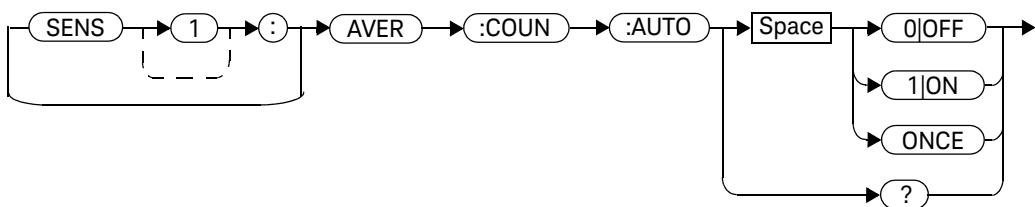
If [SENSe[1]:]AVERAge:COUNT:AUTO is set to OFF, the filter length is set by the [SENSe[1]:]AVERAge:COUNT command. Using the [SENSe[1]:]AVERAge:COUNT command disables automatic averaging.

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:COUNT:AUTO OFF

This command disables automatic filter length selection for the U2000 Series power sensor.

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

Example 1: Query Example

AVER:COUN:AUTO?

Queries whether automatic filter length selection is on or off for the U2000 Series power sensor.

Error Messages

If **SENSE[1]:AVERAGE:COUNT:AUTO** is set to **ON** while **SENSE[1]:MRATE** is set to **FAST** or **[SENSE[1]:]SPEED** is set to **110**, error -221, "Settings Conflict" occurs. However, automatic averaging is enabled but the **SENSE[1]:AVERAGE:STATE** command is not automatically set **ON**.

[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

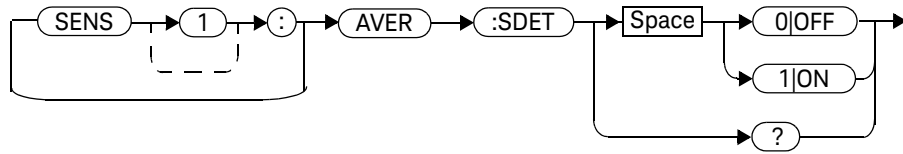
Step detection is automatically disabled when **TRIGger[1]:DELay:AUTO** is **ON** and the trigger mode is not set to free run.

Under this circumstances, the value of [SENSe[1]:]AVERAge:SDETECT is ignored. [SENSe[1]:]AVERAge:SDETECT is not set by the sensor (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

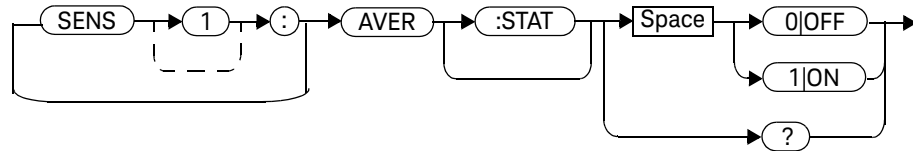
AVER:SDET?

Queries whether step detection is 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 on the U2000 Series power sensor.

Reset Condition

On reset, averaging is **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 on or off for the U2000 Series power sensor.

Error Messages

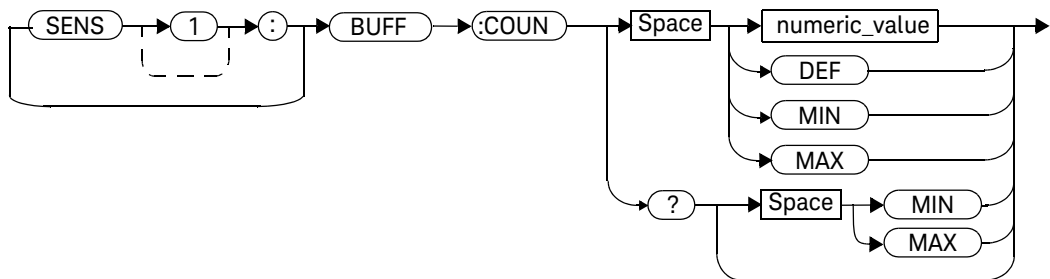
If **[SENSE[1]:]AVERage:STATE** is set to **ON** while **[SENSE[1]:]MRATe** is set to **FAST** or **[SENSE[1]:]SPEed** is set to **110**, error -221, "Settings Conflict" occurs.

[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.

It can only be set when frequency sweep is disabled (**FREQ:STEP 0**). Otherwise, this parameter will be automatically overwritten by frequency sweep step.

Syntax



Parameters

Item	Description/Default	Range of Values
numeric_value	A numeric value for buffer size: - DEF: 1 - MIN: 1 - MAX: 250	1 to 250 DEF MIN MAX

Example

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

Error Messages

- If **[SENSe[1]:]MRATe** is set to **FAST** or **[SENSe[1]:]SPeEd** is set to **110** after the **[SENSe[1]:]BUFFer:COUNT** is set to **>1** and/or **[SENSe[1]:]FREQuency[:CW]:FIXed]:STEP** is set to **>0**, error -221, "Settings conflict" occurs.
- If acquisition mode is in free run, error -221, "Settings conflict" occurs.
- If frequency sweep step is non-zero, error -221, "Settings conflict" occurs.
- If parameter is set lower than 1, error -222, "Data out of range" occurs.
- If parameter is set higher than 250, error -222, "Data out of range" occurs.

[SENSe[1]:]CORRection:CSET2 Commands

These commands are used to select the active sensor 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 sensor frequency-dependent offset table which is to be used.

NOTE

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

Syntax



Parameters

Item	Description/Default	Range of Values
string	String data representing a sensor 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 sensor 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 name of the selected table is returned as a quoted string. If no table is selected, an empty string is returned.

Query Example

CORR:CSET2?

Queries the sensor frequency-dependent offset table currently used.

Error Messages

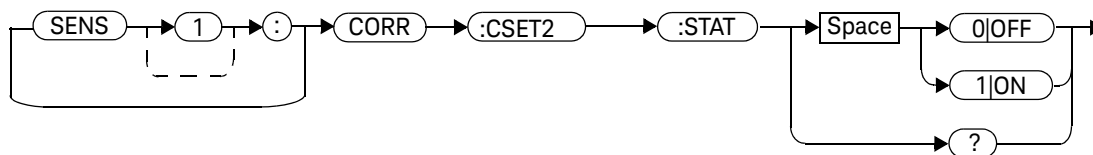
- If **<"string">** is not valid, error -224, "Illegal parameter value" occurs.
- If a table called **<"string">** does not exist, error -256, "File name not found" occurs.
- When a frequency-dependent offset table is selected, the U2000 Series power sensor verifies that the number of offset points defined 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 is to enable and disable the use of the currently active sensor frequency-dependent offset table (**CSET2**). When a table has been selected and enabled, the sensor frequency-dependent offsets stored in it can be used by specifying the required frequency using the **SENSe[1]:FREQUENCY** command.

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

Syntax



Example

CORR:CSET2:STAT 1

This command enables the use of the currently active sensor frequency-dependent offset table for the U2000 Series power sensor.

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 sensor frequency-dependent offset table for the sensor.

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**.

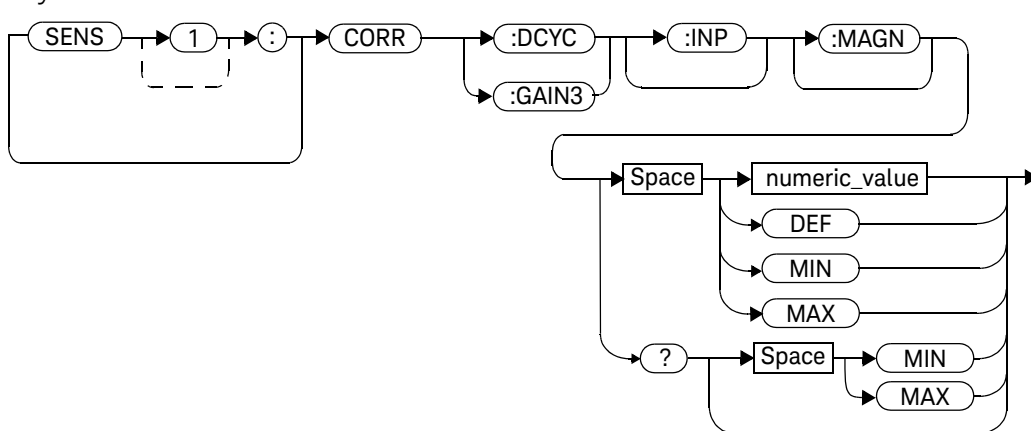
[SENSe[1]:]CORRection:DCYClE|GAIN3[:INPut][[:MAGNitude]
 <numeric_value>

This command is used to set the duty cycle for the pulse power measurement feature of the U2000 Series power sensor. 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 U2000 Series power sensor 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: <ul style="list-style-type: none"> - DEF: the default value is 1% - MIN: 0.001% - MAX: 99.999% The units are PCT, and are 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.*

Error Messages

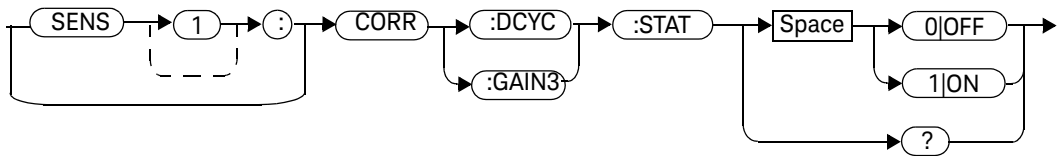
If a duty cycle value is entered using `[SENSE[1]:]CORRection:DCYClE:GAIN3` while `[SENSE[1]:]MRATE` is set to **FAST** or `[SENSE[1]:]SPEEd` is set to **110**, error -221, "Settings Conflict" occurs. However, the duty cycle value is set but the `[SENSE[1]:]CORRection:DCYClE:GAIN3:STATe` command is not automatically set **ON**.

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

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

The [SENSe[1]:]CORRection:DCYClE|GAIN3:STATe command should be used to enter the duty cycle of the signal you want to measure.

Syntax



Example

CORR:DCYC:STAT 1

This command enables the pulse measurement feature.

Reset Condition

On reset, the pulse power measurement feature 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 feature.

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

Query Example

CORR:GAIN3:STAT?

Queries whether the pulse measurement feature is on or off.

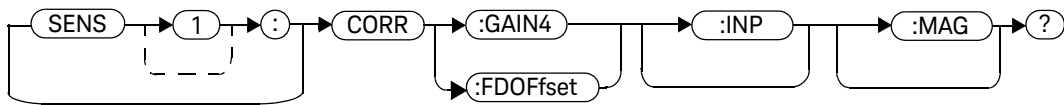
Error Messages

If [SENSE[1]:]CORREction:DCYCLE:STATus is set to **ON** while [SENSE[1]:]MRATE is set to **FAST** or [SENSE[1]:]SPEEd is set to **110**, error -221, "Settings Conflict" occurs.

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

This query is used to return 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:

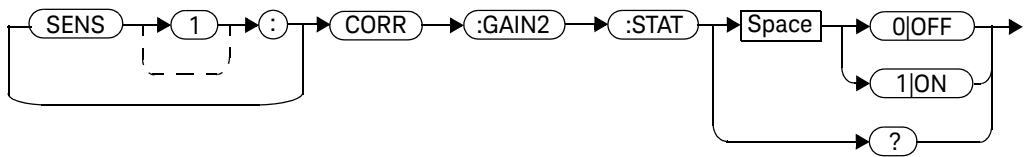
```
[SENSE[1]:]CORREction:GAIN2:STATE <boolean>
```

```
[SENSE[1]:]CORREction:GAIN2[:INPut][:MAGNitude] <numeric_value>
```


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

This command is used to enable/disable a channel offset for the U2000 Series power sensor 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, 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 offsets.

- 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.

Error Messages

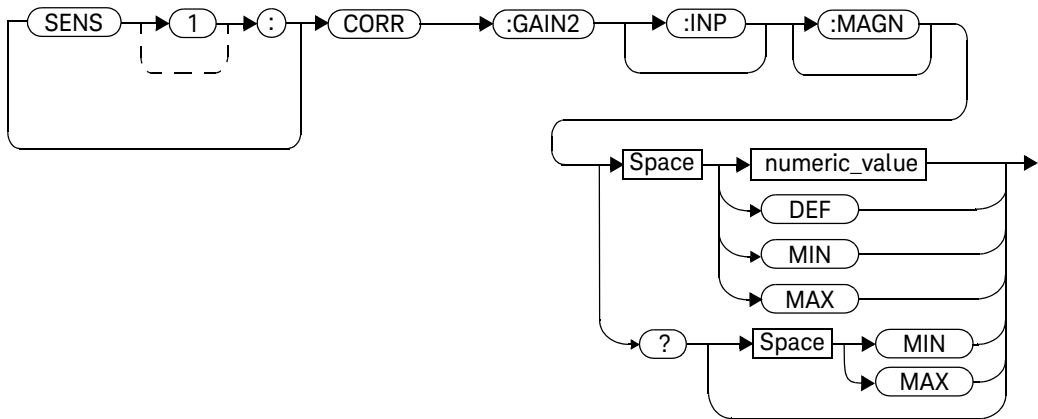
If `[SENSE[1]:]CORRection:GAIN2:STATe` is set to **ON** while `[SENSE[1]:]MRATe` is set to **FAST** or `[SENSE[1]:]SPEEd` is set to **110**, error -221, "Settings Conflict" occurs.

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

This command is used to enter a channel offset value for the U2000 Series power sensor setup, for example cable loss. The U2000 Series power sensor 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.

Error Messages

If a loss/gain correction value is entered using **[SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude]** while **[SENSe[1]:]MRATe** is set to **FAST** or **[SENSe[1]:]SPEEd** is set to **110**, error -221, "Settings Conflict" occurs. However, the correction value is set but the **[SENSe[1]:]CORRection:GAIN2:STATe** command is not automatically set **ON**.

[SENSe[1]]:DETECTOR:FUNCTION <character_data>

This command is used to set the measurement mode for the U2000 Series power sensors either to **AVERage** (chopper-based measurement) or **NORMal** (sample-based measurement).

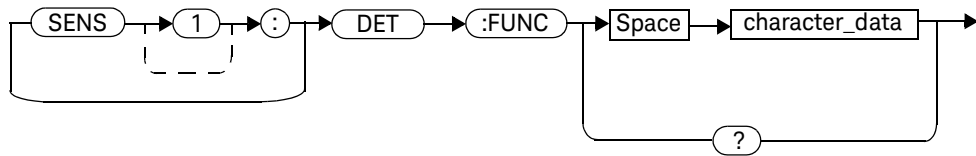
When the measurement mode is set to **AVERage**:

- If **TRIGger:SOURce** is set to **INTernal** or **EXTernal**, it is set automatically to **IMMediate**.
- **INITiate:CONTinuous** is set automatically to **ON**.
- **CALCulate:FEED** is set automatically to **"POW:AVER"** for all **CALC** block using the specified channel in their **CALCulation:MATH[:EXPRession]**.

When the measurement mode is set to **NORMal**:

- **SENSe:CORRection:DCYCLE:STATe** is set automatically to **OFF**.
- If **TRIGger:SOURce** is set to **IMMediate**, it is set automatically to **INTernal**.
- **CALCulate:FEED** is set automatically to **"POW:AVER ON SWEEP1"**.

Syntax



Parameters

Item	Description/Default	Range of Values
character_data	Defines the measurement mode: - AVERAge : sets the sensor to average only mode - NORMa1 : sets the sensor to normal mode	AVERAge NORMa1

Example

```
DET:FUNC AVER
```

This command sets the sensor to average only mode.

Reset Condition

On reset, the measurement mode is set to **AVERAge**.

Query

```
[SENSe[1]]:DETEctor:FUNctIon?
```

The command returns the current measurement mode for the sensor.

Query Example

DET:FUNC?

Queries the current measurement mode for the U2000 Series power sensor.

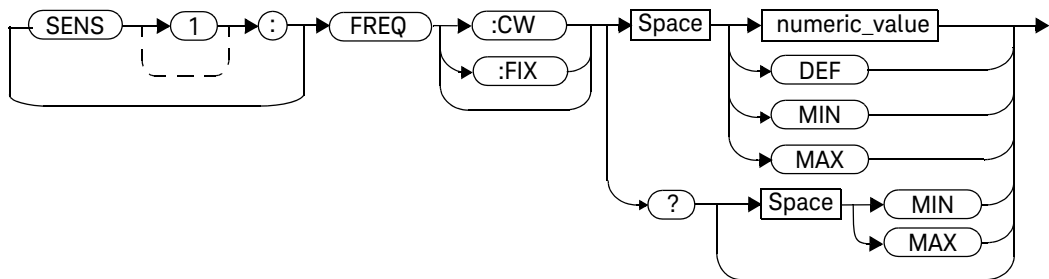
Error Messages

If **[SENSE[1]:]DETECTOR:FUNCTION** is used when U2004A model is connected, error -241, "Hardware missing" occurs.

