Keysight U2500A Series USB Simultaneous Sampling Multifunction Data Acquisition

Programmer's Reference

KEYSIGHT TECHNOLOGIES

Notices

Copyright Notice

© Keysight Technologies 2010 - 2017

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

U2541-90014

Edition

Edition 5, June 1, 2017

Printed in:

Printed in Malaysia

Published by:

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

Technology Licenses

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

Declaration of Conformity

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

U.S. Government Rights

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

Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED "AS IS," AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, KEYSIGHT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. KEYSIGHT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFORMANCE OF THIS DOCUMENT OR OF ANY INFORMATION CONTAINED HEREIN. SHOULD KEYSIGHT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WARRANTY TERMS COVERING THE MATERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT SHALL CONTROL.

Safety Information

CAUTION

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

WARNING

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

Table of Contents

1 Introduction to SCPI

Introduction to the SCPI Language 14

SCPI Conventions and Data Formats 15

Command separators 16

Syntax conventions 19

Syntax diagram conventions 20

Data types and formats 21

Input message terminators 24

Using device clear 24

SCPI Status System 25

Standard Event register 27

Status Byte register 28

2 ACQuire Subsystem

ACQuire:SRATe 30

ACQuire:POINts 32

3 APPLy Subsystem

APPLy? 36

APPLy:SINusoid 37

APPLy:SQUare 39

APPLy:SAWTooth 41

APPLy:TRIangle 43

APPLy:NOISe 45

APPLy:USER 47

4 CALibration Subsystem

CALibration:BEGin 50

5 CONFigure Subsystem

CONFigure:DIGital:DIRection 52

CONFigure:TIMEbase:SOURce 54
CONFigure:TIMEbase:ECLocK 56
CONFigure:SSI 58

6 MEASure Subsystem

MEASure[:VOLTage][:DC]? 62
MEASure:COUNter:DATA? 64
MEASure:COUNter:FREQuency? 66
MEASure:COUNter:PERiod? 68
MEASure:COUNter:PWIDth? 70
MEASure:COUNter:TOTalize? 72
MEASure:DIGital? 74
MEASure:DIGital:BIT? 76

7 OUTPut Subsystem

OUTPut 80
OUTPut:WAVeform:ITERate 82
OUTPut:WAVeform:SRATe 84
OUTPut:WAVeform:FREQuency 87
OUTPut:TRIGger:SOURce 89
OUTPut:TRIGger:TYPe 91
OUTPut:TRIGger:DCouNT 93
OUTPut:TRIGger:ATRiGger:SOURce 95
OUTPut:TRIGger:ATRiGger:CONDition 97
OUTPut:TRIGger:ATRiGger:HTHReshold 100
OUTPut:TRIGger:ATRiGger:LTHReshold 102
OUTPut:TRIGger:DTRiGger:POLarity 104

8 ROUTe Subsystem

ROUTe:CHANnel:RANGe 108
ROUTe:CHANnel:POLarity 110
ROUTe:CHANnel:STYPe? 112

ROUTe:CHANnel:RSouRCe 113
ROUTe:CHANnel:RVOLtage 115
ROUTe:ENABle 117

9 SENSe Subsystem

[SENSe:]VOLTage:RANGe 120
[SENSe:]VOLTage:POLarity 122
[SENSe:]VOLTage:STYPe? 124
[SENSe:]VOLTage:AVERage 125
[SENSe:]COUNter:FUNCtion 127
[SENSe:]COUNter:ABORt 129
[SENSe:]COUNter:GATE:SOURce 131
[SENSe:]COUNter:GATE:POLarity 133
[SENSe:]COUNter:GATE:CONTrol 135
[SENSe:]COUNter:CLocK:SOURce 137
[SENSe:]COUNter:CLocK:INTernal? 139
[SENSe:]COUNter:CLocK:EXTernal 140
[SENSe:]COUNter:CLocK:POLarity 142
[SENSe:]COUNter:TOTalize:IVALue 144
[SENSe:]COUNter:TOTalize:INITiate 146
[SENSe:]COUNter:TOTalize:CLEar 147
[SENSe:]COUNter:TOTalize:UDOWn:SOURce 149
[SENSe:]COUNter:TOTalize:UDOWn:DIRection 152

10 SOURce Subsystem

SOURce:VOLTage[:LEVel] 156
SOURce:VOLTage:POLarity 158
SOURce:VOLTage:RSouRCe 160
SOURce:VOLTage:RVOLtage 162
SOURce:DIGital:DATA 164
SOURce:DIGital:DATA:BIT 166

11 SYSTem Subsystem
SYSTem:CDEScription? 170
SYSTem:ERRor? 171

12 TRIGger Subsystem
TRIGger:SOURce 174
TRIGger:TYPe 176
TRIGger:DCouNT 179
TRIGger:ATRiGger:SOURce 182
TRIGger:ATRiGger:CONDition 184
TRIGger:ATRiGger:HTHReshold 187
TRIGger:ATRiGger:LTHReshold 189
TRIGger:DTRiGger:POLarity 191

13 WAVeform Subsystem
WAVeform:DATA? 194
WAVeform:POINts 196
WAVeform:STATus? 199
WAVeform:COMPlete? 201

14 Root Commands
DIGitize 204
RUN 206
STOP 208
MODel? 209
SERial? 210
DATA[:USER] 211
Using the IEEE-488.2 binary block format 213

15 IEEE-488.2 Common Commands
*CLS 216
*ESE 217

*ESR? 219
*IDN? 220
*OPC? 221
*RCL 222
*RST 224
*SAV 225
*SRE 226
*STB? 228
*TST? 229
*WAI 230

16 Error Messages

Error Messages 232
SCPI command errors 233
SCPI execution errors 236
SCPI device-specific errors 237
SCPI query errors 238
SCPI instrument errors 238

17 Factory Default Settings

Factory Default Settings 242
Command Quick Reference 246

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

List of Figures

Figure 1-1 A typical syntax diagram 20
Figure 1-2 Status system diagram 26

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

List of Tables

Table 15-1 Standard Event register mapping 217
Table 15-2 Status Byte register mapping 226
Table 16-1 List of SCPI command errors 233
Table 16-2 List of SCPI execution errors 236
Table 16-3 List of SCPI device-specific errors 237
Table 16-4 List of SCPI query errors 238
Table 16-5 List of SCPI instrument errors 238
Table 17-1 List of factory default settings 243

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

1 Introduction to SCPI

Introduction to the SCPI Language 14
SCPI Conventions and Data Formats 15
SCPI Status System 25

This chapter introduces the remote programming basics of the U2500A Series DAQ device. The SCPI programming commands provide the means to control this instrument remotely via a PC.

Introduction to the SCPI Language

SCPI, also known as the Standard Commands for Programmable Instruments, is an ASCII-based instrument command language designed for test and measurement instruments. SCPI commands defines how you communicate with an instrument from a bus controller.

They are based on a hierarchical structure, similar to the file systems used by many bus controllers. This hierarchical structure is also known as a tree system. In this system, associated commands are grouped together under a common node or root, thus forming subsystems. You must specify the complete path to execute the individual lower-level commands. A portion of the `SOURce` subsystem is shown below to illustrate the tree system.

```
SOURce
   :VOLTage
      [:LEVel] <value>, <ch_list>
      [:LEVel]? <ch_list>
SOURce
   :DIGital
      :DATA <value>, <ch_list>
      :DATA? <ch_list>
```

`SOURce` is the root keyword of the command, `VOLTage` and `DIGital` are second-level keywords, and `LEVel` and `DATA` are third-level keywords. A colon (`:`) separates a command keyword from a lower-level keyword.

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 `SOURce` is `SOUR`.

For shorter program lines, you can send the abbreviated form. For better program readability, you can send the long form. For example, in the above syntax statement, `SOURce` and `SOUR` are both acceptable forms. You can use a mixture of upper-case and lower-case letters. Therefore, `SOURCE`, `sour`, and `Sour` are all acceptable forms. Other forms, such as `SOU` and `sourc`, are not valid and will generate an error.

SCPI Conventions and Data Formats

Throughout this document, the following conventions and formats are used in the SCPI command examples. The examples are presented in the following manner:

Example

This programming snippet illustrates how several commands are used together to instruct the DAQ device to acquire 100 points of data through a single-shot acquisition process.

→ `ROUT:ENAB ON,(@101)`	*A right directional arrow (→) indicates a command that is sent to the instrument.* *The abbreviated form of the command is favoured over the long form for shorter program lines. See "Mnemonic forms" on page 14 for more information.* *Optional keywords are omitted in the command syntax. See "Square brackets" on page 19 for more information.*
→ `ACQ:SRAT 2000`	*Some commands may be omitted if the operation is not required.*[1]
→ `ACQ:POIN 100`	*The particular command or query in question is highlighted in the programming snippet.* *This example illustrates how the "ACQ:POIN" command is used within a larger programming module.*
→ `DIG`	
→ `WAV:DATA?`	*A command ending with a "?" indicates a query that is sent to the instrument. See "Using "?" commands" on page 18 for more information.*
← `#800000200<byte1><byte2>...<byte200>`	*A left directional arrow (←) indicates a return message from the instrument.*

[1] *In this instance, omitting the "ACQ:SRAT 2000" command will result in the AI sampling rate unchanged at 1000 Hz (the instrument default sampling rate).*

Command separators

Using a colon

A colon (`:`) is used to separate a command keyword from a lower-level keyword. 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 as shown below:

→ `SENS:COUN:GATE:POL AHI,(@301)`

An error is generated if you do not use the colon in your command string.

→ `SENS COUN GATE POL AHI,(@301)`

→ `SYST:ERR?`

Typical response:

← `-133,"Undefined header"`

When a colon is the first character of a command keyword, it indicates that the next command mnemonic is a root-level command.

→ `:ROUT:ENAB 1,(@201)`

This indicates that the `ROUT` command mnemonic is a root-level command. However, you can omit the leading colon if the command is the first of a new program line.

→ `ROUT:ENAB 1,(@201)`

Using a semicolon

Use a semicolon (`;`) to separate two commands within the same command string. For example, sending the following command string:

→ `:VOLT:RANG 5,(@101);:VOLT:POL BIP,(@101)`

is the same as sending the following two commands:

→ `VOLT:RANG 5,(@101)`

→ `VOLT:POL BIP,(@101)`

Note that in the first statement, the first colon is optional but the third is compulsory.

Using a comma

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

```
APPLy:SINusoid <amplitude>, <offset>, <ch_list>
```

The triangle brackets are not sent with the command string. See "Syntax conventions" on page 19 for more information.

```
→ APPL:SIN 5, 1, (@201)
```

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. You may omit the whitespace characters only in parameter lists.

For example, sending the following command:

```
→ APPL:SIN 5,1,(@201)
```

is the same as sending this command:

```
→ APPL:SIN 5, 1, (@201)
```

However, an error is generated if you do not use a whitespace character to separate a parameter from a command keyword in your command string.

```
→ APPL:SIN5,1,(@201)
→ SYST:ERR?
```

Typical response:

```
← -133,"Undefined header"
```

Using “?” commands

The bus controller may send commands at any time, but a SCPI-equipped instrument may only send responses when specifically instructed to do so.

Only query commands (commands that end with a “`?`”) will instruct the instrument to send a response message. Queries return either measured values or internal instrument settings.

For example, the following command sets the output waveform frequency to 5 kHz:

→ `OUTP:WAV:FREQ 5000`

You can then query the frequency setting in the instrument by sending:

→ `OUTP:WAV:FREQ?`

Typical response:

← `5000`

NOTE **If you send two query commands without reading the response from the first, then attempt to read the second response, you may receive some data from the first response followed by the complete second response. To avoid this, do not send a query command without reading the response. When you cannot avoid this situation, send a device clear before sending the second query command. See “Using device clear” on page 24 for more information.**

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. Common commands always begin with an asterisk (`*`), are three characters in length, and may include one or more parameters. The “`*`” commands are used to control reset, self-test, and status operations in the DAQ device.

→ `*RST; *CLS`

See Chapter 15, “IEEE-488.2 Common Commands” on page 215 for a complete list of all common commands supported.

Syntax conventions

The following SCPI conventions are used throughout this document.

Braces

Braces "`{ }`" enclose the parameter choices for a given command string. For example, the syntax statement below shows that you have to chose a mode (either `NONE`, `MAST` or `SLAV`) for the SSI configuration.

```
CONFigure:SSI {NONE|MAST|SLAV}
```

The braces are not sent with the command string. A vertical bar "`|`" separates multiple parameter choices for a given command string.

→ `CONF:SSI MAST`

Triangle brackets

Triangle brackets "`< >`" indicate that you must specify a value for the enclosed parameter. For example, the syntax statement below shows the `<value>` parameter enclosed in triangle brackets:

```
OUTPut:WAVeform:FREQuency <value>
```

The brackets are not sent with the command string. You must specify a value for the parameter:

→ `OUTP:WAV:FREQ 5000`

Square brackets

Some commands and parameters are enclosed in square brackets "`[ ]`". This indicates that the command or parameter is optional and can be omitted. For example, the syntax statement below shows that the `SENSe` root keyword is optional and can be omitted.

```
[SENSe:]VOLTage:RANGe <range>, <ch_list>
```

The brackets are not sent with the command string.

→ `VOLT:RANG 5,(@101)`

For parameters enclosed in square brackets, if you do not specify a value for the optional parameter, the instrument chooses a default value.

Syntax diagram conventions

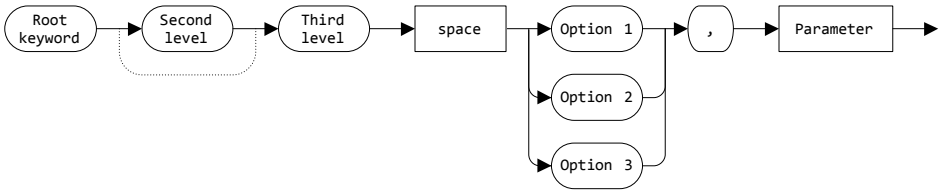


Figure 1-1 A typical syntax diagram

Solid lines represent the recommended path.

Ovals enclose the abbreviated command mnemonics. The command mnemonic must be entered exactly as shown. Ovals are also used to represent discrete parameters and command separators.

Rectangles enclose the parameters required. See "Data types and formats" on page 21 for a complete list of the various data types and formats used throughout this document. Rectangles are also used to represent whitespace characters.

Dotted lines indicate an optional path for bypassing (omitting) command keywords.

Arrows and curved intersections indicate the command path direction.

Data types and formats

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-equipped instruments are precise talkers. This means that SCPI-equipped instruments always respond to a particular query in a predefined, rigid format.

