



## Programming Manual

7027, 7029, 7037, and 7039 Pulse Sensors

This is a preliminary manual. Specifications, limits, and text are subject to change without notice. The information within this manual was as complete as possible at the time of printing.

Bird Electronic Corporation is not liable for errors. Check the Web Site for updates:

<http://www.bird-technologies.com/>

Copyright 2021 by Bird Technologies, Inc.

Manual Part Number 920-7027-SCPI Rev. P2

## Table of Contents

SCPI Basics.....	1
Pulse Sensor Error Codes .....	9
SCPI Commands .....	10
Calculate .....	10
CALCulate:AVERage:CLEar .....	10
CALCulate:AVERage:COUNT .....	10
CALCulate:AVERage:COUNT? .....	11
CALCulate:AVERage[:STATe].....	11
CALCulate:AVERage[:STATe]? .....	12
CALCulate:GATE:BEgin:DElay.....	12
CALCulate:GATE:BEgin:DElay?.....	13
CALCulate:GATE:BEgin:LEVel:HIGH .....	13
CALCulate:GATE:BEgin:LEVel:HIGH? .....	14
CALCulate:GATE:BEgin:LEVel:LOW.....	14
CALCulate:GATE:BEgin:LEVel:LOW?.....	15
CALCulate:GATE:END:DElay .....	15
CALCulate:GATE:END:DElay? .....	16
CALCulate:GATE:END:LEVel:HIGH.....	16
CALCulate:GATE:END:LEVel:HIGH?.....	17
CALCulate:GATE:END:LEVel:LOW .....	17
CALCulate:GATE:END:LEVel:LOW? .....	18
CALCulate:STATe<n>:ENable.....	18
CALC:STATe<n>:ENable? .....	19

---

CALCulate:STATe<n>:BEGin[:Delay] .....	19
CALCulate:STATe<n>:BEGin[:Delay]? .....	20
CALCulate:STATe<n>:END[:Delay] .....	20
CALCulate:STATe<n>:END[:Delay]? .....	21
Common .....	22
*CLS .....	22
*ESE .....	22
*ESE? .....	23
*ESR? .....	23
*IDN? .....	24
*OPC .....	24
*OPC? .....	25
*RST .....	25
*SRE .....	26
*SRE? .....	26
*STB? .....	27
*TRG .....	27
*TST? .....	28
*WAI .....	28
Format .....	29
FORMat:SREGister? .....	29

Measure .....	30
CONFigure .....	30
FETCh[:SCALar]:TEMPerature? .....	30
FETCh[:SCALar][:POWer]:DCYClE? .....	31
FETCh[:SCALar][:POWer]:FREQuency? .....	31
FETCh[:SCALar][:POWer]:GATE:COUnT? .....	32
FETCh[:SCALar][:POWer]:GATE:MAXimum? .....	32
FETCh[:SCALar][:POWer]:GATE:MINimum? .....	33
FETCh[:SCALar][:POWer]:GATE[:MEAN]? .....	33
FETCh[:SCALar][:POWer]:PERiod? .....	34
FETCh[:SCALar][:POWer]:PRF? .....	34
FETCh[:SCALar][:POWer]:REFLected:AVERage? .....	35
FETCh[:SCALar][:POWer]:WIDTh? .....	35
FETCh[:SCALar][:POWer][:FORWard]:AVERage? .....	36
FETCh[:SCALar][:POWer]:STATe<n>[:MEAN]? .....	36
FETCh[:SCALar][:POWer]:STATe<n>:MAXimum? .....	37
FETCh[:SCALar][:POWer]:STATe<n>:MINimum? .....	37
PnP .....	38
PNP:FILE:BLock:DATA? .....	38
PNP:FILE:BLock:NUMBer .....	38
PNP:FILE:BLock:NUMBer? .....	39
PNP:FILE:SIZE? .....	40
PNP:ITRansfer .....	40
PNP[:AVAIlable]? .....	41

---

Sense.....	42
SENSe:FREQuency.....	42
SENSe:FREQuency:AUTO .....	42
SENSe:FREQuency:RANGe:LOWer? .....	43
SENSe:FREQuency:RANGe:UPPer? .....	44
SENSe:FREQuency? .....	44
SENSe:REFLected:ENABLE .....	45
SENSe:PERiod .....	46
SENSe:PERiod? .....	46
SENSe:PERiod:AUTO .....	47
SENSe:PERiod:AUTO?.....	47
SENSe:SWEEp:DELay.....	48
SENSe:SWEEp:DELay? .....	48
SENSe:SWEEp:TIME .....	49
SENSe:SWEEp:TIME?.....	49
SENSe:SWEEp:TIME:AUTO .....	50
SENSe:SWEEp:TIME:AUTO? .....	50
SENSe:SWEEp:TIME:AUTO:PERIods.....	51
SENSe:SWEEp:TIME:AUTO:PERIods? .....	51
Status .....	52
STATus:MEASurement:CONDition? .....	52

STATus:MEASurement:ENABLE .....	52
STATus:MEASurement:ENABLE? .....	53
STATus:MEASurement:NTRansition .....	53
STATus:MEASurement:NTRansition? .....	54
STATus:MEASurement:PTRansition .....	54
STATus:MEASurement:PTRansition? .....	55
STATus:MEASurement[:EVENT]?.....	55
STATus:OPERation:CONDition? .....	56
STATus:OPERation:ENABLE .....	56
STATus:OPERation:ENABLE?.....	57
STATus:OPERation:NTRansition.....	57
STATus:OPERation:NTRansition? .....	58
STATus:OPERation:PTRansition .....	58
STATus:OPERation:PTRansition? .....	59
STATus:OPERation[:EVENT]?.....	59
STATus:PRESet .....	60
STATus:QUEStionable:CONDition? .....	60
STATus:QUEStionable:ENABLE .....	61
STATus:QUEStionable:ENABLE? .....	61
STATus:QUEStionable:NTRansition.....	62
STATus:QUEStionable:NTRansition?.....	62
STATus:QUEStionable:PTRansition .....	63
STATus:QUEStionable:PTRansition? .....	63
STATus:QUEStionable[:EVENT]?.....	64

---

System.....	65
SYSTem:ERRor:COUNT?.....	65
SYSTem:ERRor[:NEXT]?.....	65
SYSTem:IDENtity:CALDate?.....	66
SYSTem:IDENtity:FWRev? .....	66
SYSTem:IDENtity:HWRev? .....	67
SYSTem:IDENtity:MFGDate?.....	67
SYSTem:IDENtity:MODEl? .....	68
SYSTem:IDENtity:SN? .....	68
SYSTem:PRESet .....	69
SYSTem:TIMestamp? .....	69
SYSTem:VERSion? .....	70
SYSTem:CAPability? .....	70
Trigger .....	71
[TRIGger:]ABORt .....	71
[TRIGger:]INITiate:CONTInuous .....	71
[TRIGger:]INITiate:CONTInuous? .....	72
[TRIGger:]INITiate[:IMMEDIATE].....	72
TRIGer:SOURce.....	73
TRIGer:SOURce?.....	73
TRIGer:SLOPe .....	74

TRIGger:SLOPe? .....	74
TRIGger:LEVel.....	75
TRIGger:LEVel? .....	75
TRIGger:LEVel:AUTO .....	76
TRIGger:LEVel:AUTO? .....	76
TRIGger:DElay.....	77
TRIGger:DElay?.....	77
Trace .....	78
TRACe:TIME[:DATA]? .....	78
TRACe:TIME:POINTS:MAX.....	78
TRACe:TIME:POINTS:MAX?.....	79
Register Bit Definitions .....	80
Glossary .....	84



This chapter provides an overview of the Standard Commands for Programmable Instruments (SCPI) language.