[SENSe[1]:]FREQuency[:CW]:FIXed] <numeric_value>

This command is used to enter a frequency. The appropriate frequency-dependent offset corrections are applied for the frequency selected, dependant on the frequency-dependent offset data stored in the sensor 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 : 1 kHz - MAX : 1000.0 GHz The default units are Hz.	1 kHz to 1000.0 GHz ^[a] DEF MIN MAX

[a] The following measurement units can be used:

- Hz
- kHz (10³)
- MHz (10⁶)
- GHz (10⁹)

Example

FREQ 500kHz

This command enters a frequency of 500 kHz.

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 sensor 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.

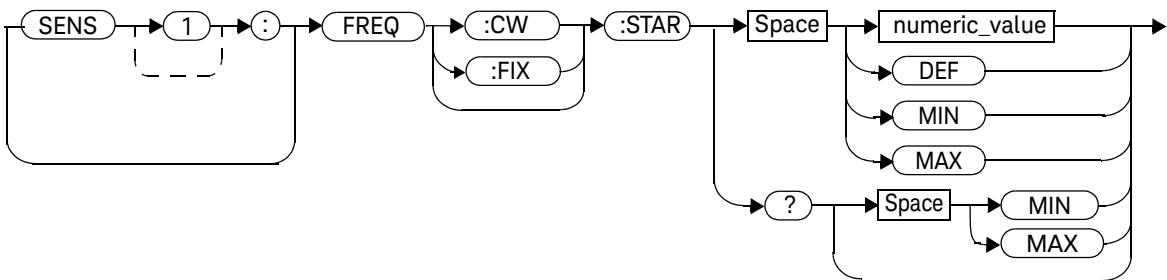
If frequency sweep is disabled (frequency sweep step set to 0), start frequency will be set but it will not take effect.

[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 frequency sweep mode is configured with frequency step size within the allowable range (1 to 250), the following applies:

- If frequency stop point is greater than frequency start point, the frequency range will be swept in an ascending order.
- If frequency stop point is less than frequency start point, the frequency range will be swept in a descending order.
- If frequency stop point and frequency start point are equal, it is equivalent to 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: 1 kHz – MAX: 1000.0 GHz The default units are Hz.	1 kHz 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 This command sets 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 is used to retrieve the start frequency of the average trigger frequency sweep. Frequency is returned in Hz.

Query Example

FREQ:STAR? *This query returns the start frequency of the frequency sweep in Hz.*

Error Messages

- If parameter set is lower than 1 kHz, error -222, "Data out of range" occurs.
- If parameter set is higher than 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 using equation below:

$$Step = \frac{f_{stop} - f_{start} + Interval}{Interval}$$

where,

Step = Number of frequency steps

fstart = Start frequency

fstop = Stop frequency

Interval = Interval between each frequency point

Example

If you wish to sweep from 1 GHz to 5 GHz, in intervals of 0.5 GHz, the Step is derived as follows:

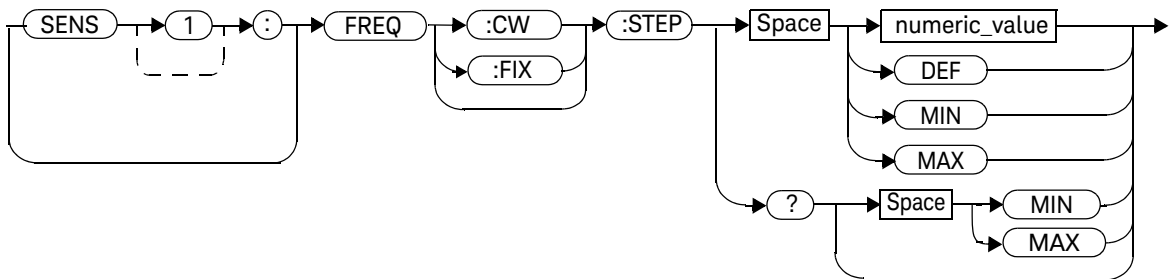
$$Step = \frac{f_{stop} - f_{start} + Interval}{Interval}$$

$$Step = \left(\frac{5 - 1 + 0.5}{0.5} \right) = 9$$

[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 U2000 Series power sensor, 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:	0 to 250
	- DEF: the default value is 0	DEF
	- MIN: 0	MIN
	- MAX: 250	MAX

Example

FREQ:STEP 10 This command 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 is used to retrieve the number of steps in the triggered frequency sweep.

Query Example

FREQ:STEP? *This query returns the number of steps in the frequency sweep.*

Error Messages

- If **[SENSe[1]:]MRATe** is set to **FAST** or **[SENSe[1]:]SPEEd** is set to **110** after the **[SENSe[1]:]BUFFer:COUNT** is set to **>1** and/or **[SENSe[1]:]FREQuency[:CW]:FIXed]:STEP** is set to **>0**, error -221, "Settings conflict" occurs.
- If parameter is set lower than **1**, error -222, "Data out of range" occurs.
- If parameter is set higher than **250**, error -222, "Data out of range" occurs.

[SENSe[1]:]FREQuency[:CW]:FIXed]:STOP <numeric_value> <unit>

This command sets the stop frequency of frequency sweep. It must be used in conjunction with an external trigger.

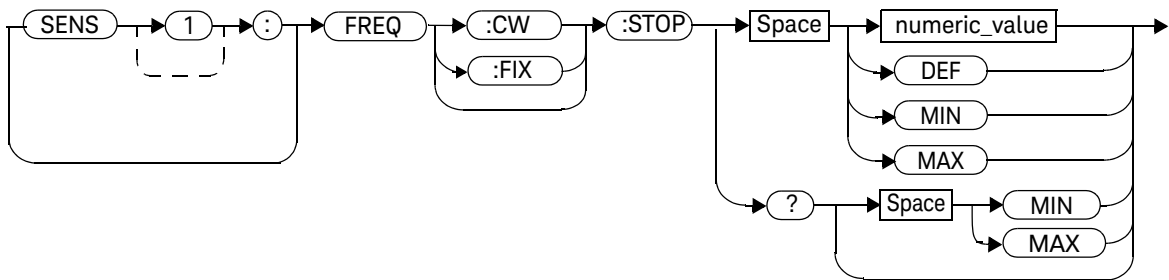
If frequency sweep is disabled (frequency sweep step set to 0), stop frequency will be set but it will not take effect.

[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 of 1 to 250, the following applies:

- If the stop frequency point is greater than frequency start point, the frequency range will be swept in an ascending order.
- If the stop frequency point is less than 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 Hz – MIN : 1 kHz – MAX : 1000.0 GHz The default units are Hz	1 kHz 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 *This command sets 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 is used to retrieve the stop frequency of the triggered frequency sweep. Frequency is returned in Hz.

Query Example

FREQ:STOP? *This query returns the stop frequency of the frequency sweep in Hz.*

Error Messages

- If parameter set is lower than 1 kHz, error -222, "Data out of range" occurs.
- If parameter set is higher than 1000 GHz, error -222, "Data out of range" occurs.

[SENSe[1]:]MRATe <character_data>

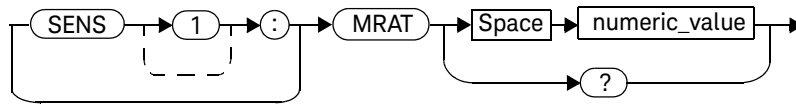
This command sets the measurement rate on the U2000 Series power sensor.

When the measurement rate is set to **FAST**, the following couplings occur:

Command	Status
[SENSe[1]:]AVERAge: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 **NORMa1** or **DOUBle**, 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 : - NORMa1 : 20 readings/s - DOUB1e : 40 readings/s - FAST : up to 110 readings/s The default is NORMa1 .	NORMa1 ^[a] DOUB1e ^[a] FAST ^[a]

[a] When the measurement rate is set to **NORMa1**, **DOUB1e**, or **FAST**, **TRIGger[:SEquence[1]]:COUNT** is set automatically to 1.

NOTE

In **FAST** mode, set the trigger count to 50 (buffered mode) to achieve up to a measurement speed of 1000 readings/s. The 1000 readings/s is the derived measurement speed from the first 50 readings only in buffered mode. The actual maximum number of measurement readings that can be taken in one second is 250. For detailed information about the trigger count setting, refer to **TRIGger[:SEquence[1]]:COUNT <numeric_value>**.

Example

MRAT DOUB

This command sets the measurement rate to 40 readings/s.

Reset Condition

On reset, the measurement rate is set to **NORMa1**.

Query

[SENSe[1]:]MRATe?

The query returns the current setting of either **NORMa1**, **DOUB1e**, or **FAST**.

Query Example

MRAT?

Queries the current measurement rate setting.

Error Messages

If **<character_data>** is not set to **NORMa1**, **DOUB1e**, or **FAST**, error -224, "Illegal parameter value" occurs.

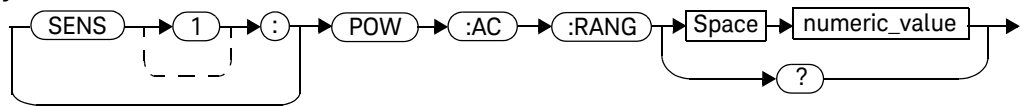
[SENSe[1]:]POWer:AC:RANGe <numeric_value>

This command selects one of two power ranges.

- If 0 is selected, the U2000 Series power sensor lower range is selected
- If 1 is selected, the U2000 Series power sensor upper range is selected

Setting a range with this command automatically switches [SENSe[1]:]POWer:AC:RANGe:AUTO to OFF.

Syntax



Example

POW:AC:RANG 0

This command sets the U2000 Series power sensor to its lower range.

Reset Condition

On reset, the upper range is selected.

Query

[SENSe[1]:]POWer:AC:RANGe?

The query enters a 1 or 0 into the output buffer indicating the status of the U2000 Series power sensor range.

- 1 is returned when the upper range is selected
- 0 is returned when the lower range is selected

Query Example

POW:AC:RANG?

Queries the current setting of the U2000 Series power sensor range.

[SENSe[1]:]POWer:AC:RANGe:AUTO <boolean>

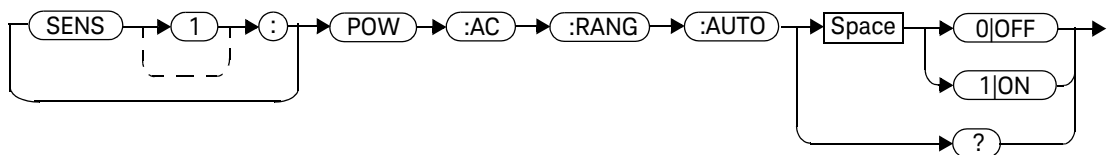
Its purpose is to enable and disable auto-ranging. When auto-ranging is **ON**, the U2000 Series power sensor selects the best measuring range. When auto-ranging is set to **OFF**, the U2000 Series power sensor remains in the currently set range.

The [SENSe[1]:]POWer:AC:RANGe command disables auto-ranging.

If **INITiate[1]:CONTinuous** is set to **ON** and **TRIGger[1]:SOURce** is set to **IMMediate**, the range tracks the input power if [SENSe[1]:]POWer:AC:RANGe:AUTO is **ON**.

If the U2000 Series power sensor is not making measurements, then auto-ranging only occurs when the U2000 Series power sensor is triggered.

Syntax



Example

`POW:AC:RANG:AUTO 0`

This command disables auto-ranging.

Reset Condition

On reset, auto-ranging is enabled.

Query

[SENSE[1]:]POWER:AC:RANGE:AUTO?

The query enters a 1 or 0 into the output buffer indicating the status of auto-ranging.

- 1 is returned when auto-ranging is enabled
- 0 is returned when auto-ranging is disabled

Query Example

POW:AC:RANG:AUTO?

Queries whether auto-ranging is on or off.

[SENSe[1]:]SPEed <numeric_value>

This command sets the measurement speed. The speed values consist of 20, 40, and up to 110 readings/s. When the speed is set to 110 readings/s, the following couplings occur:

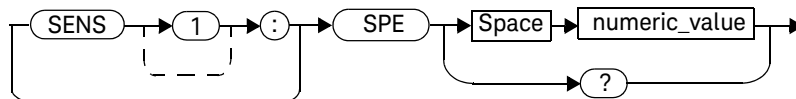
Command	Status
[SENSe[1]:]AVERAge:STATe	OFF ^[a]
[SENSe[1]:]CORREction:GAIN2:STATe	OFF ^[a]
CALCulate[1]:MATH:EXPRession	“(SENS1)”

[a] This change only occurs on the measurement specified in the [SENSe[1]:]SPEed command. When the speed is changed from 110 readings/s to either 20 or 40 readings/s, the settings that were in place when 110 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 <NORMa1|DOUB1e|FAST> command (with 20 readings/s referring to NORMa1, 40 readings/s to DOUB1e, and up to 110 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 110 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 **110**.

Query Example

SPE?

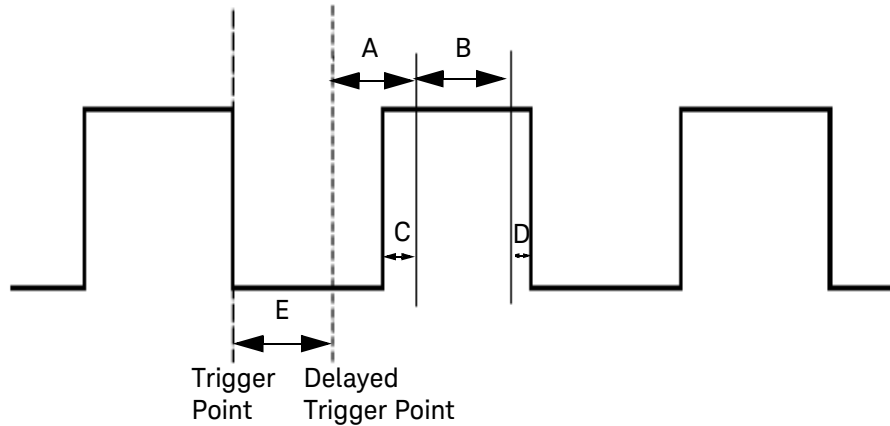
Queries the current speed setting.

Error Messages

- If **<numeric_value>** is not set to **20**, **40**, or up to **110**, error -224, "Illegal parameter value" occurs.
- If **TRIGger[1]:SOURce** is set to **EXternal** or **INTernal** while **[SENSe[1]:]SPEed** is set to **110**, error -221, "Settings Conflict" occurs.

[SENSe[1]:]SWEep[1] Commands

The [SENSe[1]:]SWEep[1] commands set the time offset and gate length as illustrated in the diagram below:



The following commands are detailed in this section:

[SENSe[1]:]SWEep[1]:OFFSet:TIME <numeric_value>

[SENSe[1]:]SWEep[1]:TIME <numeric_value>

- A is offset time: [SENSe[1]:]SWEep[1]:OFFSet:TIME
- B is time gate length: [SENSe[1]:]SWEep[1]:TIME
- C is range settling time: 150 μ s
- D is fall time: 40 μ s
- E is trigger delay: TRIGger[:SEquence]:DELay
- Trigger point is defined using TRIGger[:SEquence]:LEVel, TRIGger[:SEquence]:SLOPe, and TRIGger[:SEquence]:HYSTeresis

NOTE

It is strongly recommended for the gate to have 150 μ s offset from the pulse rising edge and 40 μ s offset from the pulse falling edge to achieve higher accuracy measurements. Samples collected during range settling time are discarded. Thus, fewer samples are used for generating a measurement.

For example, a slot 0 GSM burst signal (577 μs) with triggering point at 100 μs from the rising edge of the GSM burst. The recommended time gating measurement setup is as follows.

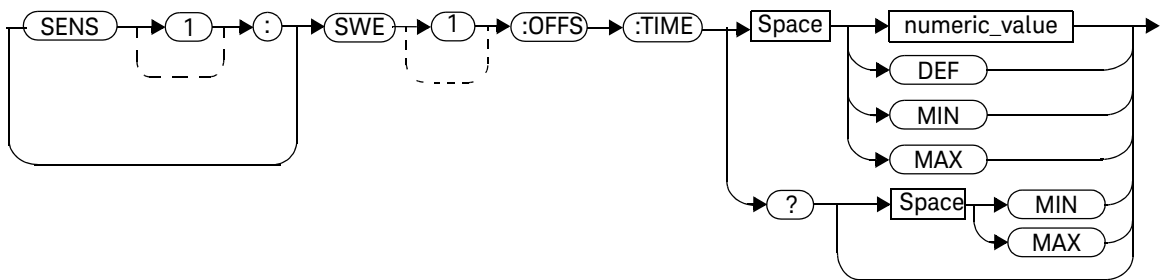
Offset time: [SENSE[1]:]SWEep[1]:OFFSet:TIME 0.00025

Time-gate length: [SENSE[1]:]SWEep[1]:TIME 0.000387

[SENSe[1]:]SWEep[1]:OFFSet:TIME <numeric_value>

This command sets the delay between the trigger point and the start of the time-gated period (offset time) for the U2000 Series power sensor.