Numeric

Parameters that accepts all commonly used decimal representations of numbers including optional signs, and decimal points, scientific notations (3e2 = 3×10^2, 5.43e–3 = 5.43×10^{-3}, or 10e6 = 10×10^6), and engineering notations (M, k, m, µ, n). Special values for numeric parameters such as `AUTO`, `MIN`, `MAX`, and `DEF` are also accepted. If only specific numeric values are accepted, the instrument will automatically round the input numeric parameters. As an example, the following command requires a numeric parameter for the range value:

```
[SENSe:]VOLTage:RANGe <range>, <ch_list>
```

NOTE **While most SCPI command sets supports the use of scientific and engineering notations, the U2500A Series DAQ device does not support the use of scientific and engineering notation. Magnitude prefixes like *mega*- (e = 6), *kilo*- (e = 3), *milli*- (e = –3), *micro*- (e = –6), or *nano*- (e = –9) are mentioned in the parameter range of values whenever applicable.**

Discrete

Parameters used to program settings that have a limited number of values such as `INTernal`, `EXternal`, and `CCG`. Some of these parameters have a short form and a long form just like command keywords. You can mix upper-case and lower-case letters. Query responses will always return the abbreviated form in all upper-case letters. As an example, the following command require discrete parameters for the digital channel direction:

```
CONFigure:DIGital:DIRection {INPut|OUTput}, <ch_list>
```

Boolean

Parameters that represent a single binary condition that is either true or false. For a false condition, the DAQ device will accept `OFF` or `0`. For a true condition, the DAQ device will accept `ON` or `1`. When you query a boolean setting, the DAQ device will always return `0` or `1`. As an example, the following command require boolean parameters for the instrument output status:

```
OUTPut {0|OFF|1|ON}
```

String

Parameters that contain virtually any set of ASCII characters. A string must begin and end with matching quotes; either with a single quote or a double quote. You can include the quote delimiter as part of the string by typing it twice without any characters in between.[1]

Channel list

Channel list parameters have the form `(@ccc)`, where `ccc` is the channel number. You can specify a single channel, multiple channels, or a range of channels. The channel list must be preceded with the "`@`" symbol and must be enclosed in parentheses "`( )`". As an example, the following commands use a channel list parameter:

```
ROUTe:ENABle <mode>, <ch_list>
```

The following command enables channel 101 for input operations.

→ `ROUT:ENAB ON,(@101)`

Consecutive channels can be specified using a colon or comma. The following command enables channel 101 and 104 for input operations.

→ `ROUT:ENAB ON,(@101,104)`

The following command enables channel 101 through 103 for input operations and channel 201 for output operations.

→ `ROUT:ENAB ON,(@101:103,201)`

When you specify a range of channels, any channel that is invalid will generate an error.

[1] The U2500A Series DAQ device currently does not support any command with string parameters.

When more than one channel is specified in the command parameter, the U2500A Series DAQ device will acquire the measurement points alternately from each specified channel when a readback query is sent (`WAV:DATA?` or `MEAS?`).

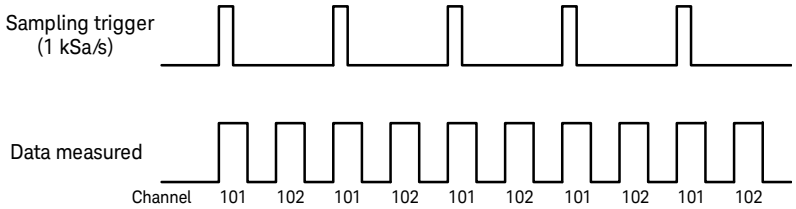
Example 1

→ `ROUT:ENAB ON,(@101,102)`

→ `ACQ:SRAT 1000;:ACQ:POIN 3;:DIG`

→ `WAV:DATA?`

→ `#800000012<byte1_ch101><byte1_ch102><byte2_ch101><byte2_ch102>...<byte6_ch101><byte6_ch102>`



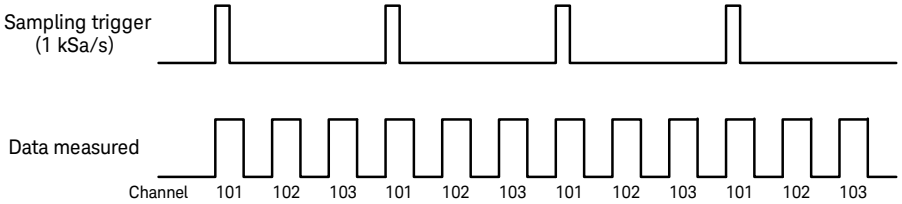
Example 2

→ `ROUT:ENAB ON,(@101:103)`

→ `ACQ:SRAT 1000;:WAV:POIN 1000;:RUN`

→ `WAV:DATA?`

← `#800006000<byte1_ch101><byte1_ch102><byte1_ch103><byte2_ch101><byte2_ch102><byte2_ch103><byte3_ch101><byte3_ch102><byte3_ch103>...<byte2000_ch101><byte2000_ch102><byte2000_ch103>`



Input message terminators

Program messages sent to a SCPI-equipped 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.

Using device clear

Device clear is an IEEE-488 low-level bus message that you can use to return the instrument to a responsive state (for example, during a lengthy query).

Different programming languages and IEEE-488 interface cards provide access to this capability through their own unique commands. 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:

1. If a scan is running, it is aborted.
2. The instrument returns to the trigger “idle” state.
3. The instrument's input and output buffers are cleared.
4. The instrument is prepared to accept a new command string.

An overlapped command, if any, will be terminated with no “Operation Complete” indication.

NOTE **It is recommended that you allow for a two-second wait following a device clear to enable the instrument to process the clear operation.**

SCPI Status System

The status system records various instrument conditions and states in several register groups. Each register group is made up of several low-level registers called the Condition register, Event register, and Enable register which control the action of specific bits within the register group.

A Condition register continuously monitors the state of the instrument. The bits in the condition register are updated in real-time and the bits are not latched or buffered. This is a read-only register and the bits are not cleared when you read the register.

An Event register latches the various events from the changes in the Condition register. There is no buffering in this register; while an event bit is set, subsequent events corresponding to that bit are ignored. This is a read-only register. Once a bit is set, it remains set until cleared by a query or clear status (*CLS) command.

An Enable register defines which bits in the event register will be reported to the Status Byte register group. You can write to or read from an Enable register.

The relationship between various registers in the U2500A Series DAQ device SCPI status system is shown in Figure 1-2.

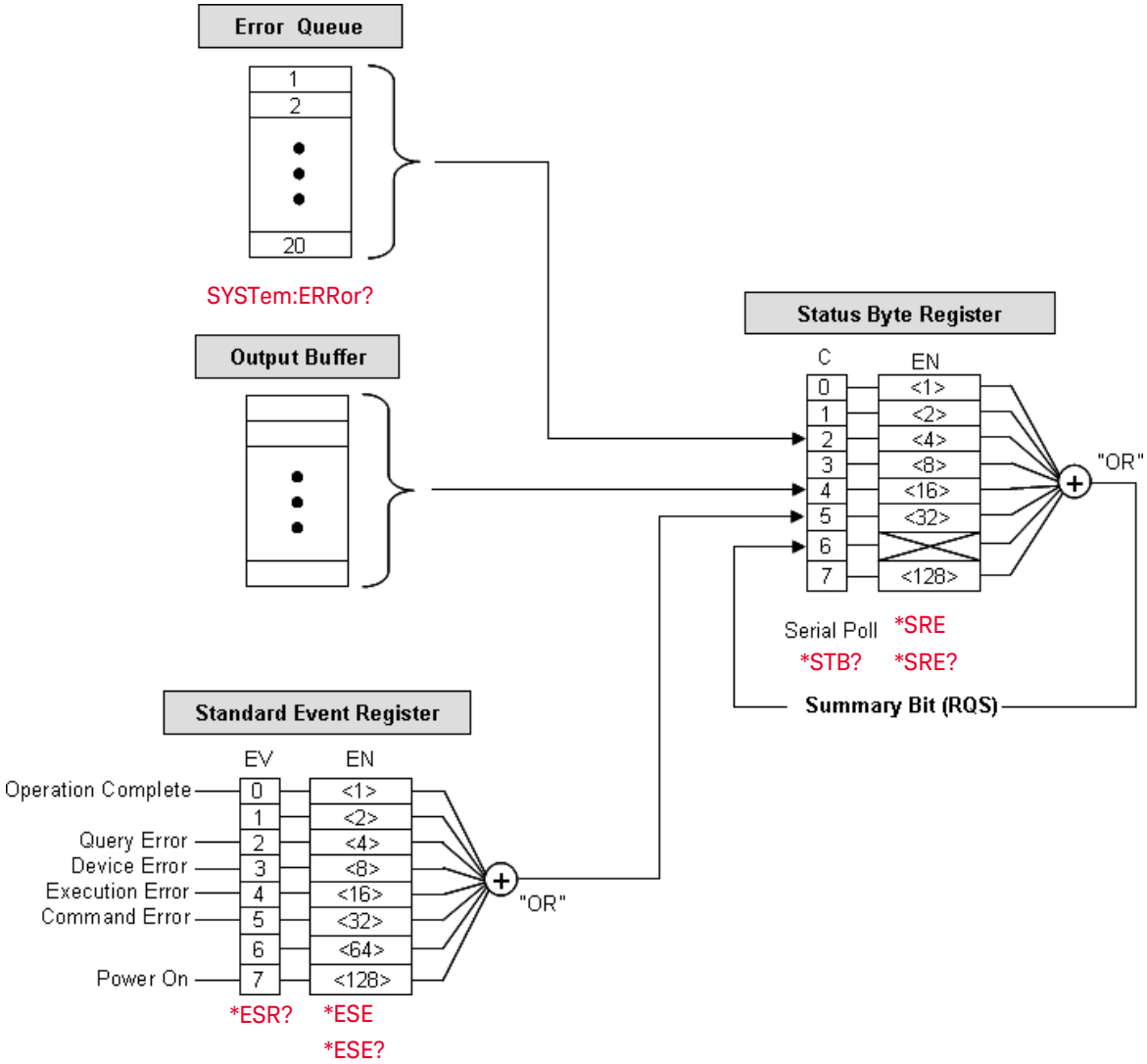


Figure 1-2 Status system diagram

Standard Event register

The Standard Event register group reports the following types of instrument events: power-on detected, command syntax errors, command execution errors, device errors (self-test or calibration), or query errors. All of these conditions can be reported in the Standard Event summary bit through the enable register. To set the enable register mask, key in a decimal value to the register using the event status enable (*ESE) command.

Bit definitions: Standard Event register

Bit number	Decimal value	Definition
0 Operation complete	1	All command have been executed.
1 Not used	Not used	“0” is returned.
2 Query error	4	The instrument tried to read the output buffer but it was empty. Or, a new command line was received before a previous query has been read. Or, both the input and output buffers are full.
3 Device error	8	A self-test, calibration, or other device-specific error has occurred.
4 Execution error	16	An execution error occurred.
5 Command error	32	A command syntax error occurred.
6 Not used	Not used	“0” is returned.
7 Power-on	128	Power has been turned off and on since the last time the event register was read or cleared.

The event register in the Standard Event is cleared when:

- you execute the clear status (*CLS) command, or
- you read the event register using the event status register (*ESR?) command.

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

NOTE **When a command, execution, device, or query error have occurred, a related error message will be generated. For a complete listing of all error messages, refer to Chapter 16, “Error Messages” on page 231.**

Status Byte register

The Status Byte register group reports the conditions from the other status registers. Clearing an event register from one of the other registers will clear the corresponding bits in the Status Byte condition register. Data that is waiting in the U2500A Series DAQ device output buffer is immediately reported on the “Message Available” bit (bit 4).

Bit definitions: Status Byte register

Bit number	Decimal value	Definition
0 Not used	Not used	“0” is returned.
1 Not used	Not used	“0” is returned.
2 Error queue	4	One or more errors have been stored in the Error Queue. Use the `SYSTem:ERRor?` query to read and delete errors.
3 Not used	Not used	“0” is returned.
4 Message available	16	Data is available in the instrument output buffer for single SCPI queries (for example, `*OPC?`, `MEAS? (@101)`, `*IDN?, SYST:ERR?`, and so on) but not in the continuous acquisition output buffer.
5 Standard Event summary	32	One or more bits are set in the Standard Event register. Bits must be enabled using the *ESE command.
6 Master Status summary	64	One or more bits are set in the Status Byte Register and may generate a Request for Service (RQS). Bits must be enabled using the *SRE command.
7 Not used	Not used	“0” is returned.

The Status Byte condition register will be cleared when:

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

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

NOTE Refer to Chapter 15, “IEEE-488.2 Common Commands” on page 215 for more details on the common commands mentioned above.

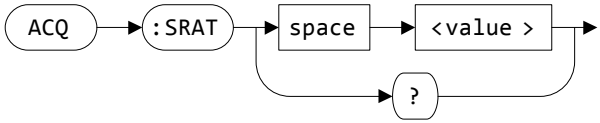
2 ACQuire Subsystem

ACQuire:SRATe 30
ACQuire:POINts 32

This chapter describes the ACQuire commands used to program the U2500A Series DAQ device over a remote interface. Use the ACQuire commands to set the sampling rate for the DAQ device input channels. You can also set the number of acquisition points for the single-shot acquisition process.

ACQuire:SRATe

Syntax



`ACQuire:SRATe <value>`

This command is used to set the sampling rate of the analog input (AI) channels.

`ACQuire:SRATe?`

This query returns a numeric value that represents the instrument sampling rate. The value returned is expressed in hertz (Hz).

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	– U2531A: 3 Hz to 2000000 Hz (2 MHz) – U2541A: 3 Hz to 250000 (250 kHz) – U2542A: 3 Hz to 500000 (500 kHz)	1000 Hz

Remarks

The sampling rate (or sampling frequency) defines the number of samples per second taken from a continuous signal to make a discrete signal. For time-domain signals, it is measured in hertz (Hz). The inverse of the sampling frequency is the sampling period or sampling interval, which is the time between samples.

NOTE **To initiate an acquisition process, use either the `DIGitize` command (for single-shot acquisition) or `RUN` command (for continuous acquisition).**

See also

"ROUTe:ENABle" on page 117

"DIGitize" on page 204

"RUN" on page 206

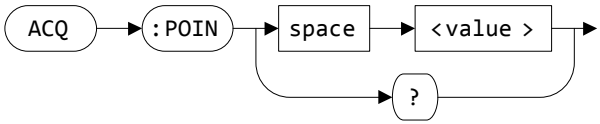
Example

This example illustrates how the **ACQuire:SRATe** command is used to set the AI sampling rate.

→ ROUT:ENAB ON,(@101)	*This command enables channel 101 for input operations.*
→ ACQ:SRAT 2000	*This command sets the AI sampling rate to 2 kHz. This implies that the measurement is taken every 0.5 ms (1 / 2 kHz).*
→ ACQ:POIN 100	*This command sets the instrument to acquire 100 sample points.*
→ DIG	*This command starts the single-shot acquisition process.*
→ WAV:COMP?	*This query returns the acquisition status of the instrument.*
← YES	*The returned value is YES, indicating that the data acquisition is completed and ready to be read back.*
→ WAV:DATA?	*This query returns the raw data acquired from channel 101.*
← #800000200<byte1><byte2>... <byte200>	*The acquired data is returned.*

ACQuire:POINts

Syntax



`ACQuire:POINts <value>`

This command is used to set the number of acquisition points for the single-shot acquisition process.

`ACQuire:POINts?`

This query returns a numeric value that represents the number of acquisition points set for the single-shot acquisition process.

NOTE The single-shot acquisition is initiated by the `DIGitize` command. Use the `STOP` command to stop the single-shot acquisition process before the number of points specified is reached.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	Maximum of 8 Msa	500

Remarks

- This command performs the same functions as the `WAVeform:POINts` command. The only difference is that the `WAVeform:POINts` command is used for continuous acquisitions, while the `ACQuire:POINts` command is used for single-shot acquisitions.
- Each acquisition point is made up of two bytes. Hence, setting the acquisition points to 100 implies that a block of 200 bytes of raw data is transferred to the PC when the `WAVeform:DATA?` query is issued.

See also

"ACQuire:SRATe" on page 30

"ROUTe:ENABle" on page 117

"WAVeform:COMPlete?" on page 201

"WAVeform:DATA?" on page 194

"WAVeform:POINts" on page 196

"DIGitize" on page 204

"STOP" on page 208

Example

This example illustrates how the **ACQuire:POINts** command is used to acquire 100 sample points (200 bytes) of raw data from AI channel 101.