## **SCPI Basics**

---

This section describes the general use of the SCPI language for the Bird Pulse Sensors. It is not intended to teach you everything about the SCPI language; the SCPI Consortium or IEEE can provide that level of detailed information.

For a list of the specific commands available for the Bird Pulse Sensors, refer to the table of contents.

### **Common Terms**

The following terms are used throughout the remainder of this section:

#### **Command**

A command is an instruction in SCPI consisting of mnemonics (keywords), parameters (arguments), and punctuation. Commands are used to control the instrument.

#### **Controller**

A controller is any device used to control the Bird Pulse Sensors, for example a computer or another instrument.

#### **Program Message**

A program message is a combination of one or more properly formatted commands. Program messages are sent by the controller to the Bird Pulse Sensors.

#### **Query**

A query is a special type of command used to instruct the Bird Pulse Sensors to make response data available to the controller. A query ends with a question mark. Generally you can query any command value that you set.

#### **Response**

A response message is a collection of data in specific SCPI formats sent from the Bird Pulse Sensor to the controller. Response messages tell the controller about the internal state of the Bird Pulse Sensors.

## Command Syntax

A command is made up of one or more keywords. Consecutive keywords are separated with colons (:). The keywords are followed by a parameter.

### Optional Keywords and Parameters

Characters	Meaning	Example
[ ]	Square brackets indicate that the enclosed keywords or parameters are optional when composing the command. These implied keywords or parameters will be executed even if they are omitted.	<p>FETCH[:SCALar][:POWer][:FORWard]:AVERage?</p> <p>SCALar, POWer, and FORWard are optional items.</p>

### Command Syntax

Characters, Keywords, and Syntax	Example
<p>Upper- case lettering indicates the minimum set of characters required to execute the command. But, each mode of the command must be in either short form or the complete long form (no in between). Example:</p> <p>Correct: :FREQ :FREQUENCY</p> <p>Incorrect: :FREQuenc</p>	<p>SENSe:FREQUENCY?, SENS:FREQ? is the minimum requirement.</p>

<p>Lower- case lettering indicates the portion of the command that is optional; it can either be included with the upper- case portion of the command or omitted. This is the flexible format principle called forgiving listening. Refer to <a href="#">Command Parameters and Responses</a> for more information.</p>	<p>:FREQuency</p> <p>Either :FREQ, :FREQuency, or :FREQUENCY is correct.</p>
<p>A colon must be placed between two command mnemonics.</p>	<p>STATus:OPERation:ENABLE</p>
<p>White space characters, such as &lt;tab&gt; and &lt;space&gt;, are generally ignored as long as they do not occur within or between keywords.</p> <p>However, you <u>must</u> use white space to separate the command from the parameter.</p>	<p>:FREQ uency or SENSE :REFlected :ENABLE are not allowed.</p> <p>A &lt;space&gt; between :ENABLE and 1 is mandatory.</p> <p>SENSe:REFlected:ENABLE 1</p>

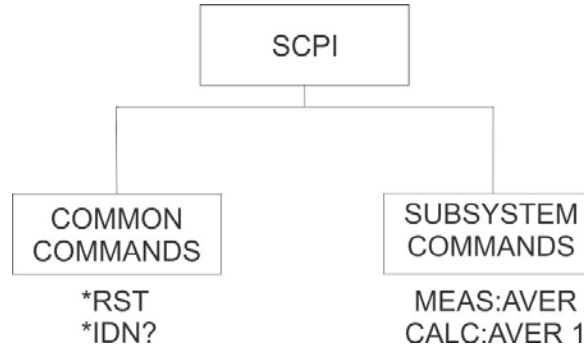
## Command Types

Commands can be separated into two groups: common commands and subsystem commands. See [Command Types](#).

Common commands are used to manage status registers, synchronization, and data storage and are defined by IEEE 488.2. They are easy to recognize because they all begin with an asterisk. For example \*IDN?, \*OPC, and \*RST are common commands. Common commands are not part of any subsystem and the Bird Pulse Sensors interprets them in the same way, regardless of the current path setting.

Subsystem commands are distinguished by the colon (:). The colon is used between keywords, as in SENSE:FREQUENCY:RANGE:UPPER?

### Command Types



---

## Command Parameters and Responses

SCPI allows different data formats to be used in program and response messages. It does this to accommodate the principle of forgiving listening and precise talking.

Forgiving listening means the command and parameter formats are flexible.

Precise talking means that the response format for a particular query is always the same.

When programming a parameter, you may generally use a flexible format type (e.g. numeric\_value>). Response data (data returned to the controller) is more concise and restricted, so that you receive a specific format type (e.g. NR2>) every time.

For example, for the CALCulate:AVERage:COUNT command, the Bird Pulse Sensors accepts CALCulate:AVERage:COUNT 200, or CALC:AVER:COUN 2.00E2 to set the average count parameter to 200. The CALCulate:AVERage:COUNT? query always returns an integer (NR1) in the response message.

The IEEE 488.2 data types employed in the program and response messages are defined in the glossary. For more information on program or response data types refer to IEEE 488.2.

**Numeric Parameters**

Numeric parameters (numeric\_value) are used in both common and subsystem commands. They accept all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation.

The following are examples of numeric parameters:

100	no decimal point required
100.	fractional digits optional
-1.23	leading signs allowed
4.56E<space>3	space allowed after the E in exponential
-7.89E-001	use either E or e in exponential
+256	leading + allowed
.5	digits left of decimal point optional

Numeric parameters greater than 32 bits in length are not supported. If the parameter is greater than 32 bits in length, only the least significant 32 bits are used by the sensor.

Sending "CALC:GATE:END:LEV:LOW #H64" doesn't error because #H64 = 100, which is within the numeric limits of the command, but sending "CALC:GATE:END:LEV:LOW #H65" does error because #H65 = 101, which is outside of the numeric limits of the command.

Sending "CALC:GATE:END:LEV:LOW #H100000064" doesn't error because the least significant 32 bits (#H00000064 = 100) produce a number that is within the numeric limits, despite the fact that the parameter that was sent was significantly outside of the numeric limits for the command.