Syntax



Parameters

Item	Description/Default	Range of Values
numeric_value	The delay between the trigger point and the start of the time-gated period: – DEF : the default value is 0 s – MIN : -0.15 s – MAX : 0.15 s Units are resolved to 1 μs.	-0.15 to 0.15 s DEF MIN MAX

Example

SWE:OFFS:TIME 0.001

This command sets the delay to 0.001 s.

Reset Condition

On reset, the value is set to 0 s.

Query

[SENSe[1]:]SWEp[1]:OFFSet:TIME? [MIN|MAX]

The query returns the current delay between the trigger point and the start of the time-gated period or the values associated with **MIN** or **MAX**.

Query Example

SWE:OFFS:TIME?

The query returns the current delay between the trigger point and the start of the time-gated period for the U2000 Series power sensor.

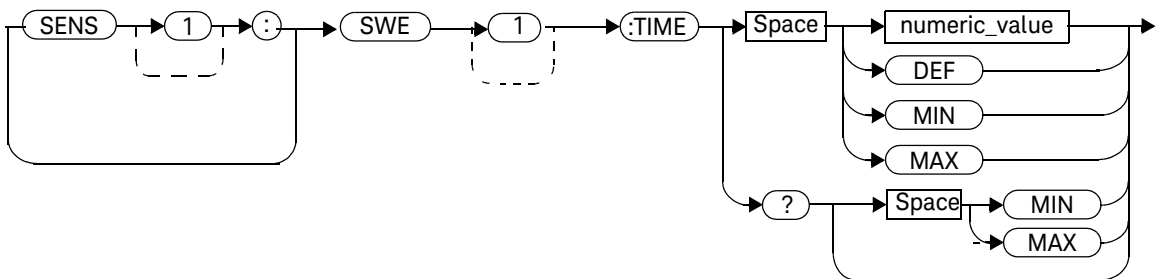
Error Messages

- If the command is used when U2004A model is connected, error -241, “Hardware missing” occurs.
- If the sensor is set to **AVERAge** mode rather than **NORMa1** mode, error -221, “Setting conflicts” occurs.

[SENSe[1]:]SWEep[1]:TIME <numeric_value>

This command sets the length of the time-gated period (time-gate duration) for time-gated measurements for the U2000 Series power sensor.

Syntax



Parameters

Item	Description/Default	Range of Values
numeric_value	The length of the time-gated period in seconds: - DEF : the default value is 100 μ s - MIN : 100 μ s - MAX : the default value is 0.15 s Units are resolved to 1 μ s.	100 μ s to 0.15 s DEF MIN MAX

Example

SWE:TIME 0.001

This command sets the length to 0.001 s.

Reset Condition

On reset, the length of the time is set to 100 μ s.

Query

[SENSE[1]:]SWEep[1]:TIME? [MIN|MAX]

The query returns the current length of the time-gated period or the values associated with **MIN** or **MAX**.

Query Example

SWE:TIME? *Queries the length of the time-gated period for the U2000 Series power sensor.*

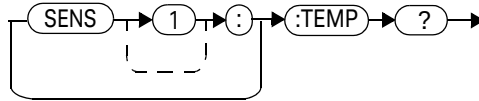
Error Messages

- If the command is used when U2004A model is connected, error -241, "Hardware missing" occurs.
- If the sensor is set to **AVERage** mode rather than **NORMa1** mode, error -221, "Settting conflicts" occurs.

[SENSe[1]:]TEMPerature?

This query returns the U2000 Series power sensor temperature in degrees Celsius.

Syntax



Parameters

Item	Description/Default	Range of Values
numeric_value	A numeric value defining sensor temperature in degrees Celsius.	-50 to 100

Example

TEMP?

This query returns the current sensor temperature.

Reset Condition

On reset, this parameter is not affected.

[SENSE[1]]:TRACe Commands

These commands are used to set:

- The delay between the delayed trigger point and the start of the trace
- The duration of the trace
- The trace unit

The following commands are detailed in this section:

SENSE[1]:TRACe:OFFSet:TIME <numeric_value>

SENSE[1]:TRACe:TIME <numeric_value>

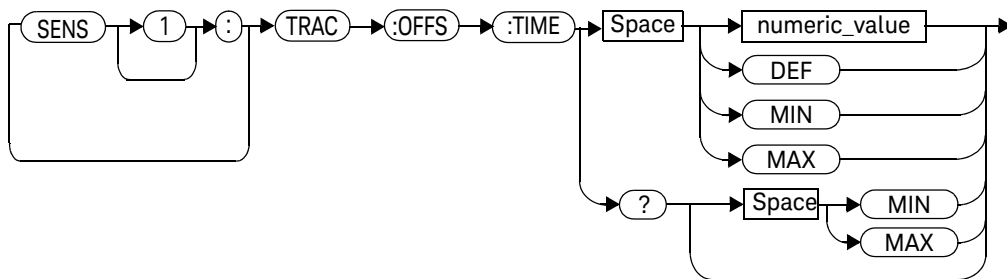
SENSE[1]:TRACe:UNIT <character_data>

[SENSe[1]]:TRACe:OFFSet:TIME <numeric_value>

This command sets the delay between the delayed trigger point and the start of the trace for the sensors that are set to **NORMa1** mode.

To set the sensor to **NORMa1** mode, refer to the command “[SENSe[1]]:DETEctor:FUNCTion <character_data>” on page 206.

Syntax



Parameters

Item	Description/Default	Range of Values
numeric_value	The length of the delay in seconds: – DEF : the default value is 0 s – MIN : -0.15 s – MAX : 0.15 s Units are resolved to 1 μ s.	-0.15 to 0.15 s DEF MIN MAX

Example

SENS:TRAC:OFFS:TIME 0.05

This command sets the delay to 0.05 seconds.

Reset Condition

On reset, the delay is set to 0 seconds.

Query

[SENSE[1]]:TRACe:OFFSet:TIME? [MIN|MAX]

The query returns the current delay between the delayed trigger point and the start of the trace or the values associated with **MIN** and **MAX**. The format of the response is **<NR3>**.

Query Example

SENS:TRAC:OFFS:TIME?

This command queries the current delay between the delayed trigger point and the start of the trace.

Error Messages

- If the command is used when U2004 model is connected, error -241, "Hardware missing" occurs.
- If the sensor is set to **AVERage** mode rather than **NORMa1** mode, error -221, "Settings conflict" occurs.

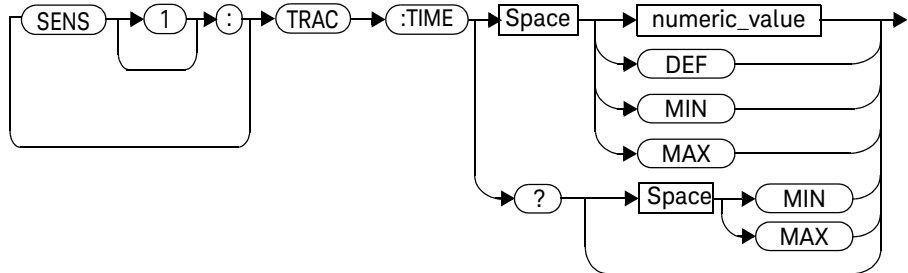
[SENSe[1]]:TRACe:TIME <numeric_value>

This command sets the duration of the trace for the sensor which is set to **NORMa1** mode. To set the sensor to **NORMa1** mode, refer to the command “[SENSe[1]]:DETEctor:FUNCTion <character_data>” on page 206.

NOTE

If **INITiate[1]:CONTInuous 1|ON** is being sent while the trace duration is set to >0.1 s, it is strongly recommended to set a timeout of approximately 30 s.

Syntax



Parameters

Item	Description/Default	Range of Values
numeric_value	The duration of the trace in seconds: – DEF : the default value is 500 μ s. – MIN : 100 μ s – MAX : 150 ms Units are resolved to 1 μ s.	100 μ s to 150 ms DEF MIN MAX

Example

SENS:TRAC:TIME 0.1

This command sets the duration of the trace to 0.1 seconds for the sensor.

Reset Condition

On reset, the duration is set to 500 μ s.

Query

[SENSe[1]]:TRACe:TIME? [MIN|MAX]

The query returns the current duration of the trace or the values associated with **MIN** and **MAX**. The format of the response is **<NR3>**.

Query Example

SENS:TRAC:TIME?

This command queries the current duration of the trace.

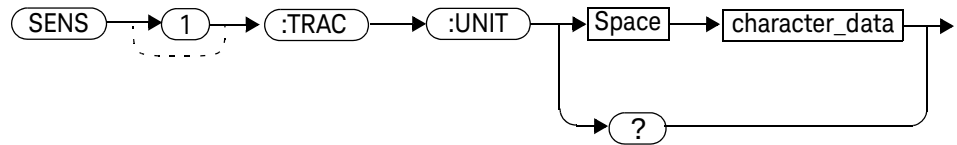
Error Messages

- If the command is used when U2004 model is connected, error -241, "Hardware missing" occurs.
- If the sensor is set to **AVERage** mode rather than **NORMa1** mode, error -221, "Settings Conflict" occurs.

[SENSe[1]]:TRACe:UNIT <character_data>

This command sets the units for the trace.

Syntax



Parameters

Item	Description/Default	Range of Values
character_data	- DBM: dBm - W: Watts	DBM W

Example

SENSe:TRAC:UNIT W

This command sets the trace units to Watts.

Reset Condition

On reset, the units are set to dBm.

Query

[SENSe[1]]:TRACe:UNIT?

The query command returns the current trace units.

Query Example

SENS:TRAC:UNIT?

This command queries the current trace units.

8 SERVICE Subsystem

SERVICE Subsystem	244
SERVICE:BIST:TRIGGER:LEVEL:STATE?	245
SERVICE:OPTION?	246
SERVICE:SENSOR[1]:CDATE?	247
SERVICE:SENSOR[1]:CDUEdate <"date">	248
SERVICE:SENSOR[1]:CPLace <"place">	249
SERVICE:SENSOR[1]:FREQUENCY:MAXimum?	251
SERVICE:SENSOR[1]:FREQUENCY:MINimum?	252
SERVICE:SENSOR[1]:POWER:AVERage:MAXimum?	253
SERVICE:SENSOR[1]:POWER:USABLE:MAXimum?	254
SERVICE:SENSOR[1]:POWER:USABLE:MINimum?	255
SERVICE:SENSOR[1]:RADC?	256
SERVICE:SENSOR[1]:SNUMBER?	257
SERVICE:SENSOR[1]:TNUMBER <"tracking_number">	258
SERVICE:SENSOR[1]:TYPE?	259
SERVICE:SNUMBER?	260

This chapter explains how the **SERVICE** command subsystem is used to obtain and set information useful for servicing the U2000 Series power sensors.

SERVICE Subsystem

The **SERVICE** command subsystem is used to load information such as the U2000 Series power sensor processor board revision version and obtain information such as the serial number of the current sensor(s) being used.

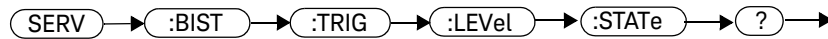
Keyword	Parameter Form	Notes	Page
SERVICE			
:BIST			
:TRIGger			
:LEVEl			
:STATe?		[query only]	page 245
:OPTion?		[query only]	page 246
:SENSor[1]			
:CDATe?		[query only]	page 247
:CDUEdate	<"date">		page 248
:CPLace	<"place">		page 249
:FREQuency			
:MAXimum?		[query only]	page 251
:MINimum?		[query only]	page 252
:POWer			
:AVERage			
:MAXimum?		[query only]	page 253
:USABLE			
:MAXimum?		[query only]	page 254
:MINimum?		[query only]	page 255
:RADc?		[query only]	page 256
:SNUMber?		[query only]	page 257
:TNUMber	<"tracking_number">		page 258
:TYPE?		[query only]	page 259
:SNUMber?		[query only]	page 260

SERvice:BIST:TRIGger:LEVel:STATe?

Queries the trigger level.

- 1 is returned when the external trigger input is high.
- 0 is returned when the external trigger input is low.

Syntax



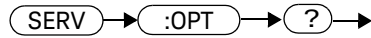
Example

SERV:BIST:TRIG:LEV:STAT? *Queries the trigger level.*

SERVICE:OPTion?

This query is used to determine the option of your U2000 Series power sensor.

Syntax



Example

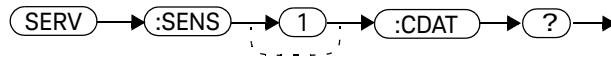
SERVICE:OPTion?

This query returns the current option string.

SERVice:SENSor[1]:CDATe?

This query returns the calibration date for the U2000 Series power sensor. The calibration date information is stored in the sensor non-volatile memory.

Syntax



Example

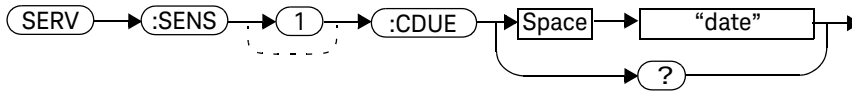
SERV:SENS:CDAT?

This query returns the calibration date of the U2000 Series power sensor.

SERVICE:SENSOR[1]:CDUEdate <"date">

This command is used to enter the calibration due date.

Syntax



Example

```
SERV:SENS:CDUE
"2006,09,21"
```

This command enters the calibration due date of the U2000 Series power sensor as 21st September 2006.

Query

```
SERVICE:SENSOR[1]:CDUEdate?
```

This query returns the calibration due date in the U2000 Series power sensor. Calibration due date information is stored in the sensor non-volatile memory.

Query Example

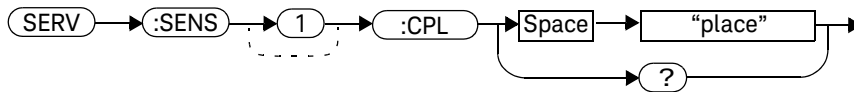
```
SERV:SENS:CDUE?
```

This query returns the calibration due date of the U2000 Series power sensor.

SERvice:SENSor[1]:CPLace <“place”>

This command is used to enter the calibration location. A maximum number of eight alphanumeric characters can be entered.

Syntax



Example

SERV:SENS:CPL "Agil-Pen"

This command enters the place of calibration of the U2000 Series power sensor as Keysight Penang.

Query

SERvice:SENSor[1]:CPLace?

This query returns a string specifying the calibration location, as stored in the U2000 Series power sensor non-volatile memory.

Query example

SERV:SENS:CPL?

This query returns the place of calibration of the U2000 Series power sensor.

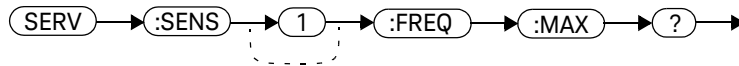
NOTE

If the calibration location is not pre-programmed, "NONE" is returned.

SERVice:SENSor[1]:FREQuency:MAXimum?

This query returns the maximum frequency that can be measured by the U2000 Series power sensor, in MHz. The maximum frequency information is stored in the U2000 Series power sensor non-volatile memory.

Syntax



Example

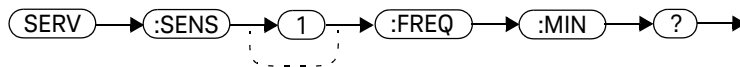
SERV:SENS:FREQ:MAX?

This query returns the maximum frequency that can be measured by the U2000 Series power sensor.

SERVice:SENSor[1]:FREQuency:MINimum?

This query returns the minimum frequency that can be measured by the U2000 Series power sensor, in MHz. The minimum frequency information is stored in the U2000 Series power sensor non-volatile memory.

Syntax



Example

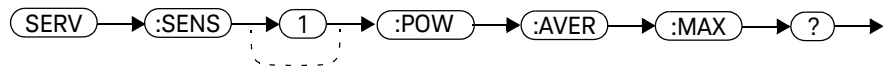
SERV:SENS:FREQ:MIN?

This query returns the minimum frequency that can be measured by the U2000 Series power sensor.

SERVice:SENSor[1]:POWer:AVERage:MAXimum?

This query returns the maximum average power that can be measured by the sensor, in dBm. The maximum average power information is stored in the sensor non-volatile memory.

Syntax



Example

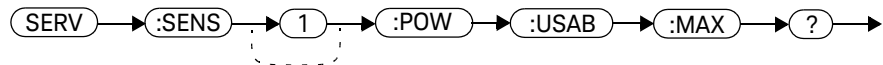
SERV:SENS:POW:AVER:MAX?

This query returns the maximum average power that can be measured by the U2000 Series power sensor.

SERvice:SENSor[1]:POWer:USABle:MAXimum?

This query returns the maximum power that can be accurately measured by the sensor, in dBm. The maximum power information is stored in the sensor non-volatile memory.