→ `ROUT:ENAB ON,(@101)`	*This command enables channel 101 for input operations.*
→ `ACQ:SRAT 1000`	*This command sets the AI sampling rate to 1 kHz.*
→ `ACQ:POIN 100`	*This command sets the instrument to acquire 100 sample points.*
→ `DIG`	*This command starts the single-shot acquisition process.*
→ `WAV:COMP?`	*This query returns the acquisition status of the instrument.*
← `YES`	*The returned value is YES, indicating that the data acquisition is completed and ready to be read back.*
→ `WAV:DATA?`	*This query returns the raw data acquired from channel 101.*
← `#800000200<byte1><byte2>...<byte200>`	*The acquired data is returned.*

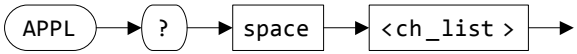
3 APPLy Subsystem

APPLy? 36
APPLy:SINusoid 37
APPLy:SQUare 39
APPLy:SAWTooth 41
APPLy:TRIangle 43
APPLy:NOISe 45
APPLy:USER 47

This chapter describes the APPLy commands used to program the U2500A Series DAQ device over a remote interface. Use the APPLY commands to output a sinusoidal, square, sawtooth, triangle, noise, or an arbitrary waveform from the output channels.

APPLy?

Syntax



`APPLy? <ch_list>`

This query returns a series of comma-separated fields indicating the current function, amplitude, and offset value of each analog output (AO) channel specified. The amplitude and offset values returned are expressed in volts (V). The abbreviated form of the function name is always returned.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	(@201)|(@202)	-

Example

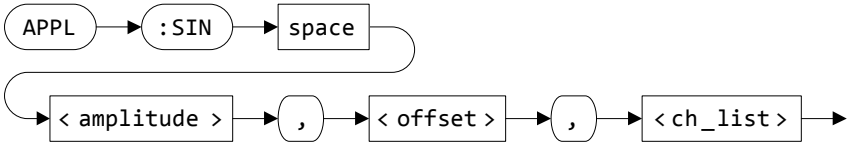
This example illustrates how the **APPLy?** query is used to return the current function, amplitude, and offset value for AO channel 201 and 202.

→ `APPL? (@201,202)`	*This query returns the current function, amplitude, and offset value for channel 201 and 202.*

← `SIN,10,0,SQU,5,0`

APPLy:SINusoid

Syntax



`APPLy:SINusoid <amplitude>, <offset>, <ch_list>`

This command is used to send a sinusoidal wave with the specified amplitude and DC offset to the instrument buffer. The waveform is generated from the AO channel(s) specified in the `<ch_list>` parameter after the `OUTPut ON` command is issued.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
amplitude	Numeric	[Amplitude value] V[a]	10 V
offset	Numeric	[Offset value] V[a]	0 V
ch_list	Channel list	(@201)|(@202)	-

[a] The amplitude and offset parameters are subjected to the following constraints:
Amplitude must be > zero.
[Amplitude + Offset] must be ≤ 10 and [Amplitude × (–1) + Offset] must be ≥ –10

See also

"OUTPut" on page 80

"OUTPut:WAVeform:ITERate" on page 82

"OUTPut:WAVeform:SRATe" on page 84

"OUTPut:WAVeform:FREQuency" on page 87

"ROUTe:ENABle" on page 117

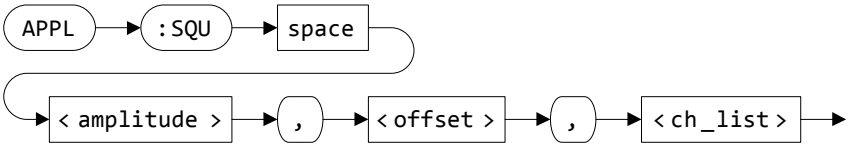
Example

This example illustrates how the **`APPLy:SINusoid`** command is used to set a 5 V sinusoidal wave with no DC offset for AO channel 201.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:WAV:ITER 0`	*This command causes the instrument to generate the waveform indefinitely.*
→ `OUTP:WAV:SRAT 0`	*This command automatically selects a suitable sampling rate for the output waveform.*
→ `OUTP:WAV:FREQ 5000`	*This command sets the output waveform frequency to 5 kHz.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201.*

APPLy:SQUare

Syntax



`APPLy:SQUare <amplitude>, <offset>, <ch_list>`

This command is used to send a square wave with the specified amplitude and DC offset to the instrument buffer. The waveform is generated from the AO channel(s) specified in the `<ch_list>` parameter after the `OUTPut ON` command is issued.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
amplitude	Numeric	[Amplitude value] V[a]	10 V
offset	Numeric	[Offset value] V[a]	0 V
ch_list	Channel list	(@201)|(@202)	-

[a] The amplitude and offset parameters are subjected to the following constraints:
Amplitude must be > zero.
[Amplitude + Offset] must be ≤ 10 and [Amplitude × (–1) + Offset] must be ≥ –10

See also

"OUTPut" on page 80

"OUTPut:WAVeform:FREQuency" on page 87

"ROUTe:ENABle" on page 117

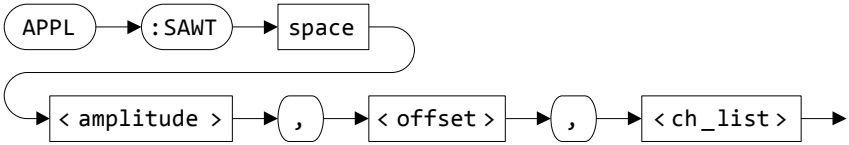
Example

This example illustrates how the **`APPLy:SQUare`** command is used to set a 3 V square wave with 1 V DC offset for AO channel 202.

→ `ROUT:ENAB ON,(@202)`	*This command enables channel 202 for output operations.*
→ `APPL:SQU 3,1,(@202)`	*This command sends a square waveform to channel 202 with an amplitude of 3 V (6 V_{pp}) and a DC offset of 1 V.*
→ `OUTP:WAV:ITER 0`	*This command causes the instrument to generate the waveform indefinitely.*
→ `OUTP:WAV:SRAT 0`	*This command automatically selects a suitable sampling rate for the output waveform.*
→ `OUTP:WAV:FREQ 5000`	*This command sets the output waveform frequency to 5 kHz.*
→ `OUTP ON`	*This command generates the square waveform from channel 201.*

APPLy:SAWTooth

Syntax



`APPLy:SAWTooth <amplitude>, <offset>, <ch_list>`

This command is used to send a sawtooth wave with the specified amplitude and DC offset to the instrument buffer. The waveform is generated from the AO channel(s) specified in the `<ch_list>` parameter after the `OUTPut ON` command is issued.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
amplitude	Numeric	[Amplitude value] V[a]	10 V
offset	Numeric	[Offset value] V[a]	0 V
ch_list	Channel list	(@201)|(@202)	-

[a] The amplitude and offset parameters are subjected to the following constraints:
Amplitude must be > zero.
[Amplitude + Offset] must be ≤ 10 and [Amplitude × (–1) + Offset] must be ≥ –10

See also

"OUTPut" on page 80

"OUTPut:WAVeform:FREQuency" on page 87

"ROUTe:ENABle" on page 117

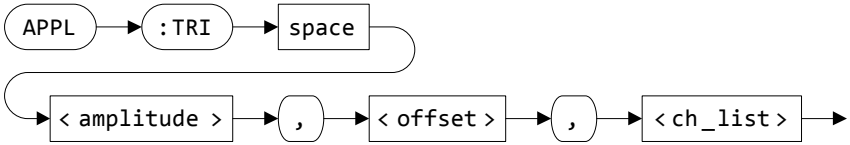
Example

This example illustrates how the **APPLy:SAWTooth** command is used to set a 4.5 V sawtooth waveform with no DC offset for AO channel 202. Note that the settings set by the **OUTPut:WAVeform** commands are applicable in all AO channels.

→ `ROUT:ENAB ON,(@201,202)`	*This command enables channel 201 and 202 for output operations.*
→ `APPL:SIN 10,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 10 V (20 V_{pp}) and a DC offset of 0 V.*
→ `APPL:SAWT 4.5,0,(@202)`	*This command sends a sawtooth waveform to channel 202 with an amplitude of 4.5 V (9 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:WAV:ITER 0`	*This command causes the instrument to generate the waveform indefinitely.*
→ `OUTP:WAV:SRAT 0`	*This command automatically selects a suitable sampling rate for the output waveform.*
→ `OUTP:WAV:FREQ 5000`	*This command sets the output waveform frequency to 5 kHz.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201 and the sawtooth waveform from channel 202.*

APPLy:TRIangle

Syntax



APPLy:TRIangle <amplitude>, <offset>, <ch_list>

This command is used to send a triangle wave with the specified amplitude and DC offset to the instrument buffer. The waveform is generated from the AO channel(s) specified in the **<ch_list>** parameter after the **OUTPut ON** command is issued.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
amplitude	Numeric	[Amplitude value] V[a]	10 V
offset	Numeric	[Offset value] V[a]	0 V
ch_list	Channel list	(@201)|(@202)	-

[a] The amplitude and offset parameters are subjected to the following constraints:
Amplitude must be > zero.
[Amplitude + Offset] must be ≤ 10 and [Amplitude × (–1) + Offset] must be ≥ –10

See also

"OUTPut" on page 80

"OUTPut:WAVeform:FREQuency" on page 87

"ROUTe:ENABle" on page 117

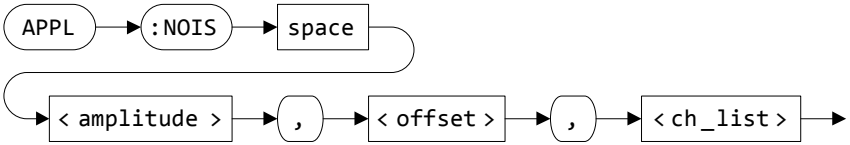
Example

This example illustrates how the **APPLy:TRIangle** command is used to set a 5 V triangle wave with –2 V DC offset for AO channel 201.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:TRI 5,-2,(@201)`	*This command sends a triangle waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of –2 V.*
→ `OUTP:WAV:ITER 0`	*This command causes the instrument to generate the waveform indefinitely.*
→ `OUTP:WAV:SRAT 0`	*This command automatically selects a suitable sampling rate for the output waveform.*
→ `OUTP:WAV:FREQ 5000`	*This command sets the output waveform frequency to 5 kHz.*
→ `OUTP ON`	*This command generates the triangle waveform from channel 201.*

APPLy:NOISe

Syntax



APPLy:NOISe <amplitude>, <offset>, <ch_list>

This command is used to send a noise wave with the specified amplitude and DC offset to the instrument buffer. The waveform is generated from the AO channel(s) specified in the **<ch_list>** parameter after the **OUTPut ON** command is issued.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
amplitude	Numeric	[Amplitude value] V[a]	10 V
offset	Numeric	[Offset value] V[a]	0 V
ch_list	Channel list	(@201)|(@202)	-

[a] The amplitude and offset parameters are subjected to the following constraints:
Amplitude must be > zero.
[Amplitude + Offset] must be ≤ 10 and [Amplitude × (–1) + Offset] must be ≥ –10

See also

"OUTPut" on page 80

"OUTPut:WAVeform:FREQuency" on page 87

"ROUTe:ENABle" on page 117

Example

This example illustrates how the **APPLy:NOISe** command is used to set a 10 V noise wave with no DC offset for AO channel 201 and 202. Note that the settings set by the **OUTPut:WAVeform** commands are applicable in all AO channels.

→ `ROUT:ENAB ON,(@201,202)`	*This command enables channel 201 and 202 for output operations.*
→ `APPL:NOIS 10,0,(@201,202)`	*This command sends a noise waveform to channel 201 and 202 with an amplitude of 10 V (20 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:WAV:ITER 0`	*This command causes the instrument to generate the waveform indefinitely.*
→ `OUTP:WAV:SRAT 0`	*This command automatically selects a suitable sampling rate for the output waveform.*
→ `OUTP:WAV:FREQ 5000`	*This command sets the output waveform frequency to 5 kHz.*
→ `OUTP ON`	*This command generates the noise waveform from channel 201 and 202.*

APPLy:USER

Syntax

`APPLy:USER <ch_list>`

This command is used to send a user-defined pattern to the instrument buffer. The arbitrary waveform is generated from the AO channel(s) specified in the `<ch_list>` parameter after the `OUTPut ON` command is issued.

NOTE For more information on how to send a user-defined pattern to the instrument buffer see "DATA[:USER]" on page 211.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	(@201)|(@202)	-

Remarks

The `OUTPut ON` command will return an error if channel 201 and 202 is configured to user-defined and predefined (`SIN`, `SQU`, `SAWT`, `TRI`, or `NOIS`) waveform patterns at the same time.

You cannot output a user-defined pattern if the waveform sampling rate is set to automatic (`OUTPut:WAVEform:SRATe 0`). To select an AO waveform sampling rate value, see "OUTPut:WAVeform:SRATe" on page 84.

See also

"OUTPut:WAVeform:SRATe" on page 84

"ROUTe:ENABle" on page 117

"DATA[:USER]" on page 211

Example

This example illustrates how the **DATA[:USER]** command is used with the **APPLy:USER** command to output an arbitrary waveform via channel 201.

→ `ROUT:ENAB OFF,(@202)`	*This command disables output operations for channel 202. This ensures that we do not accidentally output a predefined waveform from channel 202.*
→ `ROUT:ENAB ON,(@201)`	*This command enables output operations for channel 201.*
→ `DATA #800002000<byte1> <byte2>...<byte2000>`	*This command sends 1000 points (2000 bytes) of user-defined data to the instrument buffer.*
→ `OUTP:WAV:ITER 0`	*This command causes the instrument to generate the waveform indefinitely.*
→ `OUTP:WAV:SRAT 2000`	*This command sets the AO waveform sampling rate to 2 kHz.*
→ `APPL:USER (@201)`	*This command sends the user-defined pattern to channel 201.*
→ `OUTP ON`	*This command generates the user-defined pattern from channel 201.*

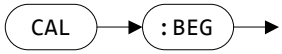
4 CALibration Subsystem

CALibration:BEGin 50

This chapter describes the CALibration command used to program the U2500A Series DAQ device over a remote interface. Use the CALibration:BEGin command to carry out the self-calibration function on your DAQ device.

CALibration:BEGin

Syntax



`CALibration:BEGin`

This command is used to calibrate your DAQ device. When this command is executed, the DAQ device goes into a self-calibration mode.

NOTE For more information on the self-calibration procedures, see Chapter 5, "Calibration" in the *Keysight U2500A Series USB Simultaneous Sampling Multifunction Data Acquisition User's Guide*.

Supported models

U2531A, U2541A, U2542A

Remarks

The functions of the DAQ device will be halted until the self-calibration process is completed. Use the `*OPC?` query to check if the calibration is completed. The calibration is complete if the returned value is "`+1`".

Example

This example illustrates how the `CALibration:BEGin` command is used to begin the self-calibration process.