This holds true for all commands that accept a numeric parameter.

## Boolean Parameters

Boolean parameters represent a single binary condition that is either true or false. The two state Boolean parameter has four arguments. The following list shows the arguments for the two-state Boolean parameter:

ON	Boolean true, upper/lower case allowed
OFF	Boolean false, upper/lower case allowed
1	Boolean true
0	Boolean false

## Real Response Data

Real response data (NR2 or NR3) represent decimal numbers in either fixed decimal or scientific notation. Most high-level programming languages handle either decimal or scientific notation transparently.

The following are examples of real response data:

+4.000000E+010, -9.990000E+002

-9.990000E+002

+4.000000000000000E+010

+1

0

## Integer Response Data

Integer response data (NR1) are decimal representations of integer values including optional signs. Most status register related queries return integer response data.

The following are examples of integer response data:

0	signs are optional
+100	leading + allowed
-100	leading - allowed
256	never any decimal point

## String Response Data

String response data are similar to string parameters. The main difference is that string response data returns double quotes, rather than single quotes. Embedded double quotes may be present in string response data. Embedded quotes appear as two adjacent double quotes with no characters between them.

The following are examples of string response data:

"This is a string"

"one double quote inside brackets: [""]"

"Hello!"

## General process for taking measurements

The general recommended workflow for capturing data using the sensor is as follows:

- Configure Sensor
  - Send an INITiate command.
  - Use fetch commands to retrieve desired measurements.
  - Loop back to initiate for next measurement.



---

## Pulse Sensor Error Codes

---

Error Type	Range
Syntax errors	-100 to -199
Execution errors	-200 to -299
Device failure errors	-300 to -349
Query errors	-400 to -499
Power on errors	-500 to -599
User request errors	-600 to -699
Request control errors	-700 to -799
Operation complete errors	-800 to -899
Custom device-specific error codes	600 to 699

## SCPI Commands

### Calculate

#### **CALCulate:AVERage:CLEar**

**Description:** Clear Average Command

**Parameter Type:**

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Clear the accumulated average and restart the averaging process with the next measurement.

#### **CALCulate:AVERage:COUNt**

**Description:** Average Count Command

**Parameter Type:** <numeric\_value>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets the number of measurements to average.

---

## **CALCulate:AVERAge:COUNT?**

**Description:** Average Count Query

**Parameter Type:** <numeric\_value>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the number of measurements to average.

## **CALCulate:AVERAge[:STATe]**

**Description:** Average Command

**Parameter Type:** <boolean>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:** 1 (ON)

**Parameter Values:** 1 (ON) | 0 (OFF)

**Parameter Units:**

**Notes:** Set the result averaging state.



## **CALCulate:AVERAge[:STATe]?**

**Description:** Average Query

**Parameter Type:** <boolean>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** 1 (ON)

**Parameter Values:** 1 (ON) | 0 (OFF)

**Parameter Units:**

**Notes:** Get the result averaging state.

## **CALCulate:GATE:BEGiN:DELaY**

**Description:** Gate Begin Delay Command

**Parameter Type:** <numeric\_value>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** 0

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** Set the delay from the begin event to the beginning of the gated time interval. Begin Delay is normally > 0 to begin the gated timing interval after the rising edge.

---

## **CALCulate:GATE:BEGiN:DELaY?**

**Description:** Gate Begin Delay Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** Get the delay from the begin event to the beginning of the gated time interval.

## **CALCulate:GATE:BEGiN:LEVEl:HIGH**

**Description:** Gate Begin Level High Command

**Parameter Type:** <numeric\_value>

**Parameter Minimum:** 0                      **Parameter Maximum:** 100

**Parameter Default:** 90

**Parameter Values:**

**Parameter Units:** PCT

**Notes:** Set the high threshold for detection of the rising edge of the pulse.



---

## **CALCulate:GATE:BEGiN:LEVel:HIGH?**

**Description:** Gate Begin Level High Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** PCT

**Notes:** Get the high threshold for detection of the rising edge of the pulse.

## **CALCulate:GATE:BEGiN:LEVel:LOW**

**Description:** Gate Begin Level Low Command

**Parameter Type:** <numeric\_value>

**Parameter Minimum:** 0                      **Parameter Maximum:** 100

**Parameter Default:** 10

**Parameter Values:**

**Parameter Units:** PCT

**Notes:** Set the low threshold for detection of the rising edge of the pulse.

---

## CALCulate:GATE:BEgin:LEVel:LOW?

**Description:** Gate Begin Level Low Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** PCT

**Notes:** Get the low threshold for detection of the rising edge of the pulse.

## CALCulate:GATE:END:DELay

**Description:** Gate End Delay Command

**Parameter Type:** <numeric\_value>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** 0

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** Set the delay from the end event to the end of the gated time interval. End:Delay is normally < 0 to end the gated timing interval before the falling edge.



---

## **CALCulate:GATE:END:DELay?**

**Description:** Gate End Delay Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** Get the delay from the end event to the end of the gated time interval.

## **CALCulate:GATE:END:LEVel:HIGH**

**Description:** Gate End Level High Command

**Parameter Type:** <numeric\_value>

**Parameter Minimum:** 0                      **Parameter Maximum:** 100

**Parameter Default:** 90

**Parameter Values:**

**Parameter Units:** PCT

**Notes:** Set the high threshold for detection of the falling edge of the pulse.



---

## CALCulate:GATE:END:LEVel:HIGH?

**Description:** Gate End Level High Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** PCT

**Notes:** Get the high threshold for detection of the falling edge of the pulse.

## CALCulate:GATE:END:LEVel:LOW

**Description:** Gate End Level Low Command

**Parameter Type:** <numeric\_value>

**Parameter Minimum:** 0                      **Parameter Maximum:** 100

**Parameter Default:** 10

**Parameter Values:**

**Parameter Units:** PCT

**Notes:** Set the low threshold for detection of the falling edge of the pulse.



---

## **CALCulate:GATE:END:LEVel:LOW?**

**Description:** Gate End Level Low Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** PCT

**Notes:** Get the low threshold for detection of the falling edge of the pulse.

## **CALCulate:STATE<n>:ENABLE**

**Description:** Enable or disable state n.

**Parameter Type:** <boolean>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** OFF

**Parameter Values:** ON | OFF

**Parameter Units:**

**Notes:** <n> is integer between 1 and 4 corresponding to state1 through state4.

---

## **CALC:STATE<n>:ENABLE?**

**Description:** Query of state <n> enable/disable status.

**Parameter Type:** <boolean>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** <n> is an integer between 1 and 4

## **CALCulate:STATE<n>:BEGin[:Delay]**

**Description:** Specifies position of state <n> relative to trigger point

**Parameter Type:** <numeric\_value>

**Parameter Minimum**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** <n> is integer between 1 and 4 corresponding to state1 through state4.