Syntax



Example

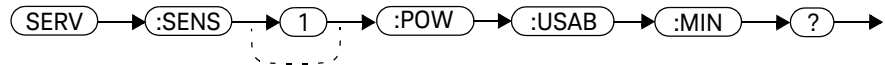
SERV:SENS:POW:USAB:MAX?

This query returns the maximum power that can be accurately measured by the U2000 Series power sensor.

SERvice:SENSor[1]:POWer:USABLE:MINimum?

This query returns the minimum power that can be accurately measured by the sensor, in dBm. The minimum power information is stored in the sensor non-volatile memory.

Syntax



Example

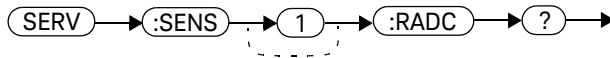
SERV:SENS:POW:USAB:MIN?

This query returns the minimum power that can be accurately measured by the U2000 Series power sensor.

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?
```

This query returns a new raw uncorrected measurement for the sensor.

NOTE

The returned raw measurement unit is float.

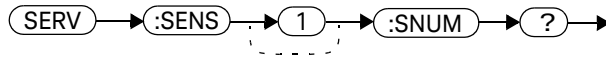
SERvice:SENSor[1]:SNUMber?

This query is used to acquire the serial number of the U2000 Series power sensor in the form of MY12345678.

NOTE

This query performs the same function as the `SERvice:SNUMber?` query.

Syntax



Example

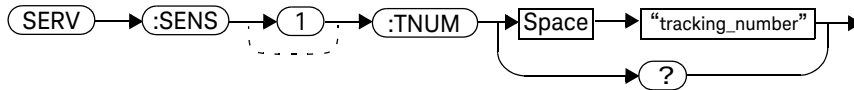
`SERV:SENS:SNUM?`

This query returns the serial number of the U2000 Series power sensor in the form of MY12345678.

SERVICE:SENSOR[1]:TNUMBER <“tracking_number”>

This command is used to enter the tracking number of the U2000 Series power sensor.

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 for the U2000 Series power sensor. The tracking number information is stored in the sensor non-volatile memory.

Query Example

SERV:SENS:TNUM? *This query returns the tracking number of the U2000 Series power sensor.*

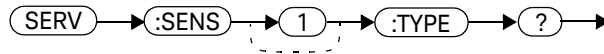
NOTE

If tracking number is not pre-programmed, “NONE” is returned.

SERVice:SENSor[1]:TYPE?

This query identifies the sensor type.

Syntax



Example

SERV:SENS:TYPE?

This query returns either U2000A, U2001A, U2002A, U2004A, U2001H, U2001B, U2000H, U2000B, U2002H, or U2002B.

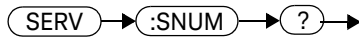
SERVICE:SNUMBER?

This query returns the U2000 Series power sensor 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?

This query returns the U2000 Series power sensor serial number in the form of MY12345678.

9 STATUS Subsystem

STATUS Subsystem	262
Status Register Set Commands	264
Device Status Register Sets	269
Operation Register Sets	270
STATUS:OPERation	271
STATUS:OPERation:CALibrating[:SUMMARY]	272
STATUS:OPERation:LLFail[:SUMMARY]	273
STATUS:OPERation:MEASuring[:SUMMARY]	274
STATUS:OPERation:SENSe[:SUMMARY]	275
STATUS:OPERation:TRIGger[:SUMMARY]	276
STATUS:OPERation:ULFail[:SUMMARY]	277
STATUS:PRESet	278
Questionable Register Sets	279
STATUS:QUEStionable	280
STATUS:QUEStionable:CALibration[:SUMMARY]	281
STATUS:QUEStionable:POWer[:SUMMARY]	282

This chapter explains how the **STATUS** command subsystem enables you to examine the status of the U2000 Series power sensor by monitoring the “Device Status Register”, “Operation Status Register” and the “Questionable Status Register”.

STATUS Subsystem

The **STATUS** command subsystem enables you to examine the status of the U2000 Series power sensor by monitoring the following status registers:

- Device status register
- Operation status register
- Questionable status register

The contents of these and other registers in the U2000 Series power sensor are determined by one or more status registers.

Table 9-1 summarizes the effects of various commands and events on these status registers:

Table 9-1 Commands and events affecting Status register

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
IEEE488.2 Registers ESE SRE	none	none	clear	none
IEEE488.2 Registers SESR 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 eleven status registers:

STATUS:DEVICE (page 269)

STATUS:OPERation (page 271)

STATUS:OPERation:CALibrating[:SUMMARY] (page 272)

STATUS:OPERation:LLFail[:SUMMARY] (page 273)

STATUS:OPERation:MEASuring[:SUMMARY] (page 274)

STATUS:OPERation:SENSe[:SUMMARY] (page 275)

STATUS:OPERation:TRIGger[:SUMMARY] (page 276)

STATUS:OPERation:ULFail[:SUMMARY] (page 277)

STATUS:QUESTionable (page 278)

STATUS:QUESTionable:CALibration[:SUMMARY] (page 281)

STATUS:QUESTionable:POWer[:SUMMARY] (page 282)

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

This chapter describes the status register set commands and the status registers which they are used to examine.

Status Register Set Commands

This section describes the five status register set commands. Each can be used to examine all of the eleven status registers listed on [page 263](#).

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:

Keyword	Parameter Form	Notes	Page
:CONDition?		[query only]	page 264
:ENABLE	<NRf> <non-decimal numeric>		page 265
[:EVENT?]		[query only]	page 265
:NTRansition	<NRf> <non-decimal numeric>		page 266
:PTRansition	<NRf> <non-decimal numeric>		page 267

: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.

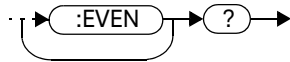
Syntax



[: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.

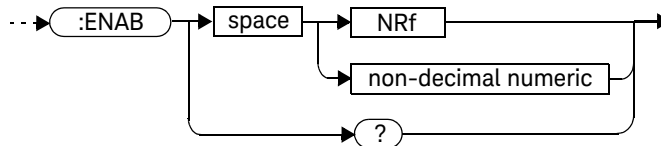
Syntax



: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 $2^{16}-1$

Query

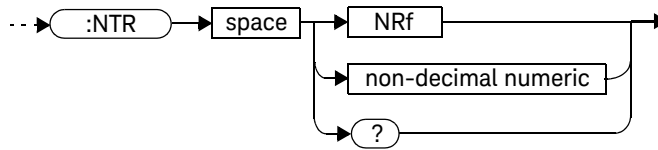
:ENABle?

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	The value used to set the NTR Register.	0 to $2^{16}-1$
non-decimal numeric		

Query

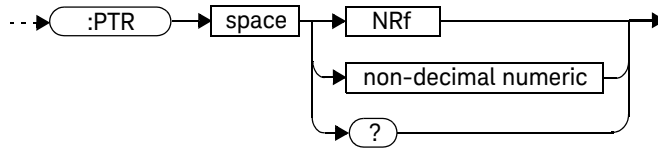
:NTRansition?

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	The value used to set the PTR Register.	0 to $2^{16}-1$
non-decimal numeric		

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 give 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 264.

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 bits in these registers are used by the U2000 Series power sensor:

Bit Number	Decimal Weight	Definition
0	-	Not used
1	2	Not used
2	4	Not used
3	8	U2000 Series power sensor error
4	16	Not used
7-15	-	Not used
14	16384	Not used
15	-	Bit 15 always 0

The U2000 Series power sensor error bits (3) are set to:

- 1, if the U2000 Series power sensor non-volatile memory has failed.
- 0, for every other condition.

Operation Register Sets

The following registers contain information which is part of the U2000 Series power sensor 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 264.

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.

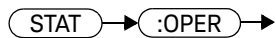
STATus:OPERation

The operation status register set contains conditions which are a part of the operation of the U2000 Series power sensor as a whole.

The following bits in these registers are used by the U2000 Series power sensor:

Bit Number	Decimal Weight	Definition
0	1	CALibrating summary
1 - 3	-	Not used
4	16	MEASuring summary
5	32	Waiting for TRIGger summary
6 - 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 always 0)

Syntax



STATus:OPERation:CALibrating[:SUMM]ary]

The operation status calibrating summary register set contains information on the calibrating status of the U2000 Series power sensor.

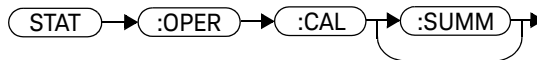
The following bits in these registers are used by the U2000 Series power sensor:

Bit Number	Decimal Weight	Definition
0	-	Not used
1	2	CALibrating status
2	4	Not used
3-15	-	Not used

These bits are set at the beginning of zeroing (**CALibration:ZERO:AUTO ONCE**). Also for the compound command/query **CALibration[:ALL]?**, this bit is set at the beginning of the calibration sequence.

These bits are 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 U2000 Series power sensor.

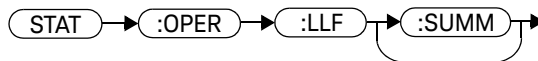
The following bits in these registers are used by the U2000 Series power sensor:

Bit Number	Decimal Weight	Definition
0	-	Not used
1	2	LLFail status
2	4	Not used
3	8	Not used
4	16	Not used
5	32	Not used
6	64	Not used
7-15	-	Not used

The appropriate bits are set if the lower limit test fails.

These bits are cleared if a measurement is made and the test is enabled and passes.

Syntax



STATus:OPERation:MEASuring[:SUMM]ary]

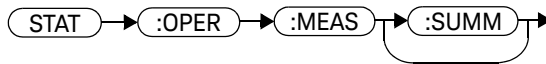
The operation status measuring summary register set contains information on the measuring status of the U2000 Series power sensor.

The following bits in these registers are used by the U2000 Series power sensor:

Bit Number	Decimal Weight	Definition
0	-	Not used
1	2	MEASuring status
2	4	Not used
3-15	-	Not used

These bits are set when the U2000 Series power sensor is taking a measurement. These bits are cleared when the measurement is finished.

Syntax



STATus:OPERation:SENSe[:SUMM]ary]

The operation status sense summary register set contains information on the status of the U2000 Series power sensor.

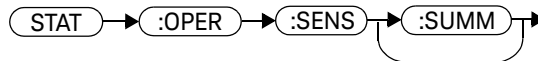
The following bits in these registers are used by the U2000 Series power sensor:

Bit Number	Decimal Weight	Definition
0	-	Not used
1	2	SENSe status
2	4	Not used
3-15	-	Not used

These bits are set when the U2000 Series power sensor is reading data from the U2000 Series power sensor non-volatile memory.

These bits are cleared when the U2000 Series power sensor is not reading data from the U2000 Series power sensor non-volatile memory.

Syntax



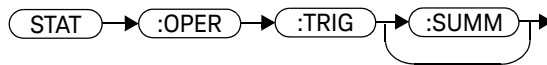
STATUS:OPERation:TRIGger[:SUMM]ary]

The operation status trigger summary register set contains information on the trigger status of the U2000 Series power sensor.

The following bits in these registers are used by the U2000 Series power sensor:

Bit Number	Decimal Weight	Definition
0	-	Not used
1	2	TRIGger status
2	4	Not used
3-15	-	Not used

Syntax



STATus:OPERation:ULFail[:SUMM]ary]

The operation status upper limit fail summary register set contains information on the upper limit fail status of the U2000 Series power sensor.

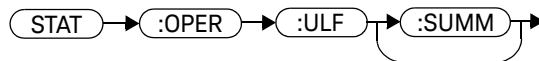
The following bits in these registers are used by the U2000 Series power sensor:

Bit Number	Decimal Weight	Definition
0	-	Not used
1	2	ULFail status
2	4	Not used
3	8	Not used
4	16	Not used
5	32	Not used
6	64	Not used
7-15	-	Not used

The appropriate bits are set if the upper limit test fails.

These bits are cleared if a measurement is made and the test is enabled and passes.

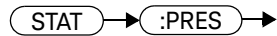
Syntax



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
OPERational	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 U2000 Series power sensor. 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 264.

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 U2000 Series power sensor.

The following bits in these registers are used by the U2000 Series power sensor:

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 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 power-on self-test fails, and cleared if it passes.

Syntax



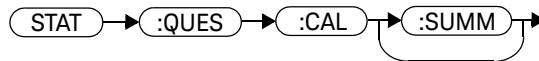
STATus:QUEStionable:CALibration[:SUMM]ary]

The questionable calibration summary register set contains bits which give an indication of the quality of the data produced by the U2000 Series power sensor due to its calibration status.

The following bits in these registers are used by the U2000 Series power sensor:

Bit Number	Decimal Weight	Definition
0	-	Not used
1	2	Summary of CALibration
2	4	Not used
3-15	-	Not used

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 U2000 Series power sensor.

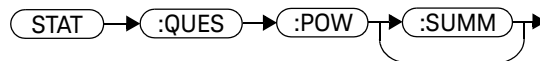
The following bits in these registers shall be used by the U2000 Series power sensor:

Bit Number	Decimal Weight	Definition
0	-	Not used
1	2	Power
2	4	Not used
3	8	Not used
4	16	Not used
5	32	Not used
6	64	Not used
7	128	Not used
8	256	Not used

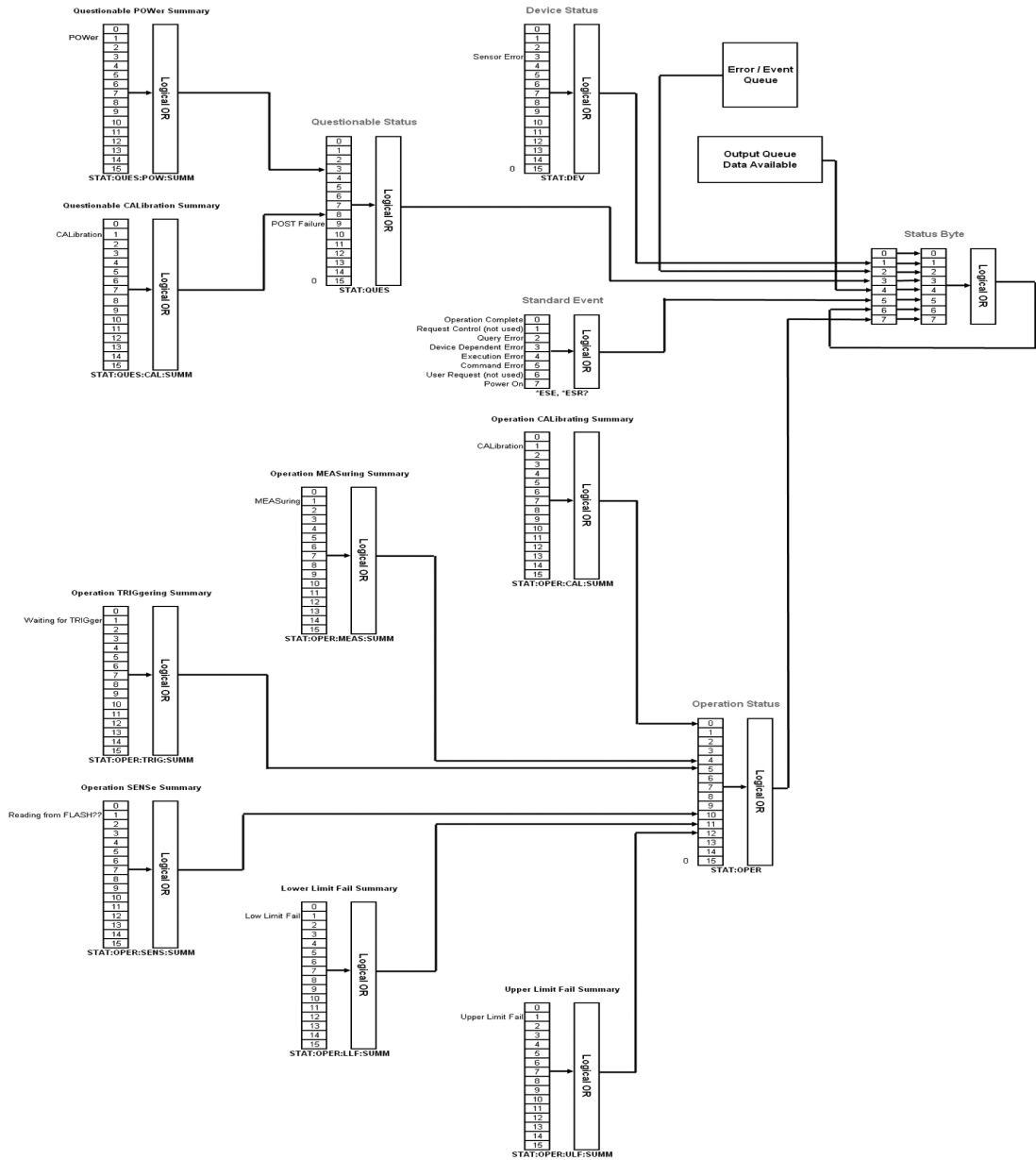
Bit 1 is set when error -231, "Data questionable; Input Overload" occurs.

This bit is cleared when no errors or events are detected by the U2000 Series power sensor during a measurement covering the causes given for it to set.

Syntax



Status Block Diagram



THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

10 SYSTEM Subsystem

SYSTEM Subsystem	286
SYSTEM:ERRor?	287
SYSTEM:HELP:HEADers?	294
SYSTEM:PERSonA:MANufacturer <"string">	295
SYSTEM:PERSonA:MANufacturer:DEFault	297
SYSTEM:PRESet <character_data>	298
SYSTEM:VERSion?	302

This chapter explains how to use the **SYSTEM** command subsystem to return error numbers and messages from the U2000 Series power sensors, preset the U2000 Series power sensors, and query the SCPI version.

SYSTem Subsystem

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

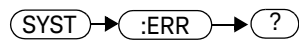
- Return error numbers and messages from the U2000 Series power sensor
- Preset the U2000 Series power sensor
- Query the SCPI version

Keyword	Parameter Form	Notes	Page
SYSTem			
:ERRor			page 287
:HELP			
:HEADers?		[query only]	page 294
:PRESet	<character_data>		page 298
:VERSion?		[query only]	page 302

SYSTEM:ERRor?

This query returns error numbers and messages from the U2000 Series power sensor error queue. When an error is generated by the U2000 Series power sensor, 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 ***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

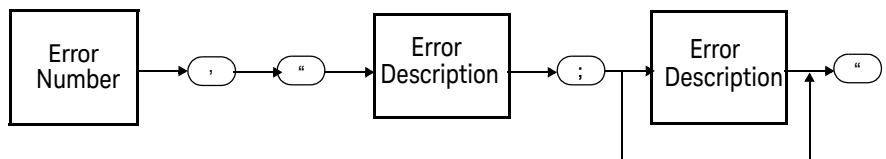


Query Example

SYSTEM:ERR?

Queries the system error.

Error queue messages have the following format:



Error Queue Message

For example, -330, "Self-test Failed;Battery Fault".

Errors are retrieved in a first in first out (FIFO) order. If more than 30 errors occur, the error queue overflows and the last error in the queue is replaced with the error -350, "Queue Overflow". Any time the queue is overflows the most recent error is discarded.

Example

SYST:ERR?

Queries the oldest error message stored in the U2000 Series power sensor 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	<p>Invalid character</p> <p>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.</p> <p>For example, LIM:LOW O#.</p>
-102	<p>Syntax error</p> <p>Invalid syntax was found in the command string.</p> <p>For example, LIM:CLE:AUTO, 1 or LIM:CLE: AUTO 1.</p>
-103	<p>Invalid separator</p> <p>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.</p> <p>For example, OUP:ROSC,1.</p>
-105	<p>GET not allowed</p> <p>A Group Execute Trigger (GET) is not allowed within a command string.</p>
-108	<p>Parameter not allowed</p> <p>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.</p> <p>For example, CAL 10.</p>
-109	<p>Missing parameter</p> <p>Fewer parameters were received than expected for the command. You omitted one or more parameters that are required for this command.</p> <p>For example, AVER:COUN.</p>
-112	<p>Program mnemonic too long</p> <p>A command header was received which contained more than the maximum 12 characters allowed.</p> <p>For example, SENS:e:AVERage:COUNT 8.</p>
-113	<p>Undefined header</p> <p>A command was received that is not valid for this U2000 Series power sensor. 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.</p> <p>For example, TRIG:SOUR IMM.</p>
-121	<p>Invalid character in number</p> <p>An invalid character was found in the number specified for a parameter value.</p> <p>For example, SENS:AVER:COUN 128#H.</p>

-123	<p>Exponent too large</p> <p>A numeric parameter was found whose exponent was larger than 32,000. For example, SENS:COUN 1E34000.</p>
-124	<p>Too many digits</p> <p>A numeric parameter was found whose mantissa contained more than 255 digits, excluding leading zeros.</p>
-128	<p>Numeric data not allowed</p> <p>A numeric value was received within a command which does not accept a numeric value. For example, MEM:CLE 24.</p>
-131	<p>Invalid suffix</p> <p>A unit was incorrectly specified for a numeric parameter. You may have misspelled the unit. For example, SENS:FREQ 200KZ.</p>
-134	<p>Suffix too long</p> <p>A unit used contained more than 12 characters. For example, SENS:FREQ 2MHZ.</p>
-138	<p>Suffix not allowed</p> <p>A unit was received following a numeric parameter which does not accept a unit. For example, INIT:CONT 0Hz.</p>
-148	<p>Character data not allowed</p> <p>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.</p>
-151	<p>Invalid string data</p> <p>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".</p>
-158	<p>String data not allowed</p> <p>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, LIM:STAT 'ON'.</p>
-161	<p>Invalid block data</p> <p>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.</p>

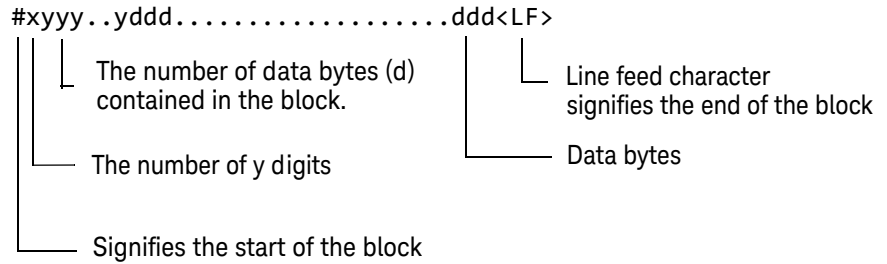
-168	<p>Block data not allowed</p> <p>A legal block data element was encountered but not allowed by the U2000 Series power sensor at this point.</p> <p>For example SYST:LANG #15FETC?</p>
-178	<p>Expression data not allowed</p> <p>A legal expression data was encountered but not allowed by the U2000 Series power sensor at this point.</p> <p>For example SYST:LANG (5+2).</p>
-211	<p>Trigger ignored</p> <p>Indicates that *TRG, or TRIG:IMM was received and recognized by the device but was ignored because the U2000 Series power sensor was not in the wait for trigger state.</p>
-213	<p>Init ignored</p> <p>Indicates that a request for a measurement initiation was ignored as the U2000 Series power sensor was already initiated.</p> <p>For example, INIT:CONT ON INIT.</p>
-214	<p>Trigger deadlock</p> <p>TRIG:SOUR was set to HOLD or BUS and a READ? or MEASure? was attempted, expecting TRIG:SOUR to be set to IMMEDIATE.</p>
-220	<p>Parameter error;Frequency list must be in ascending order.</p> <p>Indicates that the frequencies entered using the MEMory:TABLE:FREQuency command are not in ascending order.</p>
-221	<p>Settings conflict</p> <p>This message occurs under a variety of conflicting conditions. The following list gives a few examples of where this error may occur:</p> <p>If the READ? parameters do not match the current settings.</p> <p>If you are in fast mode and attempting to switch on for example, averaging, duty cycle or limits.</p> <p>Trying to clear a sensor calibration table when none is selected.</p>
-222	<p>Data out of range</p> <p>A numeric parameter value is outside the valid range for the command.</p> <p>For example, SENS:FREQ 2KHZ.</p>
-224	<p>Illegal parameter value</p> <p>A discrete parameter was received which was not a valid choice for the command. You may have used an invalid parameter choice.</p> <p>For example, TRIG:SOUR EXT.</p>

-226	Lists not same length This occurs when SENSE:CORRection:CSET2:STATe is set to ON and the frequency and calibration/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 U2000 Series power sensor state has changed such that the measurement is invalidated (for example, a change of frequency setting or triggering conditions).
-231	Data questionable;CAL ERROR U2000 Series power sensor calibration failed.
-231	Data questionable;ZERO ERROR U2000 Series power sensor zeroing failed. The most likely cause is attempting to zero when some power signal is being applied to the U2000 Series power sensor.
-310	System error;Sensor non-volatile memory Read Failed - critical data not found or unreadable This indicates a failure with your U2000 Series power sensor. Refer to your U2000 Series power sensor manual for details on returning it for repair.
-321	Out of memory The U2000 Series power sensor 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 your U2000 Series power sensor. Refer to Contacting Keysight Technologies for details of what to do with your faulty U2000 Series power sensor.
-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 off, or after *RST (reset) command has been executed.

-420	Query UNTERMINATED The U2000 Series power sensor 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 *!DN? 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 sensor. Data is returned in 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 10-1 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 alphabetical order.

Syntax



Example

SYST:HELP:HEAD?

Returns the SCPI commands supported by the sensor.

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.

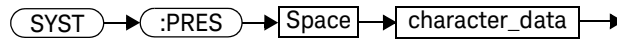
SYSTem:PRESet <character_data>

This command presets the U2000 Series power sensor to values appropriate for measuring the communications format specified by <character_data>. The U2000 Series power sensor is preset to 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 BURST

Example

SYST:PRESet DEF

This command presets the U2000 Series power sensor with default values. The same default values are set when the parameter is omitted.

Preset Values

DEfault

Table 10-1 shows the U2000 Series power sensor presets when <character_data> is set to DEfault or omitted. Values are shown for all SCPI commands:

Table 10-1 DEfault: U2000 Series power sensor presets

Command	Setting	Comments
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 SYS:PRES ON	U2000 Series power sensor in idle state U2000 Series power sensor in wait for trigger state
MEM:TABL:SEL	not affected	Active sensor calibration 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	Sensor frequency-dependent offset disabled
[SENS[1]:]CORR:FDOF GAIN4[:INP][:MAGN]	not affected	Return frequency-dependent offset
[SENS[1]:]CORR:GAIN2:STAT	OFF	Channel offset disabled

Command	Setting	Comments
[SENS[1]:]CORR:GAIN2[:INP][:MAGN]	0.0 dB	Enter channel offset value
[SENS[1]:]DET:FUNC	AVER	Measurement mode
[SENS[1]:]FREQ[:CW :FIX]	+50.000 MHz	Frequency setting
[SENS[1]:]MRAT	NORM	Measurement speed
[SENe[1]:]SWE[1]:OFFS:TIME	0	Set delay
[SENS[1]:]SWE[1]:TIME	500 ms	Set time gated period
[SENS[1]:]TRAC:OFFS:TIME	0	Delay
[SENS[1]:]:TRAC:TIME	500 ms	Duration of trace
[SENS[1]:]POW:AC:RANG	upper	Upper range selected
[SENS[1]:]POW:AC:RANG:AUTO	ON	Auto-ranging selected
TRAC[1]:STAT	OFF	Disable trace capture
TRAC[1]:UNIT	dBm	Trace units
TRIG[1]:DEL:AUTO	ON	Enable settling time delay
TRIG[:SEQ]:DEL	0	Delay between recognition of trigger event and start of a measurement
TRIG[:SEQ]:HOLD	1 μs	Trigger hold-off
TRIG[:SEQ]:HYST	0 dB	Fall/rise below/above TRIG:LEV
TRIG[:SEQ]:LEV	A model: 0 dBm B model: +30 dBm H model: +10 dBm	Power level
TRIG[:SEQ]:SLOP	POS	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	IMM	Trigger source set up
UNIT:POW	dBm	Power units

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

BURST

The burst setup provides pulse power measurement in average only mode with the following signal profile:

- Pulse width $\geq 30 \mu\text{s}$
- Pulse period $\leq 8 \text{ ms}$
- Duty cycle $\geq 1\%$

Commands are preset according to their **DEFault** values (refer to [Table 10-1](#) on page 299). However, it is recommended to set **[SENSe[1]:]AVERage:SDETect** to **OFF** and **[SENSe[1]:]AVERage:COUNT** to ≥ 256 for accurate measurement of pulsed signals.

For more information on the pulse power measurement in average only mode, refer to the *Keysight U2000 Series USB Power Sensors Operating and Service Guide*.

SYSTem:VERSion?

This query returns the version of SCPI used in the U2000 Series power sensor. 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 U2000 Series power sensor.

11 TRACe Subsystem

TRACe Subsystem	304
TRACe[1][:DATA]? <character_data>	305
TRACe[1]:STATE <boolean>	307
TRACe[1]:UNIT <character_data>	309

This chapter explains how to use the **TRACe** command subsystem to configure and capture the measured power trace.

TRACe Subsystem

The **TRACe** subsystem is used to:

- Specify the resolution type of trace to be captured.
- Enable/disable trace capture.
- Specify the trace units.

The following commands are described in this chapter:

Keyword	Parameter Form	Notes	Page
TRACe[1]			
[:DATA]?	<character_data>	[query only]	page 305
:STATe	<boolean>		page 307
:UNIT	<character_data>		page 309

NOTE

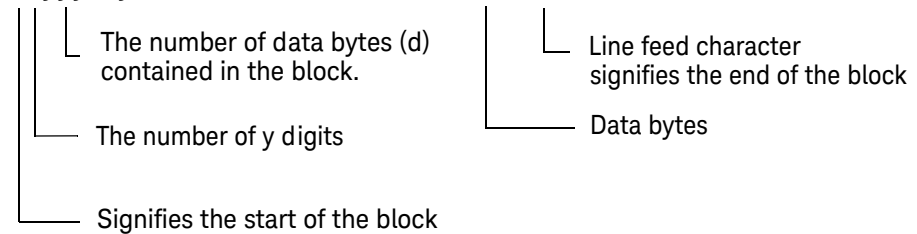
The sensor must be set to NORMa1 mode.

TRACe[1][:DATA]? <character_data>

This query returns trace data with the resolution that is determined by <character_data>.

Data is returned in IEEE 488.2 arbitrary block program data format as follows:

#xyyy...ydd.....ddd<LF>



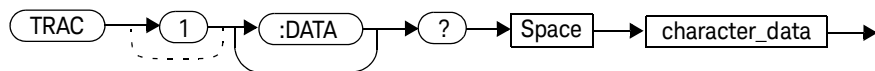
Example: if there are 12435 data bytes, $y = 12435$ and $x = 5$

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 MS byte is transmitted first. Each complete block is terminated by a line feed.

NOTE

TRACe data formatting is not affected by FORMat subsystem formatting.

Syntax



Parameters

Item	Description/Default	Range of Values
character_data	<ul style="list-style-type: none"> - HRESolution: high resolution. The complete capture buffer at the internal sample rate. The number of points in this trace is determined by the SENS:TRACe:TIME setting. - MRESolution: medium resolution. A subset of the capture buffer - the buffer contents are decimated to provide 1000 data points. - LRESolution: low resolution. A subset of the capture buffer - the buffer contents are decimated to provide 230 data points. 	<p>HRES MRES LRES</p>

Example