→ `CAL:BEG`	*This command begins the self-calibration process.*
→ `*OPC?`	
← `+1`	*The returned value "+1" indicates that the self-calibration process has been completed.*

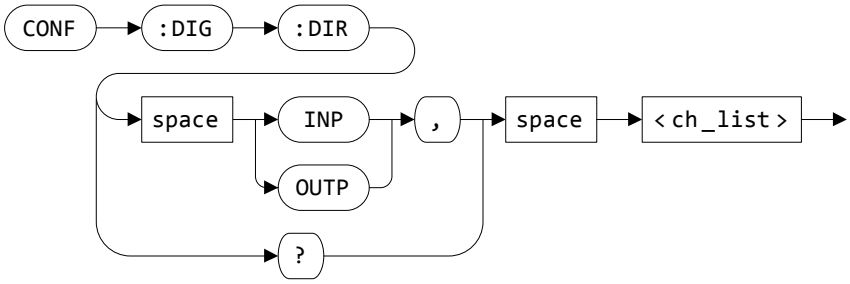
5 CONFigure Subsystem

CONFigure:DIGital:DIRection 52
CONFigure:TIMEbase:SOURce 54
CONFigure:TIMEbase:ECLocK 56
CONFigure:SSI 58

This chapter describes the CONFigure commands used to program the U2500A Series DAQ device over a remote interface. Use the CONFigure:DIGital:DIRection command to set the digital port for input or output operations. You can also select a clock source for both AI and AO channels.

CONFigure:DIGital:DIRection

Syntax



`CONFigure:DIGital:DIRection <direction>, <ch_list>`

This command is used to set the digital port direction for the digital input/output (DIO) channel(s) specified in the `<ch_list>` parameter.

The valid options are:

- INPut: Configures channel for input operations.
- OUTPut: Configures channel for output operations.

`CONFigure:DIGital:DIRection? <ch_list>`

This query returns a string value that represents the digital port direction of each DIO channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
direction	Discrete	INPut|OUTPut	INPut
ch_list	Channel list	(@501)|(@502)|(@503)|(@504)	-

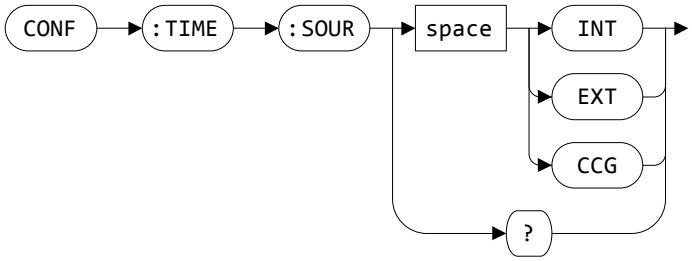
Example

This example illustrates how the **CONFigure:DIGital:DIRection** command is used to enable output operations for channel 501.

→ `CONF:DIG:DIR OUTP,(@501)`	*This command enables output operations for channel 501.*
→ `CONF:DIG:DIR? (@501:503)`	*This query returns the digital port direction for channel 501, 502, and 503.*
← `OUTP,INP,INP`	*The returned value for channel 501 is "OUTP", indicating output operations; whereas the returned value for channel 502 and 503 is "INP", indicating input operations.*

CONFigure:TIMEbase:SOURce

Syntax



`CONFigure:TIMEbase:SOURce <mode>`

This command is used to set the timebase/clock source for both AI and AO channels.

The valid options are:

- INTernal: Selects the DAQ device internal clock frequency as the timebase/clock source.
- EXTernal: Selects the EXT_TIMEBASE pin as the timebase/clock source.

NOTE

- **The clock frequency coming in from the EXT_TIMEBASE pin must be set in the DAQ device via the `CONFigure:TIMEbase:ECLocK` command. For example, if a 10 MHz clock frequency is expected from the EXT_TIMEBASE pin, you must send the `CONFigure:TIMEbase:ECLocK 10000` command to the DAQ device.**
- **See "CONFigure:TIMEbase:ECLocK" on page 56 for more information on how to set the external clock frequency.**

- CCG (Card cage): Selects the U2781A USB modular instrument chassis clock frequency as the timebase/clock source.

`CONFigure:TIMEbase:SOURce?`

This query returns a string value that represents the timebase/clock source for both AI and AO channels.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
mode	Discrete	INTernal|EXTernal|CCG	INTernal

See also

"CONFigure:TIMEbase:ECLocK" on page 56

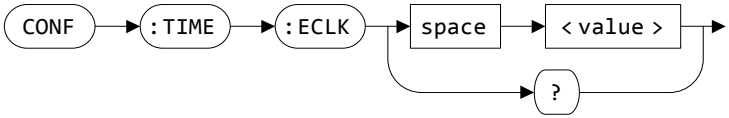
Example

This example illustrates how the **`CONFigure:TIMEbase:SOURce`** command is used to set the AI and AO channels timebase/clock source to the EXT_TIMEBASE pin.

→ `CONF:TIME:SOUR EXT`	*This command sets the instrument timebase/clock source to the EXT_TIMEBASE pin. A 20 MHz clock frequency signal is supplied to the EXT_TIMEBASE pin.*
→ `CONF:TIME:ECLK 20000`	*Since we are expecting a 20 MHz clock frequency from the EXT_TIMEBASE pin, this command is used to set the external clock frequency to 20 MHz.*
→ `CONF:TIME:SOUR?`	*This query returns the timebase/clock source for the AI and AO channels.*
← `EXT`	

CONFigure:TIMEbase:ECLocK

Syntax



`CONFigure:TIMEbase:ECLocK <value>`

This command is used to set the external clock frequency. The frequency set in the `<value>` parameter must be the same as the frequency value supplied to the EXT_TIMEBASE pin when the timebase/clock source is set to `EXTernal`. The clock frequency is expressed in units of kilohertz (kHz).

NOTE See "CONFigure:TIMEbase:SOURce" on page 54 for more information on selecting a timebase/clock source.

`CONFigure:TIMEbase:ECLocK?`

This query returns a numeric value that represents the external clock frequency. The value returned is expressed in kilohertz (kHz).

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Discrete	10000 kHz (10 MHz) to 48000 kHz (48 MHz)	10000 kHz

See also

"CONFigure:TIMEbase:SOURce" on page 54

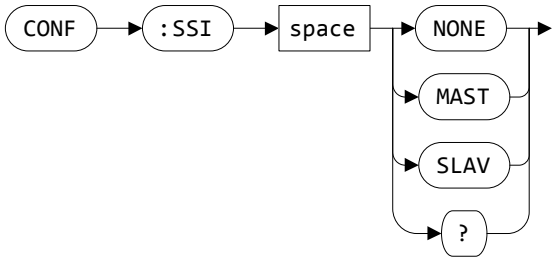
Example

This example illustrates how the **CONFigure:TIMEbase:ECLocK** command is used to set the external clock frequency.

→ `CONF:TIME:SOUR EXT`	*This command sets the instrument timebase/clock source to EXT_TIMEBASE pin. A 20 MHz clock frequency signal is supplied to the EXT_TIMEBASE pin.*
→ `CONF:TIME:ECLK 20000`	*Since we are expecting a 20 MHz clock frequency from the EXT_TIMEBASE pin, this command is used to set the external clock frequency to 20 MHz.*
→ `CONF:TIME:ECLK?`	*This query returns the external clock frequency for the AI and AO channels.*
← `20000`	

CONFigure:SSI

Syntax



`CONFigure:SSI <mode>`

This command is used to set or disable the simultaneous synchronization status of the DAQ device when used in the U2781A USB modular instrument chassis.

The valid options are:

- NONE: Disables the SSI state.
- MAST (Master): Configures the module to drive the triggering signal onto the common trigger bus.
- SLAV (Slave): Configures the module as a listener. The module will monitor the common trigger bus for any incoming triggering signals.

`CONFigure:SSI?`

This query returns a string value that represents the simultaneous synchronization status of the DAQ device when used in the U2781A USB modular instrument chassis.

Supported models

U2531A, U2541A, U2542A

Parameter

	Item	Range of values	Default value
mode	Discrete	NONE|MAST|SLAV	NONE

NOTE

- This command is only operational when used in the U2781A USB modular instrument chassis.
- There should only be one master assigned at any one time.
- For more information on the Synchronous Simultaneous Interface (SSI), refer to the *U2781A USB Modular Instrument Chassis* User's Guide.

Remarks

- This command will force all reference clock sources to the synchronous simultaneous interface.
- Once a module is configured as a slave (`CONFigure:SSI SLAV`), it becomes a listener and listens for a triggering signals on the backplane of the U2781A chassis. To undo this listener behavior, a `CONFigure:SSI NONE` command must be issued to the slave module.

Example

This example illustrates how the `CONFigure:SSI` command is used to set the DAQ device as a slave module to monitor incoming triggering signals when used in the U2781A USB modular instrument chassis.

→ `SYST:CDES?`	*This query returns the slot and chassis number that the DAQ device is plugged into.*
← `2,1`	*The returned value "2,1" indicates that the DAQ device is slotted into the second slot of the first chassis.*
→ `CONF:SSI SLAV`	*This command sets the synchronization status of the instrument as a slave.*

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

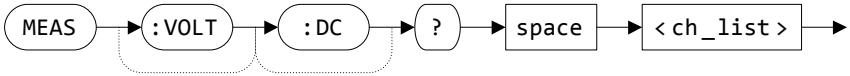
6 MEASure Subsystem

MEASure[:VOLTage][:DC]? 62
MEASure:COUNter:DATA? 64
MEASure:COUNter:FREQuency? 66
MEASure:COUNter:PERiod? 68
MEASure:COUNter:PWIDth? 70
MEASure:COUNter:TOTalize? 72
MEASure:DIGital? 74
MEASure:DIGital:BIT? 76

This chapter describes the MEASure commands used to program the U2500A Series DAQ device over a remote interface. Use the MEASure[:VOLTage][:DC]? query to make a single voltage measurement. You can also return the frequency, period, pulse width, and totalizer values.

MEASure[:VOLTage][:DC]?

Syntax



MEASure[:VOLTage][:DC]? <ch_list>

This query is used to make a single voltage measurement. This query returns a numeric value that represents the scaled voltage value from each AI channel specified. Multiple responses are separated by commas. The value returned is expressed in volts (V).

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	Differential mode: (@101) to (@104)	-

Remarks

- If the returned value is **999.9** V, the measurement voltage is out of range.
- To set the measurement range and polarity of an AI channel, use the **[SENSe:]VOLTage:RANGe** and **[SENSe:]VOLTage:POLarity** commands.

See also

"[SENSe:]VOLTage:RANGe" on page 120

"[SENSe:]VOLTage:POLarity" on page 122

"[SENSe:]VOLTage:STYPe?" on page 124

"[SENSe:]VOLTage:AVERage" on page 125

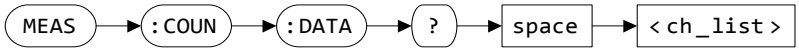
Example

This example illustrates how the `MEASure[:VOLTage][:DC]?` query is used to acquire a single voltage reading from AI channels 102 to 104.

→ `MEAS? (@102:104)`	*This query returns a single voltage measurement from channel 102, 103, and 104.*
← `20.3,40.1,999.9`	*A single voltage reading is returned from each specified channel. The returned value “999.9” for channel 104 indicates that the measurement voltage is out of range.*

MEASure:COUNter:DATA?

Syntax




```
MEASure:COUNter:DATA? <ch_list>
```

This query returns a numeric value that represents the value of the specific counter function selected (frequency, period, pulse width, or totalizer). The value returned is based on the current counter function the counter channel specified. Multiple responses are separated by commas.

NOTE For more information on selecting a counter function, see "[SENSe:]COUNter:FUNCtion" on page 127.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	(@301)|(@302)	-

See also

"[SENSe:]COUNter:FUNCtion" on page 127

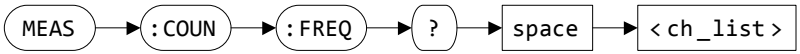
Example

This example illustrates how the `MEASure:COUNter:DATA?` query is used to return the counter value for the specified counting mode selected by the `[SENSe:]COUNter:FUNCtion` command.

→ `COUN:FUNC FREQ,(@301)`	*This command sets channel 301 to measure frequency.*
→ `MEAS:COUN:DATA? (@301)`	*This query returns the frequency for channel 301.*
← `5.599627`	*The frequency is returned in kHz.*
→ `COUN:FUNC PER,(@301)`	*This command sets channel 301 to measure period.*
→ `MEAS:COUN:DATA? (@301)`	*This query returns the period for channel 301.*
← `0.1818333`	*The period is returned in ms.*
→ `COUN:FUNC PWID,(@302)`	*This command sets channel 302 to measure pulse width.*
→ `MEAS:COUN:DATA? (@302)`	*This query returns the pulse width for channel 302.*
← `0.12725`	*The pulse width is returned in ms.*

MEASure:COUNter:FREQuency?

Syntax



`MEASure:COUNter:FREQuency? <ch_list>`

This query returns a numeric value that represents the frequency of each counter channel specified. Multiple responses are separated by commas. The value returned is expressed in kilohertz (kHz).

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	(@301)|(@302)	-

Remarks

- Executing the **`MEASure:COUNter:FREQuency?`** query will automatically change the current function of the counter channel specified. You do not need to select the frequency measurement function (via the **`SENSe:COUNter:FUNCtion FREQuency`** command) prior to sending this command.
- Alternatively, you may also use the **`MEASure:COUNter:DATA?`** query to return a frequency reading if the specified counter channel function is already set to measure frequency.

See also

"MEASure:COUNter:DATA?" on page 64

"[SENSe:]COUNter:FUNCtion" on page 127

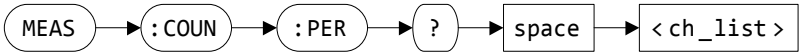
Example

This example illustrates how the `MEASure:COUNter:FREQuency?` query is used to return the frequency measurement for channel 301.

→ `COUN:FUNC?(@301)`	*This query returns the current counter function for channel 301.*
← `PER`	*The returned value "PER" indicates that channel 301 is currently set to measure the period.*
→ `MEAS:COUN:FREQ? (@301)`	*This query returns the frequency for channel 301. This query also changes the counter function.*
← `5.599627`	*The frequency is returned in kHz.*
→ `COUN:FUNC?(@301)`	
← `FREQ`	*The returned value "FREQ" indicates that channel 301 is now set to measure the frequency.*

MEASure:COUNter:PERiod?

Syntax



MEASure:COUNter:PERiod? <ch_list>

This query returns a numeric value that represents the period of each counter channel specified. Multiple responses are separated by commas. The value returned is expressed in milliseconds (ms).

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	(@301)|(@302)	-

Remarks

- Executing the **MEASure:COUNter:PERiod?** query will automatically change the current function of the counter channel specified. You do not need to select the period measurement function (via the **SENSe:COUNter:FUNCtion PERiod** command) prior to sending this command.
- Alternatively, you may also use the **MEASure:COUNter:DATA?** query to return a period reading if the specified counter channel function is already set to measure the period.

See also

"MEASure:COUNter:DATA?" on page 64

"[SENSe:]COUNter:FUNCtion" on page 127

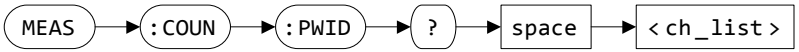
Example

This example illustrates how the **`MEASure:COUNter:PERiod?`** query is used to return the period measurement for channel 301.

→ `MEAS:COUN:PER? (@301)`	*This query returns the period for channel 301.*
← `0.1818333`	*The period is returned in ms.*

MEASure:COUNter:PWIDth?

Syntax



`MEASure:COUNter:PWIDth? <ch_list>`

This query returns a numeric value that represents the pulse width of each counter channel specified. Multiple responses are separated by commas. The value returned is expressed in milliseconds (ms).

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	(@301)|(@302)	-

Remarks

- Executing the **`MEASure:COUNter:PWIDth?`** query will automatically change the current function of the counter channel specified. You do not need to select the pulse width measurement function (via the **`SENSe:COUNter:FUNCtion PWIDth`** command) prior to sending this command.
- Alternatively, you may also use the **`MEASure:COUNter:DATA?`** query to return a pulse width reading if the specified counter channel function is already set to measure the pulse width.