---

## **CALCulate:STATE<n>:BEGin[:Delay]?**

**Description:** Returns Position of state n relative to trigger point.

**Parameter Type:** <NR2>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** <n> is integer between 1 and 4 corresponding to state1 through state4.

## **CALCulate:STATE<n>:END[:Delay]**

**Description:** Specifies position of state n relative to trigger point

**Parameter Type:** <numeric\_value>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** <n> is integer between 1 and 4 corresponding to state1 through state4.

---

**CALCulate:STATE<n>:END[:Delay]?**

**Description:** Returns position of state n relative to trigger point.

**Parameter Type:** <NR2>

**Parameter Minimum:**

**Parameter Max:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** <n> is integer between 1 and 4 corresponding to state1 through state4.

---

## Common

### \*CLS

**Description:** Clear Status Command

**Parameter Type:**

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Clears all status structures and forces the device into an idle state. Also clears command and error queues.

### \*ESE

**Description:** Standard Event Status Enable Command

**Parameter Type:** <NRf>

**Parameter Minimum:** 0

**Parameter Maximum:** 255

**Parameter Default:** 255

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets the Standard Event Status Enable Register bits as defined in section 11.5.1.3 of IEEE488.2. This determines which events are enabled.

[See Register definitions](#)

---

## \*ESE?

**Description:** Standard Event Status Enable Query

**Parameter Type:** <NR1>

**Parameter Minimum:** 0

**Parameter Maximum:** 255

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Get the standard event enable mask which indicates which events are enabled.

## \*ESR?

**Description:** Standard Event Status Register Query

**Parameter Type:** <NR1>

**Parameter Minimum:** 0

**Parameter Maximum:** 255

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Get the Standard Event Register which indicates which events have occurred.

[See Register definitions](#)

## **\*IDN?**

**Description:** Identification Query

**Parameter Type:** <arbitrary\_ascii>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:** Format BIRD,{model see Model Identification},{serial number},{firmware revision}

**Parameter Units:**

**Notes:** Get device identification.

## **\*OPC**

**Description:** Operation Complete Command

**Parameter Type:**

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Request a check for all commands to be executed.



---

## **\*OPC?**

**Description:** Operation Complete Query

**Parameter Type:** <boolean>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Places a "1" in the output queue after all commands have been executed.

## **\*RST**

**Description:** Reset Command

**Parameter Type:**

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Performs a software device reset, placing the device into a known state. Does not clear the output queue or status enable register settings.

[See Register definitions](#)

## **\*SRE**

**Description:** Service Request Enable Command

**Parameter Type:** <NRf>

**Parameter Minimum:** 0

**Parameter Maximum:** 255

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Set Service Request Enable Register. Controls which conditions in the Status Byte will generate a service request. Set bits enable a service request on the corresponding condition in the Status Byte per IEEE.IEC 60488.2-2004, Section 11.3. [See Register definitions](#)

## **\*SRE?**

**Description:** Service Request Enable Query

**Parameter Type:** <NR1>

**Parameter Minimum:** 0

**Parameter Maximum:** 255

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Get Service Request Enable Register. Indicates which conditions in the Status Byte will generate a service request.

[See Register definitions](#)

---

## **\*STB?**

**Description:** Read Status Byte Query

**Parameter Type:** <NR1>

**Parameter Minimum:** 0

**Parameter Maximum:** 255

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Get the Status Byte.

## **\*TRG**

**Description:** Trigger Command

**Parameter Type:**

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Not applicable.

## **\*TST?**

**Description:** Self-Test Query

**Parameter Type:** <NR1>

**Parameter Minimum:** -32767

**Parameter Maximum:** 32767 **Parameter**

**Default:**

**Parameter Values:** 0 (No errors) or 695 (FPGA register self-test failed)

**Parameter Units:**

**Notes:** Performs a self-test and places the result of the test in the output queue.

## **\*WAI**

**Description:** Wait-to-Continue Command

**Parameter Type:**

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** For overlapped command synchronization. Can result in no-op if overlapped commands are not used.

---

## Format

### FORMat:SREGister?

**Description:** Format Status Register Query

**Parameter Type:** <character\_data>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** ASCii

**Parameter Values:** ASCii

**Parameter Units:**

**Notes:** Gets the format of the status registers. Status registers are always formatted as ASCII.

[See Register definitions](#)

---

## Measure

### CONFigure

**Description:** Configure Command

**Parameter Type:**

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets up a measurement. Performs default configuration in the active mode.

### FETCh[:SCALAr]:TEMPerature?

**Description:** Fetch Temperature Query

**Parameter Type:** <NR2>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:** Parameter

**Values:**

**Parameter Units:** C

**Notes:** Gets the current temperature.

---

## FETCh[:SCALar][:POWer]:DCYClE?

**Description:** Fetch Pulse Duty Cycle Query

**Parameter Type:** <NR2>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** PCT

**Notes:** Gets the duty cycle for the most recent measurement. The proportion of the average pulse width to the average pulse period is returned. See Statistics. This does not initiate a new measurement.

## FETCh[:SCALar][:POWer]:FREQuency?

**Description:** Fetch Frequency Query

**Parameter Type:** <NR2>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** MHZ

**Notes:** Gets the RF carrier frequency from the most recent measurement. This does not initiate a new measurement.



---

## **FETCh[:SCALAr][:POWer]:GATE:COUNT?**

**Description:** Fetch Gate Count Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the number of pulses for which statistics have been accumulated. This does not initiate a new measurement. See Statistics.

## **FETCh[:SCALAr][:POWer]:GATE:MAXimum?**

**Description:** Fetch Gated Maximum Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** W

**Notes:** Gets the maximum of gated mean. The mean power in the gate interval is determined for each pulse. The maximum of the mean power over all the pulses is returned. See Statistics. This does not initiate a new measurement.



---

**FETCh[:SCALar][:POWer]:GATE:MINimum?**

**Description:** Fetch Gated Minimum Query

**Parameter Type:** <NR2>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** W

**Notes:** Gets the minimum of gated mean. The mean power in the gate interval is determined for each pulse. The minimum of the mean power over all the pulses is returned. See Statistics. This does not initiate a new measurement.

**FETCh[:SCALar][:POWer]:GATE[:MEAN]?**

**Description:** Fetch Gated Mean Query

**Parameter Type:** <NR2>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** W

**Notes:** Gets the mean power in the gate interval for the most recent measurement. The mean power in the gate interval is determined for each pulse. The average of the mean power over all the pulses is returned. See Statistics. This does not initiate a new measurement.



---

## **FETCh[:SCALar][:POWer]:PERiod?**

**Description:** Fetch Pulse Period Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** USEC

**Notes:** Get the pulse period for the most recent measurement. The period is determined for each pulse. The average of the period over all the pulses is returned. See Statistics. This does not initiate a new measurement.