```
TRAC:DATA? HRES
```

This command returns the trace data at high resolution.

Reset Condition

On reset, the trace data is returned at low resolution.

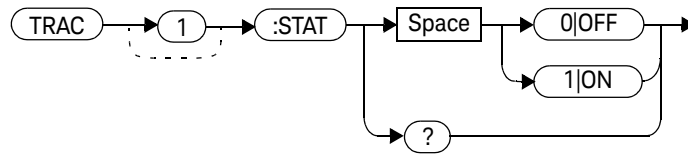
Error Messages

- If **TRAC:STAT** is off, the error -221, "Settings Conflict" occurs.
- If the command is used when U2004A model is connected, error -241, "Hardware missing" occurs.

TRACe[1]:STATe <boolean>

This command enables or disables trace capture for the sensor.

Syntax



Example

TRAC:STAT 1

This command enables trace capture for the sensor.

Reset Condition

On reset, trace capture is set to OFF.

Query

TRACe[1]:STATe?

The query command enters a 1 or 0 into the output buffer indicating whether or not trace capture is enabled or disabled.

- 1 is returned when trace capture is enabled
- 0 is returned when trace capture is disabled

Query Example

TRAC:STAT?

This command queries the current state of trace capture for the sensor.

Error Messages

- If the command is used when U2004A model is connected, error -241, “Hardware missing” occurs.
- If the sensor is set to **AVERage** mode rather than **NORMa1** mode, error -221, “Settings conflict” occurs.

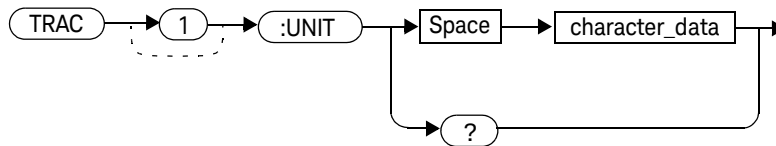
TRACe[1]:UNIT <character_data>

This command sets the units of the trace for the sensor.

NOTE

This command is included for compatibility purposes only. It has the same purpose as [SENSe[1]]:TRACe:UNIT <character_data>, which should be the preferred command.

Syntax



Parameters

Item	Description/Default	Range of Values
character_data	- DBM: dBm - W: Watt	DBM W

Example

TRAC:UNIT W

This command sets the trace units to Watt.

Reset Condition

On reset, the units are set to dBm.

Query

TRACe[1]:UNIT?

The query command returns the current setting of the trace units.

Query Example

TRAC:UNIT?

This command queries the current trace units for the sensor.

12 TRIGger Subsystem

TRIGger Subsystem	312
ABORT[1]	314
INITiate Commands	315
INITiate[1]:CONTInuous <boolean>	316
INITiate[1]:IMMEDIATE	318
INITiate[1]:CONTInuous:ALL <boolean>	319
INITiate[1]:CONTInuous:SEquence[1] <boolean>	321
INITiate[1]:IMMEDIATE:ALL	323
INITiate[1]:IMMEDIATE:SEquence[1]	324
TRIGger Commands	325
TRIGger[1]:DELay:AUTO <boolean>	326
TRIGger[1]:IMMEDIATE	328
TRIGger[1]:SOURce BUS INTerna[1] EXTerna HOLD IMMEDIATE	329
TRIGger[:SEquence]:DELay <numeric_value>	332
TRIGger[:SEquence]:HOLDoff <numeric_value>	334
TRIGger[:SEquence]:HYSTEResis <numeric_value>	336
TRIGger[:SEquence]:LEVel <numeric_value>	338
TRIGger[:SEquence]:SLOPe <character_data>	340
TRIGger[:SEquence[1]]:COUNT <numeric_value>	342
TRIGger[:SEquence[1]]:DELay:AUTO <boolean>	344
TRIGger[:SEquence[1]]:IMMEDIATE	346
TRIGger[:SEquence[1]]:SOURce BUS INTerna[1] EXTerna HOLD IMMEDIATE	347

This chapter explains how the **TRIGger** command subsystem is used to synchronize device actions with events.

TRIGger Subsystem

The **TRIGger** 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.

Keyword	Parameter Form	Notes	Page
ABOrt[1]		[no query] [non-SCPI]	page 314
INITiate[1]			
:CONTInuous			
:ALL	<boolean>		page 319
:SEQuence[1]	<boolean>		page 321
[:IMMediate]			
:ALL		[no query]	page 323
:SEQuence[1]		[no query]	page 324
TRIGger[1]			
:DELay			
:AUTO	<boolean>		page 326
[:IMMediate]		[no query]	page 328
:SOURce	BUS INTernal[1] EXTernal HOLD IMMediate		page 329
TRIGger			
[:SEQuence]			
:DELay	<numeric_value>		page 334
:HOLDoff	<numeric_value>		page 334
:HYSTeresis	<numeric_value>		page 336
:LEVel	<numeric_value>		page 338
:SLOPe	<character_data>		page 340

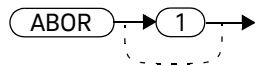
Keyword	Parameter Form	Notes	Page
<code>[:SEQuence[1]]</code>			
<code>:COUNt</code>	<code><numeric_value></code>		page 342
<code>:DELay</code>			
<code>:AUTO</code>	<code><boolean></code>		page 344
<code>:IMMediate</code>		[no query]	page 346
<code>:SOURce</code>	<code>BUS INTernal[1] EXTernal HOLD IMMediate</code>		page 347

ABORt[1]

This command removes the sensor 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 sensor in the idle state.

INITiate Commands

Initiate commands allow you to place the U2000 Series power sensor in the wait for trigger state.

The **INITiate** commands are overlapped, that is, the U2000 Series power sensor can continue parsing and executing subsequent commands while initiated. Note that the pending operation flag is set, when the U2000 Series power sensor 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:CONTinuous:ALL <boolean>
```

```
INITiate:CONTinuous:SEquence[1] <boolean>
```

```
INITiate[:IMMediate]:ALL
```

```
INITiate[:IMMediate]:SEquence[1]
```