See also

"MEASure:COUNter:DATA?" on page 64

"[SENSe:]COUNter:FUNCtion" on page 127

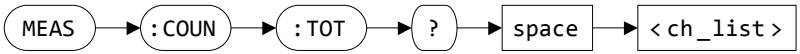
Example

This example illustrates how the **`MEASure:COUNter:PWIDth?`** query is used to return the pulse width for channel 301.

→ `MEAS:COUN:PWID? (@301)`	*This query returns the pulse width for channel 301.*
← `0.12725`	*The pulse width is returned in ms.*

MEASure:COUNter:TOTalize?

Syntax



MEASure:COUNter:TOTalize? <ch_list>

This query returns a numeric value that represents the totalized value of each counter channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	(@301)|(@302)	-

Remarks

- Executing the **MEASure:COUNter:TOTalize?** query will automatically change the current function of the counter channel specified. You do not need to select the totalize measurement function (via the **SENSe:COUNter:FUNCtion TOTalize** command) prior to sending this command.
- Alternatively, you may also use the **MEASure:COUNter:DATA?** query to return a totalized reading if the specified counter channel function is already set to the totalizer measurement function.

See also

"MEASure:COUNter:DATA?" on page 64

"[SENSe:]COUNter:FUNCtion" on page 127

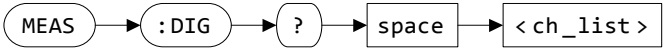
Example

This example illustrates how the `MEASure:COUNter:TOTalize?` query is used to return the totalized value for channel 301.

→ `*CLS; *RST`	*These commands clears and resets the instrument to its default settings.*
→ `COUN:TOT:INIT (@301)`	*This command starts the totalizer measurement on channel 301 with the instrument default values.*
→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301. The input signals are now allowed to pass.*
→ `MEAS:COUN:TOT? (@301)`	*This query returns the totalized value of the input signals for channel 301.*
← `56001`	*The totalized value thus far is returned.*

MEASure:DIGital?

Syntax



MEASure:DIGital? <ch_list>

This query returns a numeric value that represents the digital data of each DIO channel specified. Multiple responses are separated by commas. The specified DIO channels have to be configured as **INPut**, prior to returning any digital data.

NOTE To configure the DIO channel for input operation, see "CONFigure:DIGital:DIRection" on page 52.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	(@501)|(@502)|(@503)|(@504)	-

Remarks

- Channel 501 and 502 returns a value from 0 to 255 (8-bits).
- Channel 503 and 504 returns a value from 0 to 15 (4-bits).

See also

"CONFigure:DIGital:DIRection" on page 52

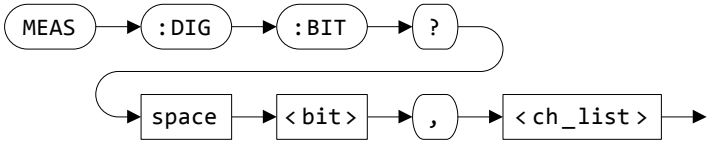
Example

This example illustrates how the **`MEASure:DIGital?`** query is used to return the digital data for DIO channel 501.

→ `CONF:DIG:DIR INP,(@501)`	*This command enables input operations for channel 501.*
→ `MEAS:DIG? (@501)`	*This query returns the digital data for channel 501.*
← `2`	*The returned value “2” indicates that bit 2 of channel 501 is at logic high.*

MEASure:DIGital:BIT?

Syntax



`MEASure:DIGital:BIT? <bit>, <ch_list>`

This query returns a boolean value that represents the digital logic state of the specified bit of each DIO channel specified. Multiple responses are separated by commas. The specified DIO channels have to be configured as `INPut`, prior to returning any digital data.

NOTE To configure the DIO channel for input operation, see "CONFigure:DIGital:DIRection" on page 52.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
bit	Numeric	– Channel 501 and 502: Bit 0 to 7 – Channel 503 and 504: Bit 0 to 3	-
ch_list	Channel list	(@501)|(@502)|(@503)|(@504)	-

Remarks

Channel 501 and 502 are of 8-bit length, while channel 503 and 504 are of 4-bit length.

See also

"CONFigure:DIGital:DIRection" on page 52

"MEASure:DIGital?" on page 74

Example

This example illustrates how the **`MEASure:DIGital:BIT?`** query is used to return the digital logic state of bit 2 for DIO channel 501.

→ `CONF:DIG:DIR INP,(@501)`	*This command enables input operations for channel 501.*
→ `MEAS:DIG? (@501)`	*This query returns the digital data for channel 501.*
← `2`	*The returned value "2" indicates that bit 2 of channel 501 is at logic high.*
→ `MEAS:DIG:BIT? 2,(@501)`	*This query returns the digital logic state of bit 2 for channel 501.*

← `1`

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

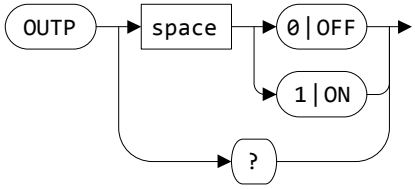
7 OUTPut Subsystem

OUTPut 80
OUTPut:WAVeform:ITERate 82
OUTPut:WAVeform:SRATe 84
OUTPut:WAVeform:FREQuency 87
OUTPut:TRIGger:SOURce 89
OUTPut:TRIGger:TYPe 91
OUTPut:TRIGger:DCouNT 93
OUTPut:TRIGger:ATRiGger:SOURce 95
OUTPut:TRIGger:ATRiGger:CONDition 97
OUTPut:TRIGger:ATRiGger:HTHReshold 100
OUTPut:TRIGger:ATRiGger:LTHReshold 102
OUTPut:TRIGger:DTRiGger:POLarity 104

This chapter describes the OUTPut commands used to program the U2500A Series DAQ device over a remote interface. Use the OUTPut command to enable or disable the analog output. You can also set the AO trigger settings.

OUTPut

Syntax



`OUTPut <mode>`

This command is used to enable or disable the analog output.

`OUTPut?`

This query returns a boolean value that represents the status of the analog output.

- "`0`" indicates that the output is disabled, and
- "`1`" indicates that the output is enabled.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
mode	Boolean	0|OFF|1|ON	0

Remarks

- The `OUTPut ON` and `OUTPut OFF` commands cannot be executed repeatedly.
- You cannot generate a predefined and a user-defined waveform at the same time.

See also

"OUTPut:WAVeform:ITERate" on page 82

"OUTPut:WAVeform:SRATe" on page 84

"OUTPut:WAVeform:FREQuency" on page 87

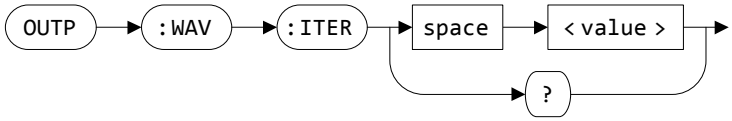
Example

This example illustrates how the **OUTPut** command is used to generate a 5 V sinusoidal wave with no DC offset from AO channel 201.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:WAV:ITER 0`	*This command causes the instrument to generate the waveform indefinitely.*
→ `OUTP:WAV:SRAT 0`	*This command automatically selects a suitable sampling rate for the output waveform.*
→ `OUTP:WAV:FREQ 5000`	*This command sets the output waveform frequency to 5 kHz.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201.*

OUTPut:WAVeform:ITERate

Syntax



`OUTPut:WAVeform:ITERate <value>`

This command is used to set the number of waveform iterations in the AO buffer.

`OUTPut:WAVeform:ITERate?`

This query returns a numeric value that represents the waveform iteration count.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	0 to 16777215 (24-bits)[a]	0 (Infinite iteration)

[a] The iteration parameter is subjected to the following constraints:
Iteration ≥ 0
Iteration < 0xffffff

Remarks

Default value zero causes the DAQ device to generate the waveform indefinitely.

See also

"OUTPut" on page 80

"OUTPut:WAVeform:SRATe" on page 84

"OUTPut:WAVeform:FREQuency" on page 87

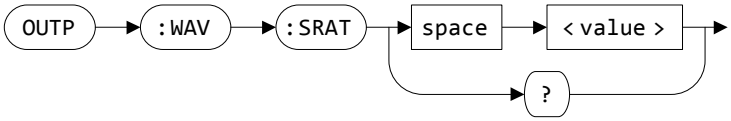
Example

This example illustrates how the `OUTPut:WAVeform:ITERation` command is used to generate 256 iterations of a 5 V sinusoidal wave with no DC offset for AO channel 201.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:WAV:ITER 256`	*This command sets an iteration value of 256 for the output waveform. This implies that channel 201 will generate a full sinusoidal wave 256 times when the OUTP ON command is issued.*
→ `OUTP:WAV:SRAT 0`	*This command automatically selects a suitable sampling rate for the output waveform.*
→ `OUTP:WAV:FREQ 5000`	*This command sets the output waveform frequency to 5 kHz.*
→ `OUTP ON`	*This command generates the sinusoidal waveform for 256 iterations from channel 201.*

OUTPut:WAVeform:SRATe

Syntax



`OUTPut:WAVeform:SRATe <value>`

This command is used to set the waveform sampling rate of the analog output.

`OUTPut:WAVeform:SRATe?`

This query returns a numeric value that represents the waveform sampling rate. The value returned is expressed in hertz (Hz).

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	(48/16777215) Hz to 1000000 Hz (1 MHz)[a]	0 (Automatic sampling)

[a] The sampling rate parameter is subjected to the following constraints:
$0 \leq$ Sampling rate $\leq 10^6$
Sampling rate $\geq (48 \times 10^6 / 0xffffff)$

Remarks

- Default value zero causes the DAQ device to automatically select a suitable waveform sampling rate for predefined waveform (`SINusoid`, `SQUare`, `SAWTooth`, `TRIangle`, and `NOISe`) generation.
- The U2500A Series DAQ device can output a maximum of 4096 points for a single waveform.
- The relationship between the waveform sampling rate, waveform frequency, and number of points in a waveform is defined by:

$$\textit{Waveform points} = \frac{\textit{Waveform sampling rate}}{\textit{Waveform frequency}} \quad (1)$$

NOTE

- **Use the `OUTPut:WAVeform:FREQuency` command to set the waveform frequency.**
- **Using a lower waveform sampling rate will increase the duration taken to generate a waveform. You are encouraged to use the default automatic sampling configuration by sending the `OUTPut:WAVeform:SRATe 0` command for predefined waveform generation.**

See also

"OUTPut" on page 80

"OUTPut:WAVeform:ITERate" on page 82

"OUTPut:WAVeform:FREQuency" on page 87

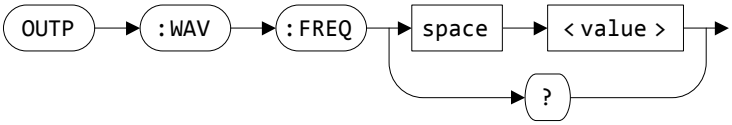
Example

This example illustrates how the `OUTPut:WAVeform:SRATe` command is used to generate a 5 V sinusoidal wave with no DC offset for AO channel 201 with a sampling rate of 500 kHz.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:WAV:ITER 0`	*This command causes the instrument to generate the waveform indefinitely.*
→ `OUTP:WAV:SRAT 500000`	*This command selects an 500 kHz sampling rate for the output waveform. This implies that the sinusoidal waveform is generated with 100 points (500 kHz / 5 kHz) for each cycle.*
→ `OUTP:WAV:FREQ 5000`	*This command sets the output waveform frequency to 5 kHz.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201.*

OUTPut:WAVeform:FREQuency

Syntax



`OUTPut:WAVeform:FREQuency <value>`

This command is used to set the analog output frequency.

`OUTPut:WAVeform:FREQuency?`

This query returns a numeric value that represents the frequency value for the analog outputs. The returned value is expressed in hertz (Hz).

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	10 Hz to 10000 Hz (10 kHz)	4000 Hz (4 kHz)

See also

"OUTPut" on page 80

"OUTPut:WAVeform:ITERate" on page 82

"OUTPut:WAVeform:SRATe" on page 84

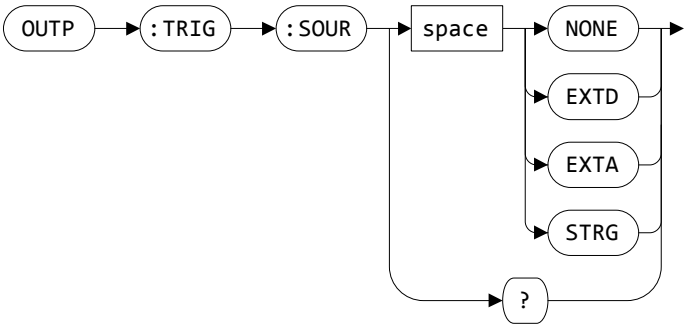
Example

This example illustrates how the `OUTPut:WAVeform:FREQuency` command is used to generate a 10 kHz 5 V sinusoidal wave with no DC offset for AO channel 201.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:WAV:ITER 0`	*This command causes the instrument to generate the waveform indefinitely.*
→ `OUTP:WAV:SRAT 0`	*This command automatically selects a suitable sampling rate for the output waveform.*
→ `OUTP:WAV:FREQ 10000`	*This command sets the output waveform frequency to 10 kHz.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201.*

OUTPut:TRIGger:SOURce

Syntax



`OUTPut:TRIGger:SOURce <mode>`

This command is used to set the trigger source for output operations.

The valid options are:

- NONE: Immediate triggering.
- EXTD: Selects the external digital trigger (EXTD_AO_TRIG) pin as the triggering source.
- EXTA: Selects the external analog trigger (EXTA_TRIG) pin as the triggering source.
- STRG: Star triggering.

NOTE

- **Star triggering (STRG) is used when the DAQ device is connected into the U2781A USB modular instrument chassis.**
- **For more information on selecting a trigger source, see “Trigger Sources” in the *Keysight U2500A Series USB Simultaneous Sampling Multifunction Data Acquisition User’s Guide*.**

`OUTPut:TRIGger:SOURce?`

This query returns a string value that represents the trigger source selected for output operations.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
mode	Discrete	NONE|EXTD|EXTA|STRG	NONE

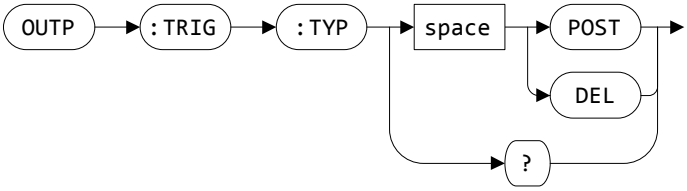
Example

This example illustrates how the **`OUTPut:TRIGger:SOURce`** command is used to select the EXTD_AO_TRIG pin as the output trigger source.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:TRIG:SOUR EXTD`	*This command sets the EXTD_AO_TRIG pin as the output triggering source.*
→ `OUTP:TRIG:DTRG:POL POS`	*This command sets a positive-edge trigger condition for the digital trigger source.*
→ `OUTP:TRIG:TYP POST`	*This command sets a post-trigger acquisition for the analog output.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201 when a positive-edge digital signal is received from the EXTD_AO_TRIG pin.*

OUTPut:TRIGger:TYPe

Syntax



`OUTPut:TRIGger:TYPe <mode>`

This command is used to set the trigger type for output operations.