## **FETCh[:SCALar][:POWer]:PRF?**

**Description:** Fetch Pulse Repetition Frequency Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** HZ

**Notes:** Gets the pulse repetition frequency for the most recent measurement. The average pulse repetition frequency is returned. See Statistics. This does not initiate a new measurement.

---

## FETCh[:SCALAr][:POWer]:REFLected:AVERage?

**Description:** Fetch Reflected Average Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** W

**Notes:** Gets the reflected average power from the most recent measurement. This does not initiate a new measurement.

## FETCh[:SCALAr][:POWer]:WIDTh?

**Description:** Fetch Pulse Width Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** USEC

**Notes:** Get the pulse width for the most recent measurement. The width is determined for each pulse. The average of the width over all the pulses is returned. See Statistics. This does not initiate a new measurement.



---

## **FETCh[:SCALar][:POWER][:FORWard]:AVERAge?**

**Description:** Fetch Forward Average Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** Parameter

**Values:**

**Parameter Units:** W

**Notes:** Gets the forward average power from the most recent measurement. This does not initiate a new measurement.

## **FETCh[:SCALar][:POWER]:STATe<n>[:MEAN]?**

**Description:** Fetch mean power in state

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** W

**Notes:** Gets the mean power in state interval for most recent measurement. <n> is integer between 1 and 4 corresponding to state1 through state4.

---

**FETCh[:SCALar][:POWer]:STATe<n>:MAXimum?**

**Description:** Fetch maximum power in state

**Parameter Type:** <NR2>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** W

**Notes:** Gets maximum power in the interval for the most recent measurement. <n> is integer between 1 and 4 corresponding to state1 through state4.

**FETCh[:SCALar][:POWer]:STATe<n>:MINimum?**

**Description:** Fetch minimum power in state

**Parameter Type:** <NR2>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** W

**Notes:** Gets minimum power in the state interval for the most recent measurement. <n> is integer between 1 and 4 corresponding to state1 through state4.

## PnP

### **PNP:FILE:BLOCK:DATA?**

**Description:** PnP File Block Data Query

**Parameter Type:** <block\_data>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the Plug and Play file block data

### **PNP:FILE:BLOCK:NUMBER**

**Description:** PnP File Block Number Command

**Parameter Type:** <NRf>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets the Plug and Play file block number to transfer.

---

## PNP:FILE:BLOCK:NUMBER?

**Description:** PnP File Block Number Query

**Parameter Type:** <NR1>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the Plug and Play file block number to transfer.

## PNP:FILE:BLOCK:TOTAL?

**Description:** PnP File Block Total Query

**Parameter Type:** <NR1>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the total number of blocks for the Plug and Play file transfer.

## **PNP:FILE:SIZE?**

**Description:** PnP File Size Query

**Parameter Type:** <NR1>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the Plug and Play file size.

## **PNP:ITRansfer**

**Description:** PnP Initiate Transfer Command

**Parameter Type:**

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Initiates a read of the Bird Plug and Play file. This reads file header details into memory and updates the file size and total number of blocks fields. The command is issued once before blocks of file data can be transferred from the device.



---

## PNP:VERSion?

**Description:** PnP Version Query

**Parameter Type:** <NR2>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the Bird Plug and Play protocol revision number. Has the format major.minor.  
Example: 1.19

## PNP[:AVAIlable]?

**Description:** PnP Available Query

**Parameter Type:** <boolean>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:** 1

**Parameter Values:**

**Parameter Units:**

**Notes:** Indicates whether the Bird Plug and Play parameters file is available.

## Sense

### SENSe:FREQuency

**Description:** Sense Frequency Command

**Parameter Type:** <NRf>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** MHZ

**Notes:** Sets the frequency used for amplitude correction when SENSe:CORRection:AUTO is disabled.

### SENSe:FREQuency:AUTO

**Description:** Sense Frequency Automatic Command

**Parameter Type:** <boolean>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:** 1

**Parameter Values:**

**Parameter Units:**

**Notes:** Set automatic frequency dependent amplitude correction state.

---

## SENSe:FREQuency:AUTO?

**Description:** Sense Frequency Auto Query

**Parameter Type:** <boolean>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:** 1

**Parameter Values:**

**Parameter Units:**

**Notes:** Get automatic frequency dependent amplitude correction state.

## SENSe:FREQuency:RANGe:LOWer?

**Description:** Sense Frequency Range Lower Query

**Parameter Type:** <NRf>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** MHZ

**Notes:** Gets minimum calibrated frequency.

## **SENSe:FREQuency:RANGe:UPPer?**

**Description:** Sense Frequency Range Upper Query

**Parameter Type:** <NRf>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** MHZ

**Notes:** Gets maximum calibrated frequency.

## **SENSe:FREQuency?**

**Description:** Sense Frequency Query

**Parameter Type:** <NRf>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** MHZ

**Notes:** Gets the frequency used for amplitude correction when SENSe:CORRection:AUTO is disabled. This is not the frequency measured.

---

## **SENSe:REFLected:ENABle**

**Description:** Sense Reflected Enable Command

**Parameter Type:** <boolean>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:** 1

**Parameter Values:**

**Parameter Units:**

**Notes:** Set reflected power measurement state. Reflected power measurement may be disabled for faster forward power measurements.

## **SENSe:REFLected:ENABle?**

**Description:** Sense Reflected Enable Query

**Parameter Type:** <boolean>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Get reflected power measurement state.

## **SENSe:PERiod**

**Description:** Specifies expected maximum pulse period.

**Parameter Type:** <NRf>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** The sensor determines an appropriate sampling window to measure the repetition rate/period of the pulse. Measurements can be performed on pulses with a lesser pulse period than the expected maximum.

## **SENSe:PERiod?**

**Description:** Gets the expected maximum pulse period.

**Parameter Type:** <NRf>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** The sensor determines an appropriate sampling window to measure the repetition rate/period of the pulse. Measurements can be performed on pulses with a lesser pulse period than the expected Maximum.

---

## SENSe:PERiod:AUTO

**Description:** Turns on automatic detection of maximum pulse period.

**Parameter Type:** <boolean>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** ON

**Parameter Values:** ON | OFF

**Parameter Units:**

**Notes:** When enabled the sensor begins with the lowest decimation rate which provides the minimum duration sampling interval. It successively increases the decimation rate until it is able to detect at least two successive repetitions of a state transition pattern.

## SENSe:PERiod:AUTO?

**Description:** Gets status of automatic detection of pulse period.

**Parameter Type:** <boolean>

**Parameter Min:**                      **Parameter Maximum:**

**Parameter Default:** ON

**Parameter Values:** ON | OFF

**Parameter Units:**

**Notes:** When enabled the sensor begins with the lowest decimation rate which provides the maximum duration sampling interval. It successively increases the decimation rate until it is able to detect at least two successive repetitions of a state transition pattern.

### **SENSe:SWEEp:DELay**

**Description:** Set sweep delay.

**Parameter Type:** <numeric\_value>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** 0

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** Specifies beginning of sweep relative to trigger reference point.