INITiate[1]:CONTinuous <boolean>

This command sets the U2000 Series power sensor for either a single trigger cycle or continuous trigger cycles. A trigger cycle means that the U2000 Series power sensor exits the wait for trigger state and starts a measurement.

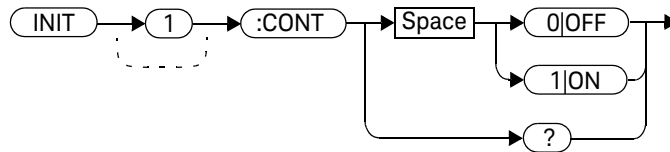
If **INITiate[1]:CONTinuous** is set to:

- **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:CONTinuous:SEquence[1]** <boolean>.

Syntax



Example

INIT:CONT ON

This command places the sensor in the wait for trigger state.

Reset Condition

On reset (***RST**), this command is set to **OFF**.

On preset (**SYSTEM:PRESet**) and sensor 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 sensor is set for single or continuous triggering.

INITiate[1][:IMMEDIATE]

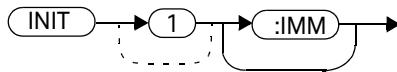
This command sets the U2000 Series power sensor in the wait for trigger state. When a trigger is received, the measurement is taken and the result placed in the U2000 Series power sensor memory. If **TRIGger[1]:SOURce** is set to **IMMEDIATE** the measurement begins as soon as **INITiate[1][:IMMEDIATE]** is executed.

Use **FETCh?** to transfer a measurement from memory to the output buffer. Refer to [“FETCh\[1\]? Query”](#) on page 93 for further details.

NOTE

This command performs the same function as **INITiate:[IMMEDIATE]:SEQUENCE[1]**.

Syntax



Example

INIT:IMM

This command places the sensor in the wait for trigger state.

Error Messages

If the U2000 Series power sensor is not in the idle state or **INITiate[1]:CONTinuous** is **ON**, error -213, "INIT ignored" occurs.

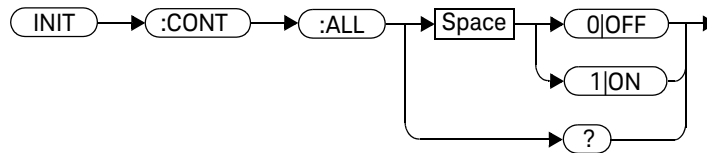
INITiate[1]:CONTinuous:ALL <boolean>

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 sensor power-up, **INITiate[1]:CONTinuous:ALL** 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 sensor is in a wait for trigger state.

INITiate[1]:CONTinuous:SEQuence[1] <boolean>

This command sets the U2000 Series power sensor for either a single trigger cycle or continuous trigger cycles. A trigger cycle means that the U2000 Series power sensor exits the wait for trigger state and starts a measurement.

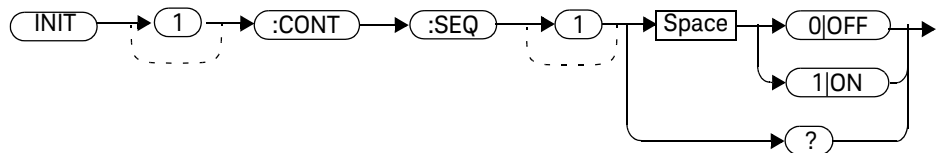
If **INITiate[1]:CONTinuous:SEQuence[1] <boolean>** is set to:

- **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 sensor in a wait for trigger state.

Reset Condition

On reset (***RST**), this command is disabled.

On preset (**SYSTem:PRESet**) and sensor 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 sensor 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 U2000 Series power sensor 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 U2000 Series power sensor in the wait for trigger state. When a trigger is received, the measurement is taken and the result placed in the U2000 Series power sensor 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 93 for further information.

NOTE

This command performs the same function as **INITiate[1]:[IMMediate]**.

Syntax



Example

INIT:SEQ

This command places the sensor in the wait for trigger state.

Error Messages

If the U2000 Series power sensor 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|INTernal[1]|EXTernal|HOLD|IMMediate

TRIGger[:SEQuence]:DELay <numeric_value>

TRIGger[:SEQuence]:HOLDoff <numeric_value>

TRIGger[:SEQuence]:HYSTEResis <numeric_value>

TRIGger[:SEQuence]:LEVel <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|INTernal[1]|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.

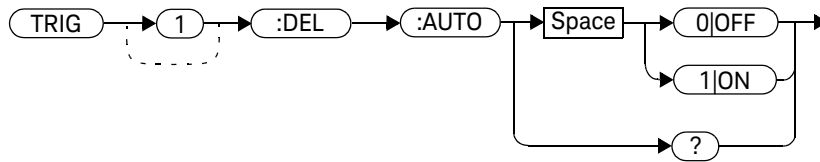
When this command is set to:

- **ON**, the U2000 Series power sensor 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 U2000 Series power sensor makes the measurement immediately after a trigger is received.

Syntax



Example

TRIG:DEL:AUTO ON

This command enables a delay on the sensor.

Reset Condition

On reset, the **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**

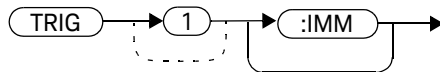
TRIGger[1][:IMMediate]

This command causes a trigger to occur immediately, provided the sensor is in the wait for trigger state. When this command is executed, the measurement result is stored in the U2000 Series power sensor 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 sensor trigger to occur immediately.

Error Messages

If the U2000 Series power sensor is not in the wait for trigger state, then **TRIGger[1][:IMMediate]** causes error -211, "Trigger ignored".

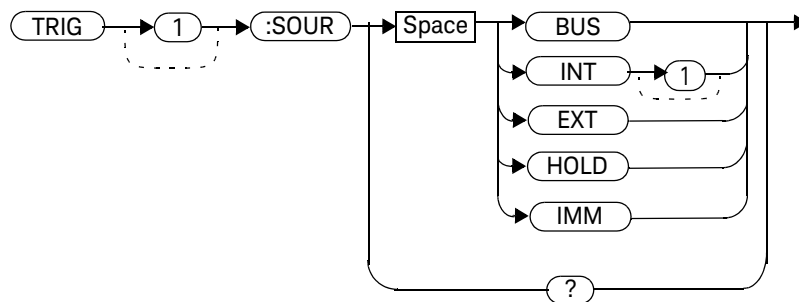
TRIGger[1]:SOURce BUS|INTerna[1]|EXTerna|HOLD|IMMEDIATE

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 U2000 Series power sensor 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|INTerna[1]|EXTerna|HOLD|IMMEDIATE** which should be used in preference.

Syntax



Parameters

Item	Description/Default	Range of Values
source	<p>Available trigger sources:</p> <ul style="list-style-type: none"> – BUS: the trigger source is a *TRG common command or the TRIGGER[1][:IMMEDIATE] SCPI command – INTernal: internal triggering – EXTernal: the trigger source is the trigger input in the sensor – HOLD: triggering is suspended. The only way to trigger the U2000 Series power sensor is to use TRIGger[1][:IMMEDIATE]. – IMMEDIATE: the trigger system is always true. If INITiate[1]:CONTinuous is ON, the U2000 Series power sensor is continually triggering free (free run mode). If an INITiate[1][:IMMEDIATE] command is sent, a measurement is triggered then the U2000 Series power sensor returns to the idle state. 	<p>BUS</p> <p>INTernal[1]</p> <p>EXTernal</p> <p>HOLD</p> <p>IMMEDIATE</p>

NOTE

The trigger source is set to IMMEDIATE on sensor 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 sensor for immediate triggering.

Reset Condition

On reset, the trigger source is set to IMMEDIATE.

Query

TRIGger:SOURCE?

The query returns the current trigger source, either **IMM**, **BUS**, **INT**, **EXT**, or **HOLD**.

Query Example

TRIG: SOUR?

Queries the sensor trigger source.

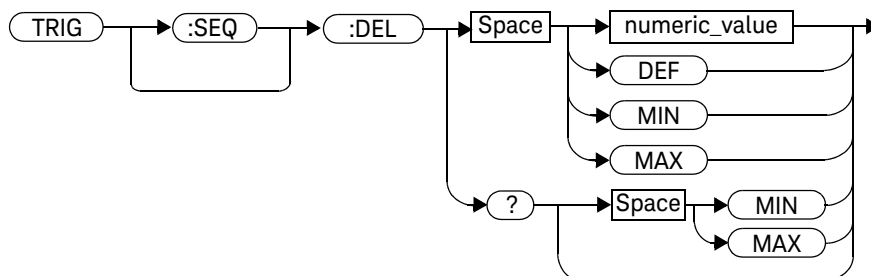
Error Message

- If the source is changed to **INT** or **EXT** while **[SENSe[1]:]MRATe** is set to **FAST** or **[SENSe[1]:]SPEEd** is set to **110**, error -221, "Settings conflict" occurs.
- If the source is changed to **INT** or **EXT** and **SENSe:DETEctor:FUNCTion** is set to **AVERAge**, error -221, "Settings conflict" occurs.
- If the source is changed to **INT** or **EXT** when U2004A power sensor is connected, error -241, "Hardware missing" occurs.

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 : -0.15 s - MAX : 0.15 s Units are resolved to 1 μ s.	-0.15 to 0.15 s DEF MIN MAX

Example

TRIG:SEQ:DEL 0.001

This command sets a delay of 1 ms for the sensor.

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

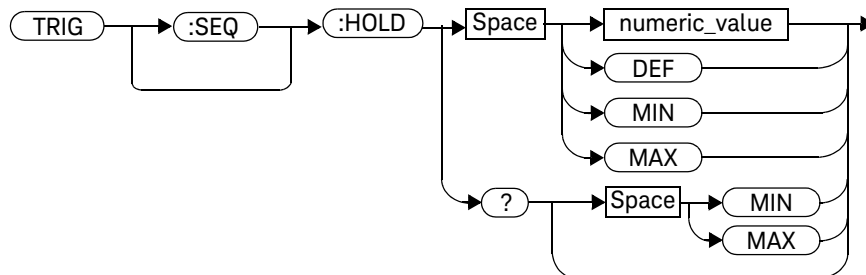
TRIG:SEQ:DEL?

This command queries the delay.

TRIGger[:SEquence]:HOLDOff <numeric_value>

This command sets the trigger hold-off in seconds.

Syntax



Parameters

Item	Description/Default	Range of Values
numeric_value	The trigger hold-off in seconds. - DEF : the default value is 1 μ s - MIN : 1 μ s - MAX : 400 ms Units are resolved to 1 μ s.	1 μ s to 400 ms DEF MIN MAX

Example

TRIG:SEQ:HOLD 0.1

This command sets the trigger holdoff to 100 ms.

Reset Condition

On reset, the trigger hold-off is set to 1 μ s.

Query

TRIGger[:SEQuence]:HOLDoff? [MIN|MAX]

The query returns the current trigger holdoff setting or the values associated with **MIN** or **MAX**.

Query Example

TRIG:SEQ:HOLD?

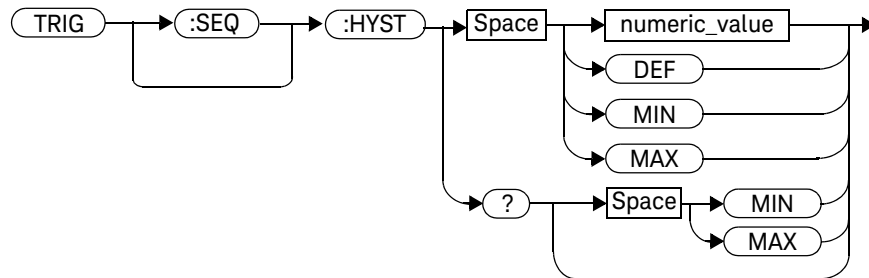
This command queries the trigger holdoff setting.

TRIGger[:SEquence]:HYSTeresis <numeric_value>

This command sets:

- How far a signal must fall below **TRIG:LEVe1** before a rising edge can be detected.
- How far a signal must rise above **TRIG:LEVe1** before a falling edge can be detected.

Syntax



Parameters

Item	Description/Default	Range of Values
numeric_value	<p>How far a signal must fall/rise before a rising or falling edge can be detected.</p> <ul style="list-style-type: none"> – DEF: the default value is 0 dB – MIN: 0 dB – MAX: 3 dB <p>Units are resolved to 0.1 dB.</p>	<p>0 to 3 dB</p> <p>DEF MIN MAX</p>

Example

TRIG:SEQ:HYST 2 *This command sets the value to 2 dB.*

Reset Condition

On reset, the value is set to 0 dB.

Query

TRIGger[:SEQuence]:HYSTeris? [MIN|MAX]

The query returns the current value or the values associated with **MIN** or **MAX** in dB.

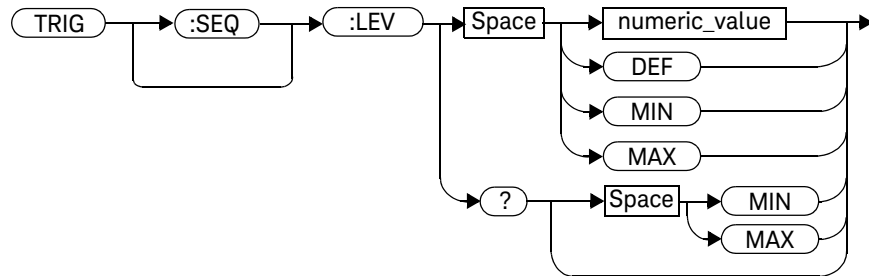
Query Example

TRIG:SEQ:HYST? *This command queries the value.*

TRIGger[:SEquence]:LEVel <numeric_value>

This command sets the power level at which a trigger event is recognized.