The valid options are:

- POST (Post-trigger): The instrument output is generated immediately after the trigger condition is met.
- DEL (Delay-trigger): The instrument output is generated when the delay count reaches zero. The delay count starts immediately after the trigger condition is met.

NOTE

- See "OUTPut:TRIGger:DCouNT" on page 93 for more information on how to set a delay counter value.
- For more information on selecting a trigger type, see "Trigger types" in the *Keysight U2500A Series USB Simultaneous Sampling Multifunction Data Acquisition User's Guide*.

`OUTPut:TRIGger:TYPe?`

This query returns a string value that represents the trigger type selected for output operations.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
mode	Discrete	POST|DEL	POST

See also

"OUTPut:TRIGger:DCouNT" on page 93

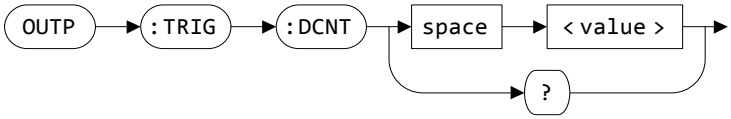
Example

This example illustrates how the **`OUTPut:TRIGger:TYPe`** command is used to select a delay-trigger acquisition as the output trigger type.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:TRIG:SOUR EXTD`	*This command sets the EXTD_AO_TRIG pin as the output triggering source.*
→ `OUTP:TRIG:DTRG:POL POS`	*This command sets a positive-edge trigger condition for the digital trigger source.*
→ `OUTP:TRIG:TYP DEL`	*This command sets a delay-trigger acquisition for the analog output.*
→ `OUTP:TRIG:DCNT 240000000`	*This command sets the delay-trigger acquisition counter value.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201 once the delay count reaches zero. The delay count starts when a positive-edge digital signal is received from the EXTD_AO_TRIG pin.*

OUTPut:TRIGger:DCouNT

Syntax



`OUTPut:TRIGger:DCouNT <value>`

This command is used to set the delay count value. When the count reaches zero, the counter stops and the DAQ device starts generating the instrument output. The count set in the `<value>` parameter will be used when the output trigger type is set to `DEL` (delay-trigger). Set the clock source using the `CONFigure:TIMEbase:SOURce` command.

NOTE

- See "OUTPut:TRIGger:TYPe" on page 91 for more information on selecting an output trigger type.
- See "CONFigure:TIMEbase:SOURce" on page 54 for more information on selecting a timebase/clock source.

`OUTPut:TRIGger:DCouNT?`

This query returns a numeric value that represents the delay count value.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	0 to 2147483647 (31-bits)[a]	0

[a] The delay count parameter is subjected to the following constraints:
Delay count ≥ 0
Delay count < 0x7fffffff

See also

"CONFigure:TIMEbase:SOURce" on page 54

"OUTPut:TRIGger:TYPe" on page 91

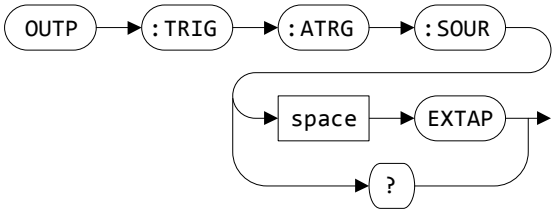
Example

This example illustrates how the **`OUTPut:TRIGger:DCouNT`** command is used to set a delay counter value of 240000000. This will set a delay time of 5 seconds (240000000/48 MHz).

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:TRIG:SOUR EXTD`	*This command sets the EXTD_AO_TRIG pin as the output triggering source.*
→ `OUTP:TRIG:DTRG:POL POS`	*This command sets a positive-edge trigger condition for the digital trigger source.*
→ `OUTP:TRIG:TYP DEL`	*This command sets a delay-trigger acquisition for the analog output.*
→ `OUTP:TRIG:DCNT 240000000`	*This command sets the delay-trigger acquisition counter value. This implies that channel 201 will generate the sinusoidal waveform 5 seconds (240000000 / 48 MHz) after the trigger condition occurs.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201 5 seconds after a positive-edge digital signal is received from the EXTD_AO_TRIG pin.*

OUTPut:TRIGger:ATRiGger:SOURce

Syntax



`OUTPut:TRIGger:ATRiGger:SOURce <mode>`

This command is used to set the analog trigger source for the AO trigger control.

EXTAP: Selects the external analog trigger (EXTA_TRIG) pin as the analog triggering source.

NOTE

- This is an adjunct command for the `TRIGger:SOURce EXTA` command.
- Use the ROUTe:SCAN command to determine the first channel in the instrument scan list.

`OUTPut:TRIGger:ATRiGger:SOURce?`

This query returns a string value that represents the analog trigger source selected for the AO trigger control.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
mode	Discrete	EXTAP|CH101|CH102|CH103|CH104	EXTAP

NOTE

Only post-trigger and pre-trigger modes can be used if trigger source is set to CH101|CH102|CH103|CH104.

See also

"OUTPut:TRIGger:SOURce" on page 89
"OUTPut:TRIGger:ATRiGger:CONDition" on page 97
"OUTPut:TRIGger:ATRiGger:HTHReshold" on page 100
"OUTPut:TRIGger:ATRiGger:LTHReshold" on page 102

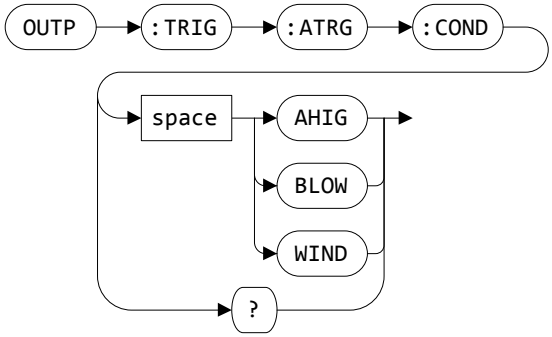
Example

This example illustrates how the **`OUTPut:TRIGger:ATRiGger:SOURce`** command is used to select the EXTA_TRIG pin of the DAQ device as the analog output trigger source.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:TRIG:SOUR EXTA`	*This command sets the external analog trigger as the output triggering source.*
→ `OUTP:TRIG:ATRG:SOUR EXTAP`	*This command sets the EXTA_TRIG pin as the analog trigger control source.*
→ `OUTP:TRIG:ATRG:COND BLOW`	*This command sets a below-low-level triggering condition.*
→ `OUTP:TRIG:ATRG:LTHR 5`	*This command sets a 5 V low-threshold.*
→ `OUTP:TRIG:TYP POST`	*This command set a post-trigger acquisition for the analog output.*
→ `RUN`	*This command starts the continuous data acquisition process.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201 when all trigger conditions are met. If the input signal from the EXTA_TRIG pin falls below 5 V, the analog output will be triggered.*

OUTPut:TRIGger:ATRiGger:CONDition

Syntax



`OUTPut:TRIGger:ATRiGger:CONDition <mode>`

This command is used to set the trigger condition for the AO trigger control.

The valid options are:

- AHIG (Above-High-Level triggering): The trigger signal is generated when the analog signal is higher than the high-threshold voltage.
- BLOW (Below-Low-Level triggering): The trigger signal is generated when the analog signal is lower than the low-threshold voltage.
- WIND (Window (inside region) triggering): The trigger signal is generated when the analog signal falls within the range of the high-threshold and low-threshold voltages.

NOTE Use the `OUTPut:TRIGger:ATRiGger:LTHReshold` and `OUTPut:TRIGger:ATRiGger:HTHReshold` commands to set the low-threshold and high-threshold voltages.

`OUTPut:TRIGger:ATRiGger:CONDition?`

This query returns a string value that represents the trigger condition for the AO trigger control.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
mode	Discrete	AHIG|BLOW|WIND	BLOW

See also

"OUTPut:TRIGger:SOURce" on page 89
"OUTPut:TRIGger:ATRiGger:SOURce" on page 95
"OUTPut:TRIGger:ATRiGger:HTHReshold" on page 100
"OUTPut:TRIGger:ATRiGger:LTHReshold" on page 102

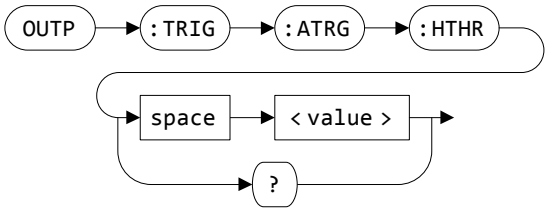
Example

This example illustrates how the `OUTPut:TRIGger:ATRiGger:CONDition` command is used to select a window triggering condition for the AO trigger control.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:TRIG:SOUR EXTA`	*This command sets the external analog trigger as the output triggering source.*
→ `OUTP:TRIG:ATRG:SOUR EXTAP`	*This command sets the EXTA_TRIG pin as the analog trigger control source.*
→ `OUTP:TRIG:ATRG:COND WIND`	*This command sets a window triggering condition.*
→ `OUTP:TRIG:ATRG:HTHR 5`	*This command sets a 5 V high-threshold.*
→ `OUTP:TRIG:ATRG:LTHR 1`	*This command sets a 1 V low-threshold.*
→ `OUTP:TRIG:TYP POST`	*This command set a post-trigger acquisition for the analog output.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201 when all trigger conditions are met. If the input signal from the EXTA_TRIG pin falls between 5 V and 1 V, the analog output will be triggered.*

OUTPut:TRIGger:ATRiGger:HTHReshold

Syntax



`OUTPut:TRIGger:ATRiGger:HTHReshold <value>`

This command sets the high-threshold voltage of the AO trigger control.

`OUTPut:TRIGger:ATRiGger:HTHReshold?`

This query returns a numeric value that represents the high-threshold voltage for the AO trigger control.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	–10 V to 10 V	0 V

Remarks

The high-threshold voltage value must be larger than the low-threshold voltage value.

See also

"OUTPut:TRIGger:SOURce" on page 89

"OUTPut:TRIGger:ATRiGger:SOURce" on page 95

"OUTPut:TRIGger:ATRiGger:CONDition" on page 97

"OUTPut:TRIGger:ATRiGger:LTHReshold" on page 102

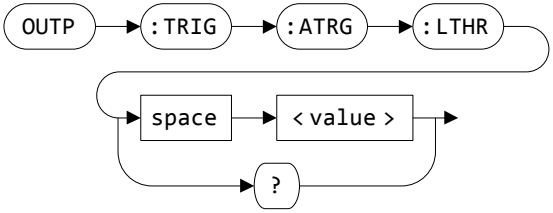
Example

This example illustrates how the `OUTPut:TRIGger:ATRiGger:HTHReshold` command is used to set a 5 V high-threshold for the AO trigger control.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:TRIG:SOUR EXTA`	*This command sets the external analog trigger as the output triggering source.*
→ `OUTP:TRIG:ATRG:SOUR EXTAP`	*This command sets the EXTA_TRIG pin as the analog trigger control source.*
→ `OUTP:TRIG:ATRG:COND AHIG`	*This command sets a above-high-level triggering condition.*
→ `OUTP:TRIG:ATRG:HTHR 5`	*This command sets a 5 V high-threshold.*
→ `OUTP:TRIG:TYP POST`	*This command set a post-trigger acquisition for the analog output.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201 when all trigger conditions are met. If the input signal rises above 5 V, the analog output will be triggered.*

OUTPut:TRIGger:ATRiGger:LTHReshold

Syntax



`OUTPut:TRIGger:ATRiGger:LTHReshold <value>`

This command is used to set the low-threshold voltage of the AO trigger control.

`OUTPut:TRIGger:ATRiGger:LTHReshold?`

This query returns a numeric value that represents the low-threshold voltage for the AO trigger control.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	–10 V to 10 V	0 V

Remarks

The high-threshold voltage value must be larger than the low-threshold voltage value.

See also

"OUTPut:TRIGger:SOURce" on page 89

"OUTPut:TRIGger:ATRiGger:SOURce" on page 95

"OUTPut:TRIGger:ATRiGger:CONDition" on page 97

"OUTPut:TRIGger:ATRiGger:HTHReshold" on page 100

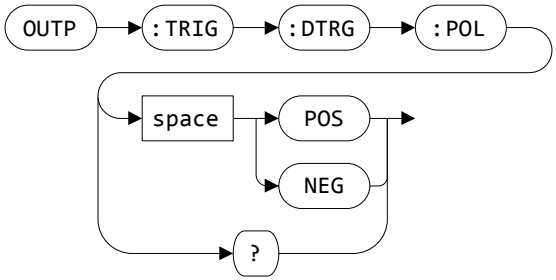
Example

This example illustrates how the `OUTPut:TRIGger:ATRiGger:LTHReshold` command is used to set a 1 V low-threshold for the AO trigger control.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:TRIG:SOUR EXTA`	*This command sets the external analog trigger as the output triggering source.*
→ `OUTP:TRIG:ATRG:SOUR EXTAP`	*This command sets the EXTA_TRIG pin as the analog trigger control source.*
→ `OUTP:TRIG:ATRG:COND BLOW`	*This command sets a below-low-level triggering condition.*
→ `OUTP:TRIG:ATRG:LTHR 1`	*This command sets a 1 V low-threshold.*
→ `OUTP:TRIG:TYP POST`	*This command set a post-trigger acquisition for the analog output.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201 when all trigger conditions are met. If the input signal falls below 1 V, the analog output will be triggered.*

OUTPut:TRIGger:DTRiGger:POLarity

Syntax



`OUTPut:TRIGger:DTRiGger:POLarity <mode>`

This command is used to set the polarity of the DIO trigger control.

The valid options are:

- POS (Positive-edge triggering): The trigger signal is generated when a rising edge is detected in the digital signal.
- NEG (Negative-edge triggering): The trigger signal is generated when a falling edge is detected in the digital signal.

`OUTPut:TRIGger:DTRiGger:POLarity?`

This query returns a string value that represents the polarity selected for the DIO trigger control.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
mode	Discrete	POS|NEG	POS

See also

"OUTPut:TRIGger:SOURce" on page 89

"OUTPut:TRIGger:TYPe" on page 91

Example

This example illustrates how the `OUTPut:TRIGger:DTRiGger:POLarity` command is used to select a negative-edge trigger condition for the DIO trigger control.

→ `ROUT:ENAB ON,(@201)`	*This command enables channel 201 for output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform to channel 201 with an amplitude of 5 V (10 V_{pp}) and a DC offset of 0 V.*
→ `OUTP:TRIG:SOUR EXTD`	*This command sets the EXTD_AO_TRIG pin as the output triggering source.*
→ `OUTP:TRIG:DTRG:POL NEG`	*This command sets a negative-edge trigger condition for the digital trigger source.*
→ `OUTP:TRIG:TYP POST`	*This command set a post-trigger acquisition for the analog output.*
→ `OUTP ON`	*This command generates the sinusoidal waveform from channel 201 when a negative-edge digital signal is received from the EXTD_AO_TRIG pin.*

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

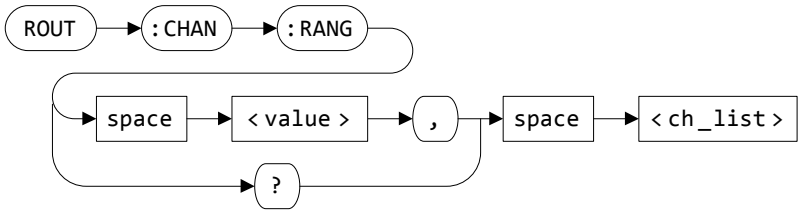
8 ROUTe Subsystem

ROUTe:CHANnel:RANGe 108
ROUTe:CHANnel:POLarity 110
ROUTe:CHANnel:STYPe? 112
ROUTe:CHANnel:RSouRCe 113
ROUTe:CHANnel:RVOLtage 115
ROUTe:ENABle 117

This chapter describes the ROUTe commands used to program the U2500A Series DAQ device over a remote interface. Use the ROUTe:ENABle command to enable or disable the signal route for the desired AI and AO channel(s). You can also configure the AI/AO channels range, polarity, input signal type, and reference voltage source.