### **SENSe:SWEEp:DELay?**

**Description:** Gets sweep delay.

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** 0

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** Returns the delay between trigger reference point and beginning of sweep.



---

## **SENSe:SWEEp:TIME**

**Description:** Sets time interval to be returned by trace.

**Parameter Type:** <numeric\_value>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** Specifies time interval to be returned by TRACe:TIME:DATA?

## **SENSe:SWEEp:TIME?**

**Description:** Gets time interval to be returned by trace.

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** Returns the time interval to be returned by TRACe:TIME:DATA?.

## **SENSe:SWEEp:TIME:AUTO**

**Description:** Enable using measured pulse period as sweep time.

**Parameter Type:** <boolean>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** ON

**Parameter Values:** ON | OFF

**Parameter Units:**

**Notes:** When enabled use measured pulse period as sweep time.

## **SENSe:SWEEp:TIME:AUTO?**

**Description:** Returns whether sweep time autoset.

**Parameter Type:** <boolean>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:** ON | OFF

**Parameter Units:**

**Notes:**

---

## **SENSe:SWEep:TIME:AUTO:PERIods**

**Description:** Set Number of pulse periods to use as sweep time.

**Parameter Type:** <NRf>

**Parameter Minimum:** 1

**Parameter Maximum:** 8

**Parameter Default:** 1

**Parameter Values:**

**Parameter Units:**

**Notes:** Number of pulse periods to use as sweep time.

## **SENSe:SWEep:TIME:AUTO:PERIods?**

**Description:** Query how many pulse periods are used as sweep time.

**Parameter Type:** <NR1>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Number of pulse periods used for sweep time.

## Status

### **STATus:MEASurement:CONDition?**

**Description:** MEASurement Condition Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:** 0

**Parameter Maximum:** 32767

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Returns the contents of the condition register associated with the status structure defined in the command. Reading the condition register does not clear it.

[See Register definitions](#)

### **STATus:MEASurement:ENABLE**

**Description:** MEASurement Enable Command

**Parameter Type:** <NRf> | <non-decimal numeric>

**Parameter Minimum:** 0

**Parameter Maximum:** 65535

**Parameter Default:** 254

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets the enable mask which allows true conditions in the event register to be reported in the summary bit. If a bit is 1 in the enable register and its associated event bit transitions to true, a positive transition will occur in the associated summary bit.

[See Register definitions](#)

---

## STATus:MEASurement:ENABLE?

**Description:** MEASurement Enable Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:** 0

**Parameter Maximum:** 32767

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the Measurement Event Enable Register see STATus:MEASurement:ENABLE, MEASurement Enable Command.

[See Register definitions](#)

## STATus:MEASurement:NTRansition

**Description:** MEASurement Negative Transition Filter Command

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets the negative transition filter. Setting a bit in the negative transition filter shall cause a 1 to 0 transition in the corresponding bit of the associated condition register to cause a 1 to be written in the associated bit of the corresponding event register.

[See Register definitions](#)

## **STATus:MEASurement:NTRansition?**

**Description:** MEASurement Negative Transition Filter Query

**Parameter Type:** <NR1> | <non-decimal numeric> **Parameter**

**Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the negative transition filter. See STATus:MEASurement:NTRansition, MEASurement Negative Transition Filter Command.

## **STATus:MEASurement:PTRansition**

**Description:** MEASurement Positive Transition Filter Command

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets the positive transition filter. Setting a bit in the positive transition filter shall cause a 0 to 1 transition in the corresponding bit of the associated condition register to cause a 1 to be written in the associated bit of the corresponding event register.

[See Register definitions](#)

---

## STATus:MEASurement:PTRansition?

**Description:** MEASurement Positive Transition Filter Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the positive transition filter. See STATus:MEASurement:PTRansition, MEASurement Positive Transition Filter Command.

## STATus:MEASurement[:EVENT]?

**Description:** MEASurement Event Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:** 0                      **Parameter Maximum:** 32767

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the contents of the Measurement Event Register and clears the register.

[See Register definitions](#)

## **STATus:OPERation:CONDition?**

**Description:** OPER Condition Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:** 0

**Parameter Maximum:** 32767

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Returns the contents of the condition register associated with the status structure defined in the command. Reading the condition register does not clear it.

[See Register definitions](#)

## **STATus:OPERation:ENABLE**

**Description:** OPER Enable Command

**Parameter Type:** <NRf> | <non-decimal numeric>

**Parameter Minimum:** 0

**Parameter Maximum:** 65535

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets the enable mask which allows true conditions in the event register to be reported in the summary bit. If a bit is 1 in the enable register and its associated event bit transitions to true, a positive transition will occur in the associated summary bit.

[See Register definitions](#)



---

## STATus:OPERation:ENABle?

**Description:** OPER Enable Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:** 0

**Parameter Maximum:** 32767

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the Operational Event Enable Register see STATus:OPERation:ENABle, OPER Enable Command.

[See Register definitions](#)

## STATus:OPERation:NTRansition

**Description:** OPER Negative Transition Filter Command

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets the negative transition filter. Setting a bit in the negative transition filter shall cause a 1 to 0 transition in the corresponding bit of the associated condition register to cause a 1 to be written in the associated bit of the corresponding event register.

[See Register definitions](#)

## **STATus:OPERation:NTRansition?**

**Description:** OPER Negative Transition Filter Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the negative transition filter. See STATus:OPERation:NTRansition, OPER Negative Transition Filter Command.

## **STATus:OPERation:PTRansition**

**Description:** OPER Positive Transition Filter Command

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets the positive transition filter. Setting a bit in the positive transition filter shall cause a 0 to 1 transition in the corresponding bit of the associated condition register to cause a 1 to be written in the associated bit of the corresponding event register.

[See Register definitions](#)

---

## STATus:OPERation:PTRansition?

**Description:** OPER Positive Transition Filter Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the negative transition filter. See STATus:OPERation:PTRansition, OPER Positive Transition Filter Command.

## STATus:OPERation[:EVENT]?

**Description:** OPER Event Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:** 0                      **Parameter Maximum:** 32767

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the contents of the Operational Event Register and clears the register.

[See Register definitions](#)

## **STATus:PRESet**

**Description:** Status Preset Command

**Parameter Type:** <NR1> | <non-decimal numeric> **Parameter**

**Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Restores status subsystem and error queue to the factory preset settings.

## **STATus:QUEStionable:CONDition?**

**Description:** QUES Condition Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:** 0                      **Parameter Maximum:** 32767

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Returns the contents of the condition register associated with the status structure defined in the command. Reading the condition register does not clear it.

[See Register definitions](#)

---

## STATus:QUEStionable:ENABle

**Description:** QUES Enable Command

**Parameter Type:** <NRf> | <non-decimal numeric>

**Parameter Minimum:** 0

**Parameter Maximum:** 65535

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets the enable mask which allows true conditions in the event register to be reported in the summary bit. If a bit is 1 in the enable register and its associated event bit transitions to true, a positive transition will occur in the associated summary bit.