Syntax



Parameters

Item	Description/Default	Range of Values ^[a]
numeric_value	<p>The power level at which a trigger event is recognized.</p> <ul style="list-style-type: none"> - DEF: sensor dependent <ul style="list-style-type: none"> - A model: 0 dBm - B model: +30 dBm - H model: +10 dBm - MIN: sensor dependent <ul style="list-style-type: none"> - A model: -50 dBm - B model: -20 dBm - H model: -40 dBm - MAX: sensor dependent <ul style="list-style-type: none"> - A model: +20 dBm - B model: +44 dBm - H model: +30 dBm <p>Units are resolved to 0.1 dBm.</p>	<p>A model: -50 to +20 dBm B model: -20 to +44 dBm H model: -40 to +30 dBm DEF MIN MAX</p>

[a] If a channel offset has been previously set, a higher numeric value is permitted. See "Setting Offsets" on page 47 for more information.

Example

TRIG:SEQ:LEV 10

This command sets the power level for a trigger event to 10 dBm.

Reset Condition

On reset, the power level is set to 0 dBm (A model), +30 dBm (B model), or +10 dBm (H model).

Query

TRIGger[:SEQuence]:LEVe1? [MIN|MAX]

The query returns the current power level setting or the values associated with **MIN** or **MAX**.

Query Example

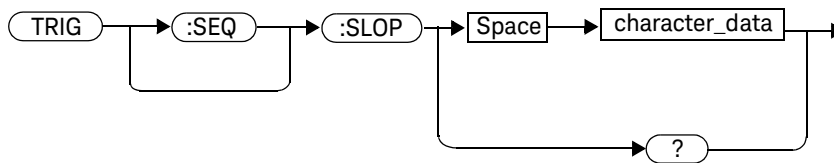
TRIG:SEQ:LEV?

This command queries the power level setting.

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 sensor.*

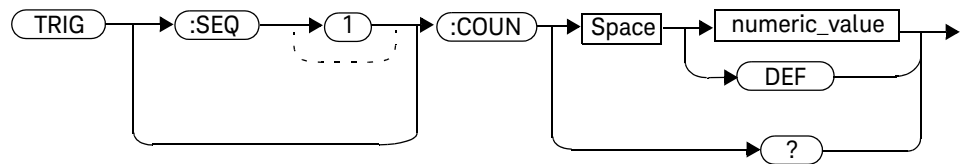
TRIGger[:SEquence[1]]:COUNT <numeric_value>

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 are performed. That is, **COUNT** measurements are performed in response to **COUNT** trigger events.

COUNT can be set to a value >1 only when:

- [SENSe[1]:]MRATe <character_data> is set to FAST
- and
- TRIGger[1]:SOURce set to IMMEDIATE or HOLD.

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 50 DEF

Example

TRIG:COUN 10

This command sets the number of triggered events to 10 for the sensor 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 sensor.

Query Example

TRIG:COUN?

Queries the number of triggered events for the sensor measurement cycle.

Error Messages

If **COUNT** >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.

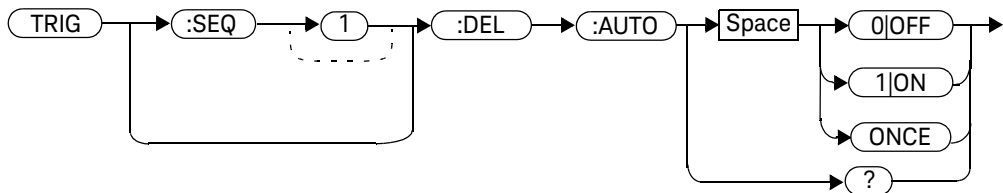
When this command is set to:

- **ON**, the U2000 Series power sensor 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 U2000 Series power sensor makes the measurement immediately a trigger is received.

Syntax



Example

TRIG:DEL:AUTO ON

This command enables a delay on the sensor.

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

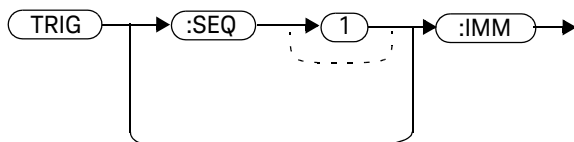
TRIG:DEL:AUTO?

Queries the settling-time delay of the sensor.

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 sensor stops waiting for a trigger and takes a measurement ignoring any delay set by **TRIG:DELaY**.

Syntax



Example

TRIG:IMM

This command initiates a measurement on the sensor.

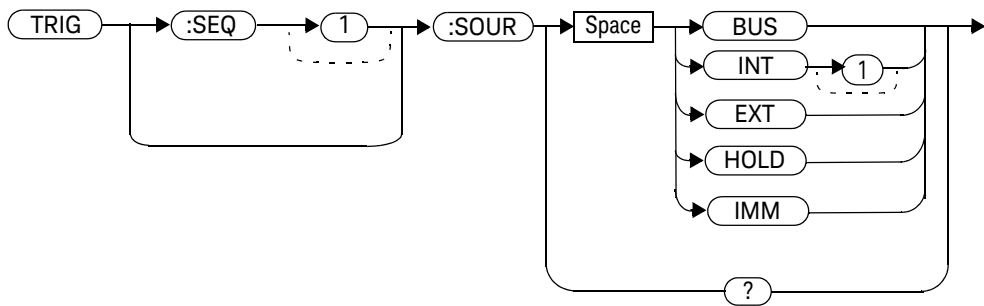
TRIGger[:SEQuence[1]]:SOURce BUS|INTernal[1]|EXTernal|HOLD|IMMEDIATE

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 U2000 Series power sensor in the wait for trigger state.

NOTE

This command has the same purpose as TRIGger[1]:SOURce
BUS|INTernal[1]|EXTernal|HOLD|IMMEDIATE.

Syntax



Parameters

Item	Description/Default	Range of Values
source	<p>Available trigger sources:</p> <ul style="list-style-type: none"> – BUS: the trigger source is a *TRG common command or the TRIGGER[1][:IMMEDIATE] SCPI command. – INTernal: internal triggering – EXTernal: the trigger source is the trigger input in the sensor – HOLD: triggering is suspended. The only way to trigger the U2000 Series power sensor is to use TRIGGER[1][:IMMEDIATE]. – IMMEDIATE: the trigger system is always true. If INITiate[1]:CONTinuous is ON the U2000 Series power sensor is continually triggering free (free run mode). If an INITiate[1][:IMMEDIATE] command is sent, a measurement is triggered and then the U2000 Series power sensor returns to the idle state. 	<p>BUS</p> <p>INTernal</p> <p>EXTernal</p> <p>HOLD</p> <p>IMMEDIATE</p>

NOTE

The trigger source is set to IMMEDIATE on sensor 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 sensor 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 sensor.

Error Message

- If the source is changed to **INT** or **EXT** while **[SENSe[1]:]MRATE** is set to **FAST** or **[SENSe[1]:]SPEEd** is set to **110**, error -221, "**Settings conflict**" occurs.
- If the source is changed to **INT** or **EXT** and **SENSe:DETEctor:FUNCTion** is set to **AVERAge**, error -221, "**Settings conflict**" occurs.
- If the source is changed to **INT** or **EXT** when U2004A power sensor is connected, error -241, "**Hardware missing**" occurs.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

13 UNIT Subsystem

UNIT Subsystem [352](#)

UNIT[1]:POWer <amplitude_unit> [353](#)

This chapter explains how the **UNIT** command subsystem is used to set the U2000 Series power sensor measurement units to Watts or dBm.

UNIT Subsystem

The **UNIT** command subsystem sets power measurement units to dBm or Watts.

The following command is described in this section:

Keyword	Parameter Form	Notes	Page
UNIT[1]			
:POWer	<amplitude_unit>		page 353

The **UNIT[1]:POWer** command is as follows:

- If **UNIT[1]:POWer** is set to dBm.
- If **UNIT[1]:POWer** is set to W.

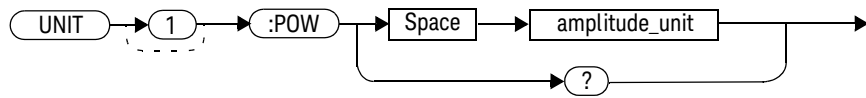
UNIT[1]:POWer <amplitude_unit>

This command sets the power measurement units. The power measurement unit set by **UNIT[1]:POWer** is used for any command which accepts a numeric value in more than one unit.

For single measurement:

- **UNIT[1]:POWer** sets the power measurement units.

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 units.

Reset Condition

On reset, all measurements are set to **DBM**.

Query

UNIT[1]:POWER?

The query returns the current setting of the power measurement units.

Query Example

UNIT:POW?

Queries which measurement units are being used on the current measurement.

14 IEEE 488.2 Command Reference

SCPI Compliance Information	356
*CLS	357
*ESE <Nrf>	358
*ESR?	360
*IDN?	361
*OPC	362
*OPT?	363
*RCL <Nrf>	364
*RST	365
*SAV <Nrf>	366
*SRE <Nrf>	367
*STB?	369
*TRG	371
*TST?	372
*WAI	373
USBTCM/USB488 Universal Commands	374

This chapter contains information on the IEEE 488.2 Common Commands supported by the U2000 Series power sensors.

SCPI Compliance Information

This chapter contains information on the IEEE-488 Common Commands that the U2000 Series power sensor 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 alphabetical order.

*CLS	Clear Status	page 357
*ESE and *ESE?	Event Status Enable	page 358
*ESR?	Event Status Register	page 360
*IDN?	Identify	page 361
*OPC and *OPC?	Operation Complete	page 362
*OPT?	Options	page 363
*RCL	Recall	page 364
*RST	Reset	page 365
*SAV	Save	page 366
*SRE and *SRE?	Service Request Enable	page 367
*STB?	Status Byte	page 369
*TRG	Trigger	page 371
*TST?	Test	page 372
*WAI	Wait	page 373

*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

***CLS** →

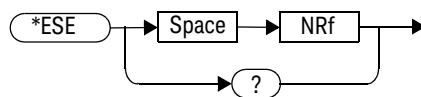
***ESE <NRf>**

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	Not used
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 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	Not used
2	4	Query Error
3	8	Device Dependent Error
4	16	Execution Error
5	32	Command Error
6	64	Not used
7	128	Power On

Syntax



*IDN?

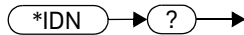
The ***IDN?** query allows the U2000 Series power sensor to identify itself. The string returned is:

Keysight Technologies,U2000A,<serial number>,A1.XX.YY

where:

- **<serial number>** uniquely identifies each U2000 Series power sensor.
- **A1.XX.YY** represents the firmware revision with **XX** and **YY** representing the major and minor revisions respectively.

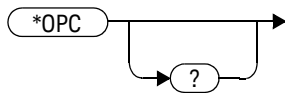
Syntax



*OPC

The ***OPC** (OPeration Complete) command causes the U2000 Series power sensor to set the operation complete bit in the Standard Event Status Register when all pending device operations have completed.

Syntax



Query

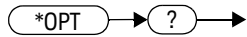
***OPC?**

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 U2000 Series power sensor and returns " " empty string for a standard sensor.

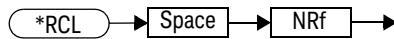
Syntax



*RCL <NRf>

The ***RCL <NRf>** (ReCaLL) command restores the state of the U2000 Series power sensor 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 U2000 Series power sensor in a known state. Refer to “**SYSTem:PRESet <character_data>**” on page 298 for information on reset values.

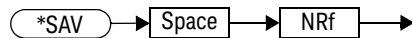
Syntax

***RST** →

***SAV <NRf>**

The ***SAV <NRf>** (SAVe) command stores the current state of the U2000 Series power sensor in the specified register.

Syntax



Parameters

Item	Description/Default	Range of Values
NRf	The number of the register that the current state of the U2000 Series power sensor is to be saved to.	1 to 10

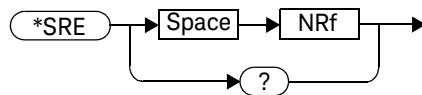
***SRE <NRf>**

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 the bits 0 to 5 and bit 7 of the Service Request Enable Register. Bit 6 is always 0. [Table 14-3](#) shows the contents of this register. Refer to the pullout at the end of Chapter 10 for further information.

Table 14-3 *SRE mapping

Bit	Weight	Meaning
0	1	Not used
1	2	Not used
2	4	Device Dependent
3	8	QUESTionable Status Summary
4	16	Message Available
5	32	Event Status Bit
6	64	Not used
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 bits 0 to 5 and bit 7 of the Service Request Enable Register. The format of the return is **<NR1>** in the ranges of 0 to 63 or 128 to 191 (that is, bit 6 is always 0).

*STB?

The ***STB?** (Status Byte) query returns bit 0 to 5 and bit 7 of the U2000 Series power sensor 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 ranges of 0 to 255. [Table 14-4](#) shows the contents of this register. Refer to the Status Block Diagram at the end of [Chapter 9, "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 - U2000 Series power sensor 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 U2000 Series power sensor that are in the wait for trigger state.

Syntax

***TRG** →

Error Message

- If **TRIGger[1]:SOURce** is not set to **BUS**, error -211, "Trigger ignored" occurs.
- If the U2000 Series power sensor is not in the wait-for-trigger state, error -211, "Trigger ignored" occurs.

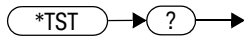
*TST?

The ***TST?** (TeST) query causes the U2000 Series power sensor to perform a self-test. The test takes approximately 40 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** (WAI) command causes the U2000 Series power sensor 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

***WAI** →

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 U2000 Series power sensor:

- All pending operations are halted, that is, ***OPC?** and ***WAI**.
- 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

Example 1: Identifying the U2000 Series Power Sensor In Use	376
Example 2: Internal/External Zeroing	377
Example 3: FETCh, MEASure, and READ Queries	378
Example 4: CW Power Measurement from +44 dBm to -60 dBm	380
Example 5: Average Modulated Power Measurement	383
Example 6: Average Pulse Power Measurement with Duty Cycle	384
Example 7: Time-Gated Burst Power Measurement via Internal/ External Trigger	385
Example 8: Creating a Trace Graph Display	387
Example 9: Acquiring 100, 250, and 1000 Readings/s	388
Example 10: Frequency-Dependent Offset	390
Example 11: Power Sweep Operation	391
Example 12: Frequency Sweep Operation	394

This chapter displays the programming sequences or examples to remotely control the U2000 Series power sensors using SCPI commands.

Example 1: Identifying the U2000 Series Power Sensor In Use

You can verify whether you are communicating with the right U2000 Series power sensor. Refer to “[Error Messages](#)” on page 288 for more information.

```
-> *IDN? // Queries the identification of the
          U2000 Series power sensor and
          checks whether you are
          communicating with the right
          U2000 Series power sensor.

-> SYST:ERR? // Checks the U2000 Series
              power sensor system error
              queue.
```

NOTE

“->” indicates the commands that you send to the U2000 Series power sensor.

“<-” indicates the response from the U2000 Series power sensor.

Example 2: Internal/External Zeroing

Internal zeroing can be performed without removing the RF input signal from the U2000 Series power sensor.

```
-> CAL:ZERO:TYPE INT // Performs internal zeroing.
-> CAL // Initializes calibration.
-> *OPC? // Waits for the operation to complete.
<- 1 // Returns a 1 when internal zeroing has
      completed.
```

During the external zeroing operation, ensure that the RF input signal is turned off or the device under test is disconnected from the U2000 Series power sensor.

```
-> CAL:ZERO:TYPE EXT // Performs external zeroing.
-> CAL // Initializes calibration.
-> *OPC? // Waits for the operation to complete.
<- 1 // Returns a 1 when external zeroing has
      completed.
```

NOTE

“->” indicates the commands that you send to the U2000 Series power sensor.

“<-” indicates the response from the U2000 Series power sensor.

Example 3: FETCh, MEASure, and READ Queries

There are three different ways to query the power measurement results using **FETC?**, **MEAS?**, and **READ?**.

In **Free Run** or **Continuous** mode, you can use either **FETC?** or **MEAS?** to query the power measurement results.

```
-> INIT:CONT ON // Sets the U2000 Series power
                  sensor to Free Run mode.
-> FETC? // Queries the measurement
          results from the buffer.
```

or

```
-> INIT:CONT ON // Sets the U2000 Series power
                  sensor to Free Run mode.
-> MEAS? // Reads the measurement
          results. MEAS? is equivalent to
CONF followed by a READ?.
```

In **Single Trigger** mode, you can use **FETC?**, **MEAS?**, or **READ?** to query the power measurement.

```
-> INIT:CONT OFF // Sets the U2000 Series power
                  sensor to 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 U2000 Series power
                  // sensor to Single Trigger mode.
-> MEAS?          // Reads the measurement
                  // results. MEAS? is equivalent to
                  // CONF followed by a READ?.