ROUTe:CHANnel:RANGe

Syntax



`ROUTe:CHANnel:RANGe <value>, <ch_list>`

This command is used to set the range for the AI channel(s) specified in the `<ch_list>` parameter.

`ROUTe:CHANnel:RANGe? <ch_list>`

This query returns a numeric value that represents the range of each AI channel specified. Multiple responses are separated by commas. The value returned is expressed in volts (V).

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
value	Discrete	Unipolar/Bipolar mode: {10|5|2.5|1.25}	10 V
ch_list	Channel list	(@101) to (@104)	-

Remarks

If the AI channel polarity is set to unipolar, setting a value of 10 in the `<value>` parameter will result in a range from 0 V to +10 V. Whereas, if the AI channel polarity is set to bipolar, setting a value of 10 in the `<value>` parameter will result in a range from –10 V to +10 V.

NOTE For more information on selecting the AI channel polarity, see "ROUTe:CHANnel:POLarity" on page 110.

See also

"ROUTe:CHANnel:POLarity" on page 110

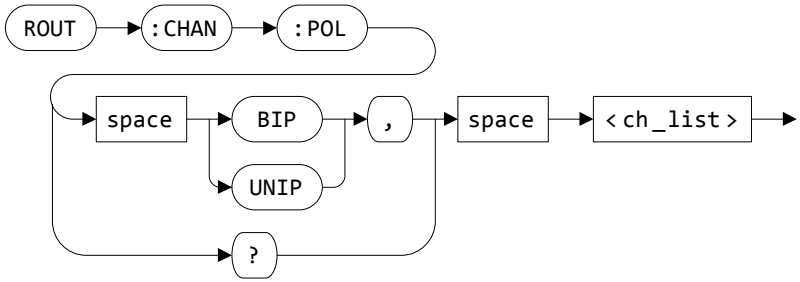
Example

This example illustrates how the **ROUTe:CHANnel:RANGe** command is used to specify the range of AI channel 101.

→ `ROUT:CHAN:POL BIP,(@101)`	*This command sets the polarity of channel 101 to bipolar.*
→ `ROUT:CHAN:RANG 5,(@101)`	*This command sets channel 101 to expect input signals from a range of –5 V to +5 V.*
→ `ROUT:CHAN:POL UNIP,(@101)`	*This command sets the polarity of channel 101 to unipolar.*
→ `ROUT:CHAN:RANG 5,(@101)`	*This command sets channel 101 to expect input signals from a range of 0 V to +5 V.*

ROUTe:CHANnel:POLarity

Syntax



`ROUTe:CHANnel:POLarity <mode>, <ch_list>`

This command is used to set the polarity of the AI or AO channel(s) specified in the `<ch_list>` parameter.

The valid options are:

- BIPolar
- UNIPolar

`ROUTe:CHANnel:POLarity? <ch_list>`

This query returns a string value that represents the polarity of each AI or AO channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	UNIPolar|BIPolar	BIPolar
ch_list	Channel list	(@101) to (@104)	-

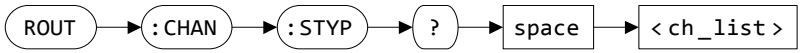
Example

This example illustrates how the **ROUTe:CHANnel:POLarity** command is used to change the polarity of AI channel 101.

→ `ROUT:CHAN:POL?(@101:103)`	*This query returns the polarity of channel 101 to 103.*
← `BIP,BIP,BIP`	*The returned value "BIP" for all channels indicates the instrument default polarity – bipolar.*
→ `ROUT:CHAN:POL UNIP,(@101)`	*This command changes the polarity of channel 101 to unipolar.*
→ `ROUT:CHAN:POL?(@101:103)`	
← `UNIP,BIP,BIP`	*The returned value "UNIP" for channel 101 indicates a change in the polarity of channel 101.*

ROUTe:CHANnel:STYPe?

Syntax



`ROUTe:CHANnel:STYPe? <ch_list>`

This query returns a string value that represents the input signal type of each AI channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
ch_list	Channel list	(@101) to (@104)	-

Example

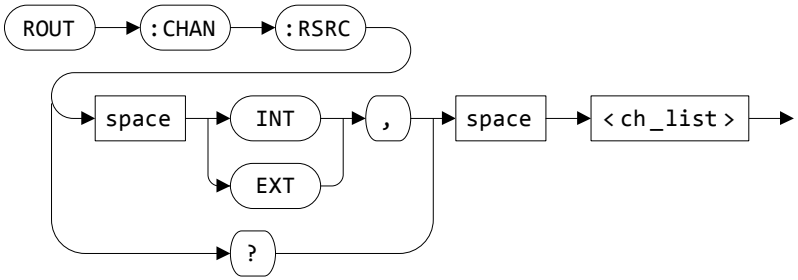
This example illustrates how the **`ROUTe:CHANnel:STYPe?`** query is used to return the input signal type of AI channel 101.

→ `ROUT:CHAN:STYP? (@101)`	*This query returns the input signal type for channel 101.*

← `DIFF`

ROUTe:CHANnel:RSouRCe

Syntax



`ROUTe:CHANnel:RSouRCe <mode>, <ch_list>`

This command is used to set the reference voltage source for the AO channel(s) specified in the `<ch_list>` parameter.

The valid options are:

- INTernal: Internal on-board reference voltage.
- EXTernal: Selects the AO_EXT_REF pin as the reference voltage source.

NOTE

- **The reference voltage coming in from the AO_EXT_REF pin must be set in the DAQ device via the `ROUTe:CHANnel:RVOLtage` command. For example, if a 5 V reference voltage is expected from the AO_EXT_REF pin, you must send the `ROUTe:CHANnel:RVOLtage 5` command to the DAQ device.**
- **See "ROUTe:CHANnel:RVOLtage" on page 115 for more information on how to set the external reference voltage value.**

`ROUTe:CHANnel:RSouRCe? <ch_list>`

This query returns a string value that represents the reference voltage source of each AO channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	INTernal|EXTernal	INTernal
ch_list	Channel list	(@101) to (@104)	-

See also

"ROUTe:CHANnel:RVOLtage" on page 115

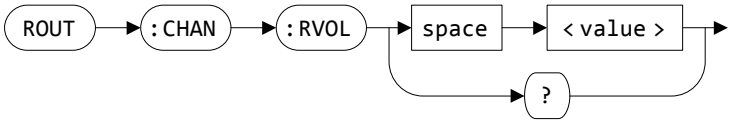
Example

This example illustrates how the `ROUTe:CHANnel:RSouRCe` command is used to change the AO channels reference voltage source.

→ `ROUT:CHAN:RSRC EXT`	*This command sets the instrument reference voltage source to the AO_EXT_REF pin. A 5 V voltage source is then supplied to the AO_EXT_REF pin.*
→ `ROUT:CHAN:RVOL 5`	*Since we are expecting a 5 V voltage source from the AO_EXT_REF pin, this command is used to set the external reference voltage to 5 V.*
→ `ROUT:CHAN:RSRC?`	*This query returns the reference voltage source for the AO channels.*
← `EXT`	

ROUTe:CHANnel:RVOLtage

Syntax



`ROUTe:CHANnel:RVOLtage <value>`

This command is used to set the external reference voltage value for the AO channels. The reference voltage set in the `<value>` parameter must be the same as the voltage value supplied to the AO_EXT_REF pin when the reference voltage source is set to `EXTernal`. The reference voltage value is expressed in units of volts (V).

NOTE See "ROUTe:CHANnel:RSouRCe" on page 113 for more information on selecting a reference voltage source.

`ROUTe:CHANnel:RVOLtage?`

This query returns a string value that represents the external reference voltage value. The value returned is expressed in volts (V).

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	0 < value ≤ 10	10 V

Remarks

The reference voltage value cannot be set to zero or a negative value.

See also

"ROUTe:CHANnel:RSouRCe" on page 113

Example

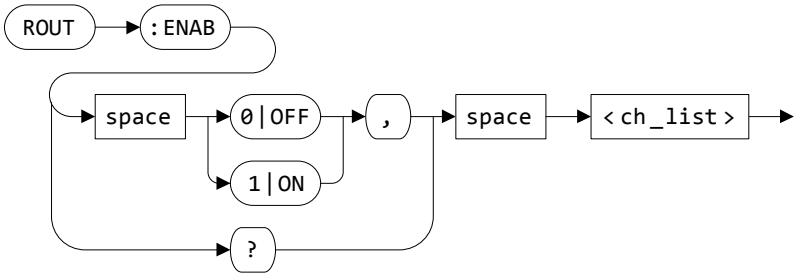
This example illustrates how the **ROUTe:CHANnel:RVOLtage** command is used to set the external reference voltage value.

→ `ROUT:CHAN:RSRC EXT`	*This command sets the instrument reference voltage source to the AO_EXT_REF pin. A 5 V voltage source is then supplied to the AO_EXT_REF pin.*
→ `ROUT:CHAN:RVOL 5`	*Since we are expecting a 5 V voltage source from the AO_EXT_REF pin, this command is used to set the external reference voltage to 5 V.*
→ `ROUT:CHAN:RVOL?`	*This query returns the external reference voltage value for the AO channels.*

← `5`

ROUTe:ENABle

Syntax



`ROUTe:ENABle <mode>, <ch_list>`

This command is used to enable or disable the signal route for the AI or AO channel(s) specified in the `<ch_list>` parameter. By enabling a channel, electrical signals are allowed to pass through that enabled channel. Likewise, by disabling a particular channel, electrical signals are not allowed to pass through the said channel.

`ROUTe:ENABle? <@ch_list>`

This query returns a boolean value that represents the status of each AI or AO channel specified. Multiple responses are separated by commas.

- "`0`" indicates that the specific channel is disabled, and
- "`1`" indicates that the specific channel is enabled.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Boolean	0|OFF|1|ON	0
ch_list	Channel list	– AI: (@101)|(@102)|(@103)|(@104) – AO: (@201)|(@202)	-

Example

This example illustrates how the **`ROUTe:ENABle command`** is used to enable channel 201 and 202 for AO operations.

→ `ROUT:ENAB 1,(@201:202)`	*This command enables channel 201 and 202 for analog output operations.*
→ `APPL:SIN 5,0,(@201)`	*This command sends a sinusoidal waveform with 5 V (10 V_{pp}) and 0 V offset to the instrument buffer.*
→ `APPL:SQU 5,0,(@202)`	*This command sends a square waveform with 5 V (10 V_{pp}) and 0 V offset to the instrument buffer.*
→ `OUTP ON`	*This command outputs a sinusoidal waveform from channel 201 and a square waveform from channel 202.*

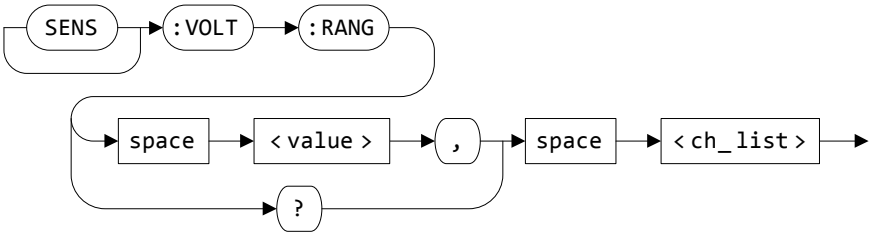
9 SENSe Subsystem

[SENSe:]VOLTage:RANGe 120
[SENSe:]VOLTage:POLarity 122
[SENSe:]VOLTage:STYPe? 124
[SENSe:]VOLTage:AVERage 125
[SENSe:]COUNter:FUNCtion 127
[SENSe:]COUNter:ABORt 129
[SENSe:]COUNter:GATE:SOURce 131
[SENSe:]COUNter:GATE:POLarity 133
[SENSe:]COUNter:GATE:CONTrol 135
[SENSe:]COUNter:CLocK:SOURce 137
[SENSe:]COUNter:CLocK:INTernal? 139
[SENSe:]COUNter:CLocK:EXTernal 140
[SENSe:]COUNter:CLocK:POLarity 142
[SENSe:]COUNter:TOTalize:IVALue 144
[SENSe:]COUNter:TOTalize:INITiate 146
[SENSe:]COUNter:TOTalize:CLEar 147
[SENSe:]COUNter:TOTalize:UDOWn:SOURce 149
[SENSe:]COUNter:TOTalize:UDOWn:DIRection 152

This chapter describes the SENSe commands used to program the U2500A Series DAQ device over a remote interface. Use the SENSe commands to change the AI channels settings prior to making a single voltage measurement. You can also select a counter function and configure its properties.

[SENSe:]VOLTage:RANGe

Syntax



`[SENSe:]VOLTage:RANGe <value>, <ch_list>`

This command is used to set the range for the AI channel(s) specified in the `<ch_list>` parameter. This command is applicable for polling mode operations.

`[SENSe:]VOLTage:RANGe? <ch_list>`

This query returns a numeric value that represents the voltage range of each AI channel specified. Multiple responses are separated by commas. The value returned is expressed in volts (V).

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
value	Discrete	Unipolar/Bipolar mode: {AUTO|10|5|2.5|1.25}	AUTO
ch_list	Channel list	(@101) to (@104)	-

Remarks

If the input voltage polarity is set to unipolar, setting a value of 10 in the `<value>` parameter will result in a range from 0 V to +10 V. Whereas, if the input voltage polarity is set to bipolar, setting a value of 10 in the `<value>` parameter will result in a range from –10 V to +10 V.

See also

"MEASure[:VOLTage][:DC]?" on page 62

"[SENSe:]VOLTage:POLarity" on page 122

"[SENSe:]VOLTage:STYPe?" on page 124

"[SENSe:]VOLTage:AVERage" on page 125

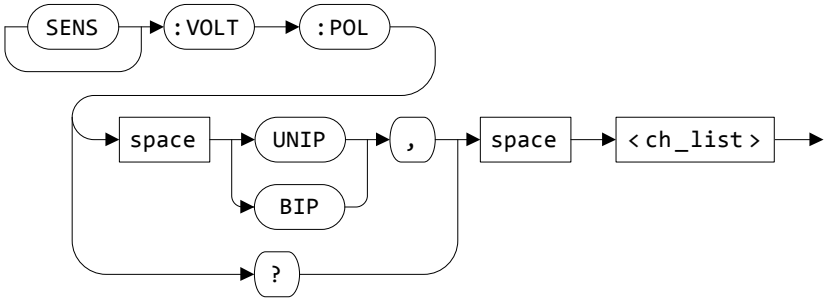
Example

This example illustrates how the **`MEASure[:VOLTage][:DC]?`** query is used to make a single voltage measurement from AI channel 101. The **`[SENSe:]VOLTage:RANGe`** command is used to change the input range of AI channel 101.

→ `VOLT:POL BIP,(@101)`	*This command changes the polarity of channel 101 to bipolar.*
→ `VOLT:RANG 5,(@101)`	*This command sets channel 101 to expect an input voltage within a range of –5 V to +5 V.*
→ `MEAS? (@101)`	*This query returns a single voltage measurement from channel 101.*
← `4.102`	

[SENSe:]VOLTage:POLarity

Syntax



`[SENSe:]VOLTage:POLarity <mode>, <ch_list>`

This command is used to set the polarity of the AI channel(s) specified in the `<ch_list>` parameter. This command is applicable for polling mode operations.