[See Register definitions](#)

## STATus:QUEStionable:ENABle?

**Description:** QUES Enable Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:** 0

**Parameter Maximum:** 32767

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the Questionable Status Event Enable Register see STATus:QUEStionable:ENABle, QUES Enable Command.

[See Register definitions](#)

## **STATus:QUEStionable:NTRansition**

**Description:** QUES Negative Transition Filter Command

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets the negative transition filter. Setting a bit in the negative transition filter shall cause a 1 to 0 transition in the corresponding bit of the associated condition register to cause a 1 to be written in the associated bit of the corresponding event register.

[See Register definitions](#)

## **STATus:QUEStionable:NTRansition?**

**Description:** QUES Negative Transition Filter Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the negative transition filter. See STATus:QUEStionable:NTRansition, QUES Negative Transition Filter Command.

---

## STATus:QUEStionable:PTRansition

**Description:** QUES Positive Transition Filter Command

**Parameter Type:** <NR1> | <non-decimal numeric> **Parameter**

**Minimum:**                    **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Sets the positive transition filter. Setting a bit in the positive transition filter shall cause a 0 to 1 transition in the corresponding bit of the associated condition register to cause a 1 to be written in the associated bit of the corresponding event register.

[See Register definitions](#)

## STATus:QUEStionable:PTRansition?

**Description:** QUES Positive Transition Filter Query

**Parameter Type:** <NR1> | <non-decimal numeric> **Parameter**

**Minimum:**                    **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the negative transition filter. See STATus:QUEStionable:PTRansition, QUES Positive Transition Filter Command.

## **STATus:QUEStionable[:EVENT]?**

**Description:** QUES Event Query

**Parameter Type:** <NR1> | <non-decimal numeric>

**Parameter Minimum:** 0

**Parameter Maximum:** 32767

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the contents of the Questionable Status Event Register and clears the register.

[See Register definitions](#)



---

## System

### **SYSTEM:ERRor:COUNT?**

**Description:** System Error Count Query

**Parameter Type:** <NR1>

**Parameter Minimum:** 0

**Parameter Maximum:** 32767

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Get the number of errors in the error queue.

### **SYSTEM:ERRor[:NEXT]?**

**Description:** System Error Message Query

**Parameter Type:** <error\_resp>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Get the next (oldest) entry in the system error queue.

## **SYSTEM:IDENTITY:CALDate?**

**Description:** System Identity Calibration Date Query

**Parameter Type:** <NR1>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Get the Calibration Date. For example January 18th 2013: 20130118.

## **SYSTEM:IDENTITY:FWRev?**

**Description:** System Identity Firmware Revision Query

**Parameter Type:** <arbitrary\_ascii>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the MCU Firmware Revision.

---

## SYSTEM:IDENTity:HWRev?

**Description:** System Identity Hardware Revision Query

**Parameter Type:** <arbitrary\_ascii>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the FPGA Configuration Revision.

## SYSTEM:IDENTity:MFGDate?

**Description:** System Identity Manufacture Date Query

**Parameter Type:** <NR1>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Get the Manufacture Date. For example January 18th 2013: 20130118.

## **SYSTEM:IDENTITY:MODEL?**

**Description:** System Identity Model Query

**Parameter Type:** <arbitrary\_ascii>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Get the model number as given in the \*IDN? response.

## **SYSTEM:IDENTITY:SN?**

**Description:** System Identity Serial Number Query

**Parameter Type:** <arbitrary\_ascii>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Get the serial number as given in the \*IDN? response.

---

## SYSTEM:PRESet

**Description:** System Preset Command

**Parameter Type:**

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Restore factory presets, but do not clear status & error messages.

## SYSTEM:TIMEstamp?

**Description:** System Timestamp Query

**Parameter Type:** <NR1>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Gets the current timestamp in milliseconds since the sensor was started.

## **SYSTem:VERsion?**

**Description:** System Version Query

**Parameter Type:** <NR2>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Get the SCPI version.

## **SYSTem:CAPability?**

**Description:** Get device Capability.

**Parameter Type:** <arbitrary\_ascii>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Arbitrary 7-bit ASCII Response data giving instrument capability. See SCPI-99 section 4, chapter 1.4 "Instrument Classification" for details. For example: (PSENSOR | DIGITIZER | SPSENSOR)

---

## Trigger

### [TRIGger:]ABORt

**Description:** Trigger Abort Command

**Parameter Type:**

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Places trigger into IDLE state. Removes the device from the "wait for trigger" state and places it in the "idle" state. It does not affect any other settings of the trigger system.

### [TRIGger:]INITiate:CONTInuous

**Description:** Trigger Initiate Continuous Command

**Parameter Type:** <boolean>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** 0

**Parameter Values:**

**Parameter Units:**

**Notes:** Set continuous trigger mode.



## **[TRIGger:]INITiate:CONTInuous?**

**Description:** Trigger Initiate Continuous Query

**Parameter Type:** <boolean>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** 0

**Parameter Values:**

**Parameter Units:**

**Notes:** Get continuous trigger mode.

## **[TRIGger:]INITiate[:IMMediate]**

**Description:** Trigger Initiate Immediate Command

**Parameter Type:**

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** Causes trigger to exit the IDLE state. Removes the device from the "wait for trigger" state and places it in the "idle" state. It does not affect any other settings of the trigger system.



---

## TRIGer:SOURce

**Description:** Selects the trigger source

**Parameter Type:** <character\_data>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** INTernal

**Parameter Values:** EXTernal | INTernal

**Parameter Units:**

**Notes:**

## TRIGer:SOURce?

**Description:** Gets the trigger source

**Parameter Type:** <character\_data>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** INTernal

**Parameter Values:** EXTernal | INTernal

**Parameter Units:**

**Notes:**



## **TRIGer:SLOPe**

**Description:** Specifies rising or falling edge trigger

**Parameter Type:** <charater\_data>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** POSitive

**Parameter Values:** POSitive } NEGative | EITHer

**Parameter Units:**

**Notes:**

## **TRIGer:SLOPe?**

**Description:** Returns trigger edge

**Parameter Type:** <character\_data>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:** POSitive

**Parameter Values:** POSitive | NEGative | EITHer

**Parameter Units:**

**Notes**

---

## TRIGger:LEVel

**Description:** Specifies trigger level

**Parameter Type:** <numeric\_value>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:**

## TRIGger:LEVel?

**Description:** Gets trigger level

**Parameter Type:** <NR2>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:**



## **TRIGger:LEVel:AUTO**

**Description:** Enable use of 50% of waveform as trigger level.

**Parameter Type:** <boolean>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:** ON

**Parameter Values:** ON | OFF

**Parameter Units:**

**Notes:**

## **TRIGger:LEVel:AUTO?**

**Description:** Returns weather trigger level auto set.

**Parameter Type:** <boolean>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Default:** ON

**Parameter Values:** ON | OFF

**Parameter Units**

**Notes:**

---

## TRIGger:DELay

**Description:** Sets the trigger delay

**Parameter Type:** <numeric\_value>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** Time between the occurrence of the trigger and the trigger reference point.