```

or

```

-> INIT:CONT OFF // Sets the U2000 Series power
                  // sensor to 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 U2000 Series power sensor.

“<-” indicates the response from the U2000 Series power sensor.

Example 4: CW Power Measurement from +44 dBm to –60 dBm

The SCPI programming sequence examples for CW power measurement from +44 dBm to –60 dBm with Free Run and Single Trigger modes are shown as follows.

Free Run

```
-> SYST:PRES DEF // Presets the U2000 Series
                    power sensor.
->  FREQ 1000MHZ // Sets the frequency (for
                    example: 1000 MHz).
->  CAL:ZERO:TYPE EXT // Performs external zeroing.
->  CAL // Initializes the calibration.
->  *OPC? // Waits for the operation to
                    complete.
<- 1 // Returns a 1 when external
                    zeroing has completed.
->  INIT:CONT ON // Sets the U2000 Series power
                    sensor to Free Run mode.
```

NOTE

It is strongly advisable to perform external zeroing on the U2000 Series power sensor for power measurement level below –30 dBm for accurate measurements. During the external zeroing process, the RF input signal must be switched off or the device-under-test disconnected from the U2000 Series power sensor.

Signal level from +44 dBm to –45 dBm:

```
-> FETC? // Queries the measurement
                    results.
```

NOTE

It is advisable to use FETC? to query the measurement results.

Signal level from -46 dBm to -60dBm:

```
-> MEAS? // Measures low power from -46 dBm to -60 dBm.
```

NOTE

MEAS? can also be used but it will trade off the measurement speed.

Single Trigger

Signal level from +44 dBm to -60 dBm:

```
-> SYST:PRES DEF // Presets the U2000 Series power sensor.
-> FREQ 1000MHz // Sets the frequency (for example: 1000 MHz).
-> CAL:ZERO:TYPE EXT // Performs external zeroing.
-> CAL // Initializes calibration.
-> *OPC? // Waits for the operation to complete.
<- 1 // Returns a 1 when external zeroing has completed.
-> INIT:CONT OFF // Sets the U2000 Series power sensor to Single Trigger mode.
```

NOTE

It is strongly advisable to perform external zeroing on the U2000 Series power sensor for power measurement level below -30 dBm for accurate measurements. During the external zeroing process, the RF input signal must be switched off or the device-under-test disconnected from the U2000 Series power sensor.

```
-> INIT // Initializes the measurement.  
-> FETC? // Queries the measurement  
         results.
```

NOTE

In Single Trigger mode, INIT must be executed before FETC?.

or

```
-> MEAS? // Reads the measurement  
         results.
```

NOTE

In Single Trigger mode, MEAS? can be used without executing INIT.

or

```
-> CONF // Configures the measurement.  
-> READ? // Reads the measurement  
         results.
```

NOTE

In Single Trigger mode, READ? can be used without executing INIT.

NOTE

“->” indicates the commands that you send to the U2000 Series power sensor.
“<-” indicates the response from the U2000 Series power sensor.

Example 5: Average Modulated Power Measurement

The SCPI programming sequence example for average modulated power measurement is shown as follows.

You are recommended to turn off the RF signal source before proceeding.

```
-> SYST:PRES BURST           // Presets the U2000 Series
                               power sensor to burst mode.
->  FREQ 1000MHz              // Sets the frequency (for
                               example: 1000 MHz).
->  INIT:CONT ON              // Sets the U2000 Series power
                               sensor to Free Run mode.
->  FETC?                     // Queries the measurement
                               results.
```

For more information on how to query the measurement results, refer to [“Example 3: FETCH, MEASure, and READ Queries”](#) on page 378.

NOTE

“->” indicates the commands that you send to the U2000 Series power sensor.

“<-” indicates the response from the U2000 Series power sensor.

Example 6: Average Pulse Power Measurement with Duty Cycle

The SCPI programming sequence example for average pulse power measurement with duty cycle is shown as follows.

```

-> SYST:PRES BURST           // Presets the U2000 Series
                             // power sensor to burst mode.
-> FREQ 1000MHZ             // Sets the frequency (for
                             // example: 1000 MHz).
-> CORR:DCYC:STAT 1        // Enables duty cycle.
-> CORR:DCYC 50            // Sets the duty cycle (for
                             // example: 50%).
-> INIT:CONT ON            // Sets the U2000 Series power
                             // sensor to Free Run mode.
-> FETC?                   // Queries the measurement
                             // results.

```

For more information on how to query the measurement results, refer to [“Example 3: FETCh, MEASure, and READ Queries”](#) on page 378.

NOTE

“->” indicates the commands that you send to the U2000 Series power sensor.

“<-” indicates the response from the U2000 Series power sensor.

Example 7: Time-Gated Burst Power Measurement via Internal/External Trigger

The SCPI programming sequence examples for time-gated burst power measurement via internal/external trigger are shown as follows.

Internal Trigger

```

-> SYST:PRES DEF // Presets the U2000 Series
                    power sensor.
-> FREQ 1000MHz // Sets the frequency (for
                  example: 1000 MHz).
-> DET:FUNC NORM // Sets the measurement mode
                  to NORMa1.
-> TRIG:SOUR INT // Sets the trigger source to
                  INTernal.
-> TRIG:LEV -6 // Specifies the power level at
                which a trigger event is
                recognized (for example:
                -6 dBm).
-> SWE:TIME 0.01 // Specifies the gate duration (for
                  example: 10 ms).
-> SWE:OFFS:TIME 0.0002 // Specifies the offset duration
                          (for example: 200 μs).
-> CALC:FEED "POW:AVER ON SWEEP" // Performs time-gated burst
                                   power measurement.
-> FETC? // Retrieves the power
          measurement results.

```

External Trigger

```

-> SYST:PRES DEF // Presets the U2000 Series
                    power sensor.
-> FREQ 1000MHz // Sets the frequency (for
                 example: 1000 MHz).
-> DET:FUNC NORM // Sets the measurement mode
                 to NORMa1.
-> TRIG:SOUR EXT // Sets the trigger source to
                 EXTernal.
-> SWE:TIME 0.01 // Specifies the gate duration (for
                 example: 10 ms).
-> SWE:OFFS:TIME 0.0001 // Specifies the offset duration
                        (for example: 100  $\mu$ s).
-> CALC:FEED "POW:AVER ON SWEEP" // Performs time-gated burst
                                   power measurement.
-> FETC? // Retrieves the power
          measurement results.

```

For more information on how to query the measurement results, refer to [“Example 3: FETCh, MEASure, and READ Queries”](#) on page 378.

NOTE

“->” indicates the commands that you send to the U2000 Series power sensor.

“<-” indicates the response from the U2000 Series power sensor.

Example 8: Creating a Trace Graph Display

The SCPI programming sequence example for configuring measurement results in trace format is shown as follows.

```

-> DET:FUNC NORM // Sets the measurement mode
                    to NORMAL.
-> TRIG:SOUR INT|EXT // Sets the trigger source to
                    INTERNAL or EXTERNAL.
-> TRAC:STAT ON // Enables trace capture.
-> TRAC:TIME 0.001 // Specifies the trace duration
                    (for example: 1 ms).
-> TRAC:OFFS:TIME 0.0002 // Specifies the offset duration
                    (for example: 200 μs).
-> TRAC:DATA? MRES // Retrieves the trace data.
                    A trace display format must be
                    set (for example: medium
                    resolution).

```

NOTE

“->” indicates the commands that you send to the U2000 Series power sensor.

“<-” indicates the response from the U2000 Series power sensor.

Example 9: Acquiring 100, 250, and 1000 Readings/s

To acquire 100 readings/s, set the trigger mode to Free Run and it will continuously take measurements. Set the measurement rate to **FAST** mode and trigger count to **1** before querying the measurement.

NOTE

Non-Burst Mode (Free Run): >100 readings/s.

```

-> INIT:CONT ON           // Sets to Free Run mode.
-> MRAT FAST             // Sets the measurement rate to
                        // FAST.
-> TRIG:COUN 1          // Automatically set to 1 in
                        // NORMa1 or DOUB1e mode.
-> FETC?                // Retrieves the data in Free Run
                        // mode.

```

To acquire 250 readings/s, set the trigger mode to Single Trigger. Set the measurement rate to **FAST** mode and trigger count to **50** before querying the measurement.

NOTE

Burst Mode (Single Trigger): >250 readings/s.

```

-> INIT:CONT OFF        // Sets to Single Trigger mode.
-> MRAT FAST           // Sets measurement rate to
                        // FAST mode.
-> TRIG:COUN 50        // To attain 250 readings/s.
-> READ?              // Retrieves the data in Single
                        // Trigger mode.

```

NOTE

“->” indicates the commands that you send to the U2000 Series power sensor.

“<-” indicates the response from the U2000 Series power sensor.

To acquire 1000 readings/s, set the trigger mode to Free Run and it will continuously take the measurements. Set the measurement rate to **FAST** mode and trigger count to **50** (buffered mode) before querying the measurement results. The 1000 readings/s is the derived measurement speed from the first 50 readings in buffered mode.

NOTE

Burst Mode (Free Run): >1000 readings/s.

```

-> INIT:CONT ON // Sets to Free Run mode.
-> MRAT FAST // Sets measurement rate to
FAST mode.
-> TRIG:COUN 50 // To attain 1000 readings/s.
-> FETC? // Retrieves the data in Free Run
mode.
```

Example 10: Frequency-Dependent Offset

The frequency-dependent offset (FDO) feature provides you a convenient way to store the offset values in a handy table. It is useful in applications involving sequential frequency sweep and connected with multiple external test setups. 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 preentered offset values will be retrieved to compensate your external test setup over a range of frequencies.

The following SCPI 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 a FDO table named
                                     // "CUSTOM_A".
-> MEM:TABL:FREQ 50MHz                // Enters the frequency into the
                                     // selected table (for example:
                                     // 50 MHz).
-> MEM:TABL:GAIN 50                    // Enters the reference offset
                                     // factor into the selected table
                                     // (for example: 50%).
-> CORR:CSET2 "CUSTOM_A"              // Enters the name of the
                                     // selected table.
-> CORR:CSET2:STAT 1                  // Enables the selected table.
-> FREQ 1000MHz                       // Sets the frequency (for
                                     // example: 1000 MHz).
-> FETC?                              // Queries the measurement
                                     // results.
```

NOTE

“->” indicates the commands that you send to the U2000 Series power sensor.

“<-” indicates the response from the U2000 Series power sensor.

Example 11: 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 U2000 Series power sensor 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 U2000 Series power sensor, the U2000 Series power sensor 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 U2000 Series power sensor will take up to 128 readings and perform averaging, to acquire one reading for 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 SCPI programming sequence for the Power Sweep usage is shown below.

```
-> TRIG:SOUR EXT           // Sets to external trigger source, which is
                           // required for the Power Sweep operation.
-> TRIG:SLOP POS           // Sets the U2000 Series power sensor to
                           // accept an external positive-edge trigger.
-> AVER:COUN 64            // Sets the filter length to 64.
-> *OPC                     // Enables the OPC feature.
-> *ESR?                    // The *ESR? is issued for the first time.
<- 129                     // Some non-zero value (any value ranging from
                           // 0 to 255) will be returned when the *ESR? is
                           // issued for the first time.

-> *ESR?                    // The *ESR? is issued for the second time.
<- 0                       // The returned value will be cleared to 0 when
                           // the *ESR? is issued for the second time.

-> BUFF:COUN 2             // Sets the Power Sweep mode to capture two
                           // triggers.

-> INIT:CONT ON            // Sets the U2000 Series power sensor to
                           // accept continuous trigger cycles.
```

Sends a positive-edged trigger to the U2000 Series power sensor 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 U2000 Series power sensor 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 U2000 Series power sensor.

“<-” indicates the response from the U2000 Series power sensor.

Example 12: 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 frequency, inclusive of the start and stop frequency.

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 unit, the U2000 Series power sensor 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 U2000 Series power sensor will take 128 readings and perform averaging to acquire one reading for 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 SCPI programming sequence for the Frequency Sweep operation is shown below.

```

-> TRIG:SOUR EXT           // Sets to external trigger source, which is
                           // required for the Frequency Sweep operation.
-> TRIG:SLOP POS          // Sets the U2000 Series power sensor to
                           // accept an external positive-edge trigger.
-> AVER:COUN 64           // Sets the filter length to 64.
-> *OPC                   // Enables the OPC feature.
-> *ESR?                  // The *ESR? is issued for the first time.
<- 129                   // Some non-zero value (any value ranging from
                           // 0 to 255) will be returned when the *ESR? is
                           // issued for the first time.

-> *ESR?                  // The *ESR? is issued for the second time.
<- 0                     // The returned value will be cleared to 0 when
                           // the *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 U2000 Series power sensor to
                           // accept continuous trigger cycles.

```

Sends a positive-edged trigger to the U2000 Series power sensor 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 U2000 Series power sensor 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.
```


NOTE

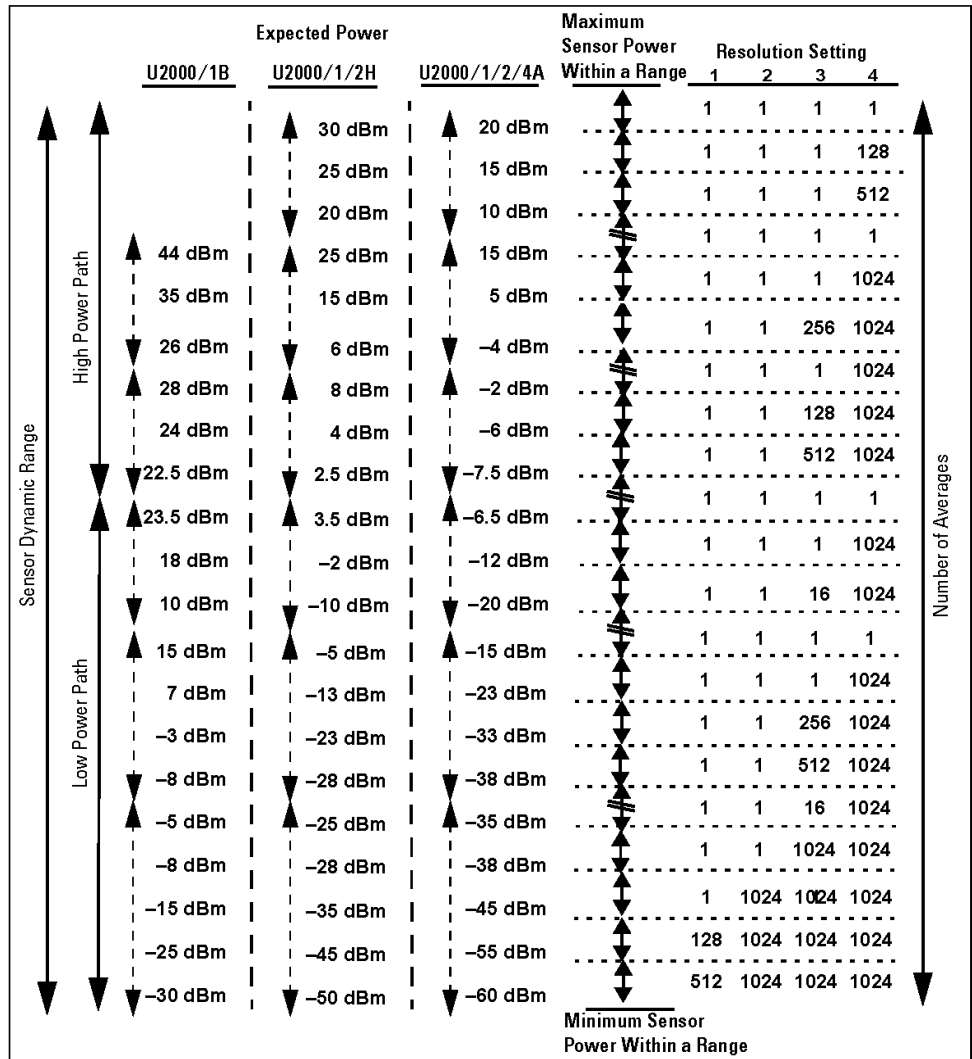
- “->” indicates the commands that you send to the U2000 Series power sensor.
 - “<-” indicates the response from the U2000 Series power sensor.
 - To switch to Power Sweep, the SENS:FREQ:STEP command has to be set to 0. The SENS:BUFF:COUN command will only take effect if the SENS:FREQ:STEP is set to 0.
-

A Appendix A

Auto-Averaging Settings [A-398](#)

Auto-Averaging Settings

In the following figure, the dotted-line arrow  indicates the internal range based on the internal circuitry of the power sensor. The ranges will be automatically selected in correspondence with the power level to best fit the operating conditions and settings.





This information is subject to change without notice. Always refer to the Keysight website for the latest revision.

© Keysight Technologies 2007–2016
Edition 7, April 15, 2016

Printed in Malaysia



U2000-90411

www.keysight.com