## TRIGger:DELay?

**Description:** Gets the trigger delay.

**Parameter Type:** <NR2>

**Parameter Minimum:**

**Parameter Maximum:**

**Parameter Values:**

**Parameter Units:** SEC

**Notes:** Time between the occurrence of the trigger and the trigger reference point.

## Trace

### TRACe:TIME[:DATA]?

**Description:** Get the time domain mode dataset.

**Parameter Type:** <block\_data>

**Parameter Minimum:**                      **Parameter Maximum:**

**Parameter Default:**

**Parameter Values:**

**Parameter Units:**

**Notes:** IEEE488.2 Definite Length Arbitrary Block Response Data having the format: #nsss...sddd...d<LF>. The block data contains a Time Domain mode data set binary structure. See the Structures worksheet for details. Data is encoded as big endian where the most significant byte is transmitted first. Each block is terminated by a line feed character (0x0A hex).

### TRACe:TIME:POINTS:MAX

**Description:** Set the maximum number of points to return in the time domain dataset.

**Parameter Type:** <NRf>

**Parameter Minimum:** 1                      **Parameter Maximum:** 65536

**Parameter Default:** 1000

**Parameter Values:**

**Parameter Units:**

**Notes:**

---

## TRACe:TIME:POINTS:MAX?

**Description:** Get the maximum number of points to return in the time domain dataset.

**Parameter Type:** <NR1>

**Parameter Minimum:** 1

**Parameter Maximum:** 65536

**Parameter Default:** 1000

**Parameter Values:**

**Parameter Units:**

**Notes:**

## Register Bit Definitions

Status Byte Register (IEEE 488.1)	
Bit	Description
0	(Device-specific) A bit is set in the Measurement Available condition register.
1	(Device-specific) A bit is set in the Measurement Available event register.
2	Error queue is not empty.
3	A bit is set in the Questionable status register.
4	The output queue is not empty. A response message is available.
5	A bit is set in the Standard Event status register.
6	Master Status Summary (MSS).
6	Service request from host controller.
7	A bit is set in the Operational status register.

Standard Event Status Register (IEEE 488.1)	
Bit	Description
0	Operation complete. All commands have completed including the previous commands, the *OPC command, and the overlapping commands.
1	Request Control (not used)
2	Query error. Output buffer was empty, or previous query was not read before new command was received, or input and output.
3	Device error. Device-specific error exists including self test or calibration.
4	Execution error exists.
5	Command syntax error exists.
6	Request Control (not used)
7	Power has been turned off and on. Event register is cleared.

Register Bit Definitions



---

Status Measurement Available (STAT:MEAS:?)	
Bit	Description
0	Frequency is available.
1	Temperature is available.
2	Average forward power is available.
3	Average reflected power is available.
4	Pulse parameters are available.
5	Reserved (Always 0)
6	Reserved (Always 0)
7	Reserved (Always 0)
8	Reserved (Always 0)
9	Reserved (Always 0)
10	Reserved (Always 0)
11	Reserved (Always 0)
12	Reserved (Always 0)
13	Reserved (Always 0)
14	Reserved (Always 0)
15	Reserved (Always 0)



Operation Status Register (STAT:OPER:?)	
Bit	Description
0	The instrument is performing a calibration.
1	Waiting for the sensing hardware to stabilize.
2	Reserved
3	Performing a measurement sweep.
4	The instrument is actively measuring.
5	Waiting for a trigger event.
6	Waiting for the trigger to become armed.
7	Performing a runtime measurement correction.
8	Loading an image into the FPGA.
9	Supercap charging.
10	Reserved
11	Reserved
12	Media I/O operation in progress.
13	Reserved
14	Reserved
15	Reserved

---

Questionable Status Register (STAT:QUES:?)	
Bit	Description
0	Reserved
1	Reserved
2	Reserved
3	Power measurement error.
4	Temperature is out of operating range.
5	Not enough power to measure frequency.
6	Reserved
7	Reserved
8	Measurement calibration error, calibration memory loss, or calibration is unsecured.
9	Insufficient data to measure pulse parameters
10	Begin Gate and End Gate overlap
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved

## Glossary

### A

**arbitrary\_ascii:** Arbitrary 7-bit ASCII response data. Not enclosed in quotes.

### B

**block\_data:** Definite length block of binary data. Has the form #nsss...sddd...d<LF>, where '#' indicates a block data type, n gives the number of characters that follow, s is the size of the block in bytes (the number of binary data bytes), d represents a binary data byte, and <LF> is a line feed character that signifies the end of the block. Example: A block of 6 bytes: #40006<b0><b1><b2><b3><b4><b5><LF>

**boolean:** Represents ON|OFF|<NRf>. Has a value of 0 or 1 and is unitless. NRf is rounded to an integer and a non-zero result is interpreted as 1. Queries always return 1 or 0, never ON or OFF.

### C

**character\_data:** Represents 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. Character data represents a discrete parameter and not an arbitrary string of characters.

### E

**error\_resp:** Error response structure having the form: code,"Description;timestamp", where code is the signed 16-bit error code followed by a comma and then a description string enclosed in double quotes. The description string can be an empty string (""). The timestamp is given as an unsigned integer value representing the number of milliseconds since power-on/reset. Example: -102,"Invalid syntax was found in the command string.;50123" Error code values range from -32768 to 32767.

### N

**NAN:** Not a number (NAN) is represented as 9.91E37.

**non-decimal numeric:** Represents numeric information in bases other than 10 (i.e. hexadecimal). Examples of non-decimal numeric include #HA2F and #ha4e.

**NR1:** Signed integer numeric response data. Examples: 123, +456, -789

**NR2:** Floating point numeric response data without exponent. Examples: 1.23, +4.56, -0.789

**NR3:** Floating point numeric response data with exponent. Examples: +1.234E+03, +4.567E-06, -

78.900E+12

**NRf:** Denotes a flexible numeric representation. Examples: +200, -56, +9.9E36 **numeric\_list:** Comma separated list of <NR3> elements. All elements in the list have the same data format. For example: +1.234E+03, +4.567E-06, -78.900E+12

**numeric\_value:** Decimal numeric element. Also includes character program data constants defined in SCPI. Includes the following: <NRf>, <NR1>, <NR2>, <NR3>, MINimum, MAXimum, DEFault, UP, DOWN, NAN, INFinity, NINFinity.

## S

**string:** ASCII character string. Program data encloses a string in single or double quotes. Example "hello" or 'hello'. Response data always encloses a string in double quotes. Example: "A RESPONSE STRING".