The valid options are:

- BIPolar
- UNIPolar

`[SENSe]:VOLTage:POLarity? <ch_list>`

This query returns a string value that represents the polarity of each AI channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	UNIPolar|BIPolar	BIPolar
ch_list	Channel list	(@101) to (@104)	-

See also

"MEASure[:VOLTage][:DC]?" on page 62

"[SENSe:]VOLTage:RANGe" on page 120

"[SENSe:]VOLTage:STYPe?" on page 124

"[SENSe:]VOLTage:AVERage" on page 125

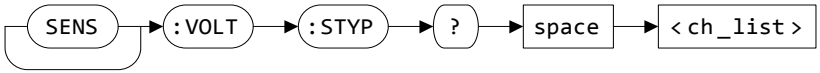
Example

This example illustrates how the **`MEASure[:VOLTage][:DC]?`** query is used to make a single voltage measurement from AI channel 101. The **`[SENSe:]VOLTage:POLarity`** command is used to change the polarity of AI channel 101.

→ `VOLT:POL UNIP,(@101)`	*This command changes the polarity of channel 101 to unipolar.*
→ `VOLT:RANG 5,(@101)`	*This command sets channel 101 to expect an input voltage within a range of 0 V to +5 V.*
→ `MEAS? (@101)`	*This query returns a single voltage measurement from channel 101.*
← `4.102`	

[SENSe:]VOLTage:STYPe?

Syntax



`[SENSe]:VOLTage:STYPe? <ch_list>`

This query returns a string value that represents the input signal type of each AI channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	(@101) to (@104)	-

See also

"MEASure[:VOLTage][:DC]?" on page 62

"[SENSe:]VOLTage:RANGe" on page 120

"[SENSe:]VOLTage:POLarity" on page 122

"[SENSe:]VOLTage:AVERage" on page 125

Example

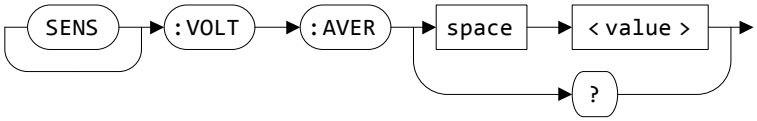
This example illustrates how the **`[SENSe:]VOLTage:STYPe?`** query is used to return the input signal type of AI channel 101.

→ `VOLT:STYP? (@101)`	*This query returns the input signal type for channel 101.*

← `DIFF`

[SENSe:]VOLTage:AVERage

Syntax



`[SENSe]:VOLTage:AVERage <value>`

This command is used to set the number of measurement points to average. The DAQ device averages a range of measurements (specified in the `<value>` parameter) before returning the average value. This command is applicable for polling mode operations.

`[SENSe]:VOLTage:AVERage?`

This query returns a numeric value that represents the number of measurement points to average.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	1 to 1000	1

Remarks

Once the average number is set, the effects are applicable across all AI channels for polling mode operations.

See also

"MEASure[:VOLTage][:DC]?" on page 62

"[SENSe:]VOLTage:RANGe" on page 120

"[SENSe:]VOLTage:POLarity" on page 122

"[SENSe:]VOLTage:STYPe?" on page 124

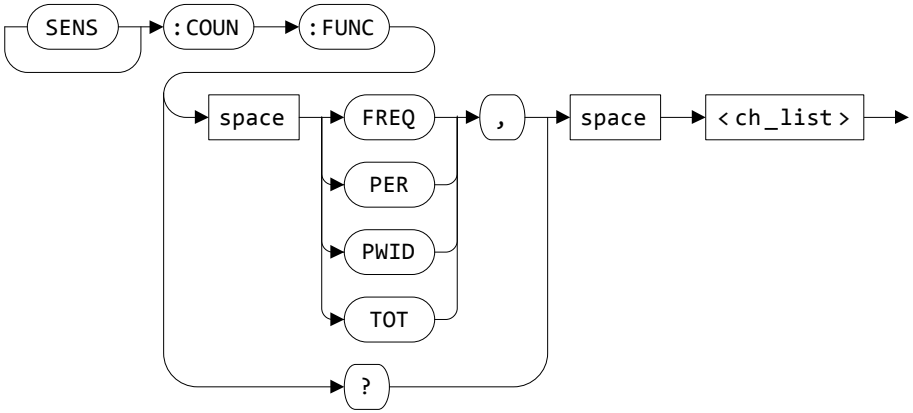
Example

This example illustrates how the `MEASure[:VOLTage][:DC]?` query is used to return the average of 10 voltage measurements from AI channel 101. The `[SENSe:]VOLTage:AVERage` command is used to set the number of measurement points to average.

→ `VOLT:POL UNIP,(@101)`	*This command changes the polarity of channel 101 to unipolar.*
→ `VOLT:RANG 5,(@101)`	*This command sets channel 101 to expect an input voltage within a range of 0 V to +5 V.*
→ `VOLT:AVER 10`	*This command sets 10 measurement points to average.*
→ `MEAS? (@101)`	*This query returns the average value of 10 voltage measurements from channel 101.*
← `3.109018`	

[SENSe:]COUNter:FUNCtion

Syntax



`[SENSe:]COUNter:FUNCtion <mode>, <ch_list>`

This command is used to select a counter function for the counter channel(s) specified in the `<ch_list>` parameter.

The valid options are:

- FREQuency: Frequency measurement
- PERiod: Period measurement
- PWIDth: Pulse width measurement
- TOTalize: Totalizer measurement

`[SENSe:]COUNter:FUNCtion? <ch_list>`

This query returns a string value that represents the current counter function of each counter channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	TOTalize|FREQuency|PERiod|PWIDth	TOTalize
ch_list	Channel list	(@301)|(@302)	-

Remarks

This is an adjunct command for the `MEASure:COUNter:DATA?` query.

See also

"MEASure:COUNter:DATA?" on page 64

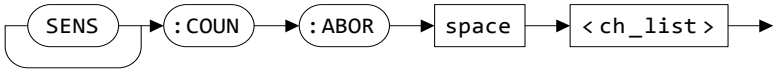
Example

This example illustrates how the `[SENSe:]COUNter:FUNCtion` command is used to make a single frequency measurement on counter channel 301.

→ `*CLS; *RST`	*These commands clears and resets the instrument to its default settings.*
→ `COUN:FUNC FREQ,(@301)`	*This command sets channel 301 to measure frequency.*
→ `MEAS:COUN:DATA? (@301)`	*This query returns the frequency for channel 301.*
← `5.599627`	*The frequency value is returned in kHz.*

[SENSe:]COUNter:ABORt

Syntax



`[SENSe:]COUNter:ABORt <ch_list>`

This command is used to abort a counter or totalizer measurement in progress for the counter channel(s) specified in the `<ch_list>` parameter. This command may be effective to abort a long internally-gated or externally-gated measurement.

NOTE To initiate a totalizer measurement, use the `[SENSe:]COUNter:TOTalize:INITiate` command. To make a single counter measurement (frequency, pulse width, or period) use the `[SENSe:]COUNter:FUNCtion` command together with the `MEASure:COUNter:DATA?` query.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	(@301)|(@302)	-

Remarks

This command does not clear or invalidate any measurement in the instrument memory.

See also

"MEASure:COUNter:DATA?" on page 64

"[SENSe:]COUNter:FUNCtion" on page 127

"[SENSe:]COUNter:TOTalize:INITiate" on page 146

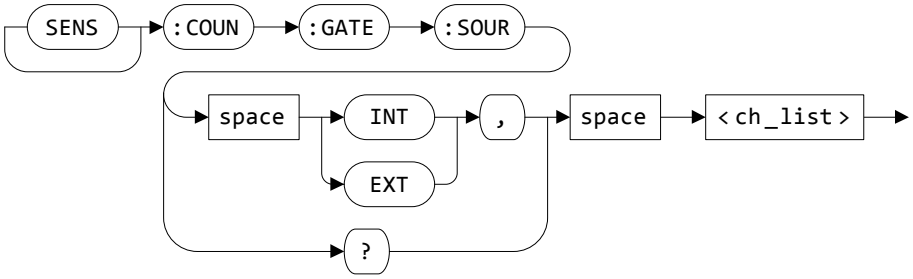
Example

This example illustrates how the `[SENSe:]COUNter:ABORt` command is used to abort the totalizer measurement running on counter channel 301.

→ `*CLS; *RST`	*These commands clears and resets the instrument to its default settings.*
→ `COUN:FUNC TOT,(@301)`	*This command selects the totalizer measurement for channel 301.*
→ `COUN:TOT:INIT (@301)`	*This command starts the totalizer measurement on channel 301 with the instrument default values.*
→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301. The input signals are now allowed to pass.*
→ `MEAS:COUN:DATA? (@301)`	*This query returns the totalized value of the input signals.*
← `64838`	*The totalized value thus far is returned.*
→ `COUN:GATE:CONT DIS (@301)`	*This command disables the gate of channel 301. The totalizing process is halted.*
→ `COUN:ABOR (@301)`	*This command aborts the current totalizer measurement in progress.*
→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301. The input signals are now allowed to pass again; the totalizing process is resumed.*
→ `MEAS:COUN:DATA? (@301)`	
← `64838`	*The returned value “64838” is the same value as the previous query. This is because the previously initiated totalizer measurement has already been aborted.*

[SENSe:]COUNter:GATE:SOURce

Syntax



`[SENSe:]COUNter:GATE:SOURce <mode>, <ch_list>`

This command is used to select a gate source for the counter channel(s) specified in the `<ch_list>` parameter. Do not change the gate source while an initiated measurement is in progress.

The valid options are:

- INTernal: The gate is controlled over a remote interface using SCPI commands.
- EXTernal: The gate is controlled by the COUNT301_GATE or COUNT302_GATE pins.

NOTE For more information on controlling the general purpose digital counter gate over a remote interface, see "[SENSe:]COUNter:GATE:CONTrol" on page 135.

`[SENSe:]COUNter:GATE:SOURce? <ch_list>`

This query returns a string value that represents the gate source selected for each counter channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	INTernal|EXTernal	INTernal
ch_list	Channel list	(@301)|(@302)	-

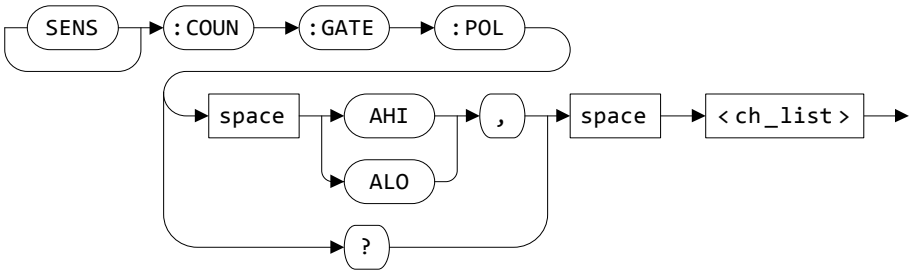
Example

This example illustrates how the `[SENSe:]COUNter:GATE:SOURce` command is used to change the gate source of counter channel 301.

→ `COUN:FUNC TOT,(@301)`	*This command selects the totalizer measurement for channel 301.*
→ `COUN:GATE:SOUR EXT,(@301)`	*This command selects an external gate source for the totalizer measurement on channel 301. This implies that the input signal from the COUNT301_GATE pin will be used to start and stop the counting duration. Supply an input signal through the COUNT301_GATE pin to begin the totalizing process.*
→ `COUN:GATE:POL AHI,(@301)`	*This command sets an active high logic polarity at the gate input line of channel 301.*
→ `COUN:CLK:POL AHI,(@301)`	*This command sets an active high clock polarity for the totalizer measurement on channel 301.*
→ `COUN:CLK:SOUR INT,(@301)`	*This command selects an internal clock source for the totalizer measurement on channel 301.*
→ `COUN:TOT:IVAL 0,(@301)`	*This command sets an initial counter value of 0.*
→ `COUN:TOT:UDOW:DIR UP,(@301)`	*This command sets a counting up direction for the totalizer measurement on channel 301.*
→ `COUN:TOT:INIT (@301)`	*This command starts the totalizer measurement on channel 301.*
→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301. The input signals are now allowed to pass.*

[SENSe:]COUNter:GATE:POLarity

Syntax



`[SENSe:]COUNter:GATE:POLarity <mode>, <ch_list>`

This command is used to set the logic polarity of the gate input line for the counter channel(s) specified in the `<ch_list>` parameter.

The valid options are:

- AHI: Active high polarity
- ALO: Active low polarity

`[SENSe:]COUNter:GATE:POLarity? <ch_list>`

This query returns a string value that represents the gate polarity of each counter channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	AHI|ALO	AHI
ch_list	Channel list	(@301)|(@302)	-

Remarks

Changing the gate polarity while an initiated measurement is in progress is not allowed.

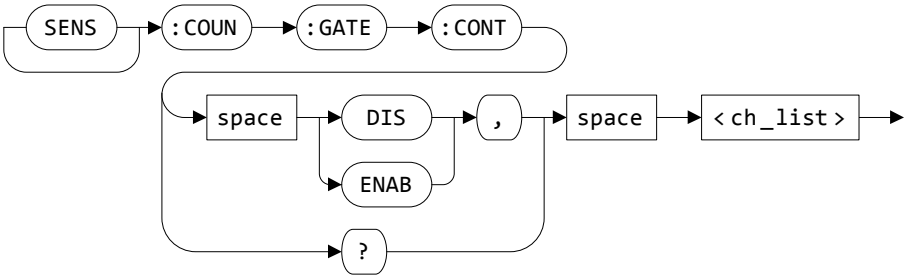
Example

This example illustrates how the `[SENSe:]COUNter:GATE:POLarity` command is used to change the logic polarity of the gate input line on counter channel 301.

→ `COUN:FUNC TOT,(@301)`	*This command selects the totalizer measurement for channel 301.*
→ `COUN:GATE:SOUR INT,(@301)`	*This command selects an internal gate source for the totalizer measurement on channel 301.*
→ `COUN:GATE:POL ALO,(@301)`	*This command sets an active low logic polarity at the gate input line of channel 301. This implies that a falling edge is a valid transition.*
→ `COUN:CLK:POL ALO,(@301)`	*This command sets an active low clock polarity for the totalizer measurement on channel 301.*
→ `COUN:CLK:SOUR INT,(@301)`	*This command selects an internal clock source for the totalizer measurement on channel 301.*
→ `COUN:TOT:IVAL 0,(@301)`	*This command sets an initial counter value of 0.*
→ `COUN:TOT:UDOW:DIR UP,(@301)`	*This command sets a counting up direction for the totalizer measurement on channel 301.*
→ `COUN:TOT:INIT (@301)`	*This command starts the totalizer measurement on channel 301.*
→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301. The input signals are now allowed to pass.*
→ `MEAS:COUN:DATA? (@301)`	*This query returns the totalized value of the input signals.*
← `64838`	*The totalized value thus far is returned.*

[SENSe:]COUNter:GATE:CONTrol

Syntax



`[SENSe:]COUNter:GATE:CONTrol <mode>, <ch_list>`

This command is used to enable or disable the gate for the counter channel(s) specified in the `<ch_list>` parameter.

The valid options are:

- ENAB: Enables gate; input signals are allowed to pass through.
- DIS: Disables gate; input signals are not allowed to pass through.

`[SENSe:]COUNter:GATE:CONTrol? <ch_list>`

This query returns a string value that represents the gate status of each counter channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	DIS|ENAB	DIS
ch_list	Channel list	(@301)|(@302)	-

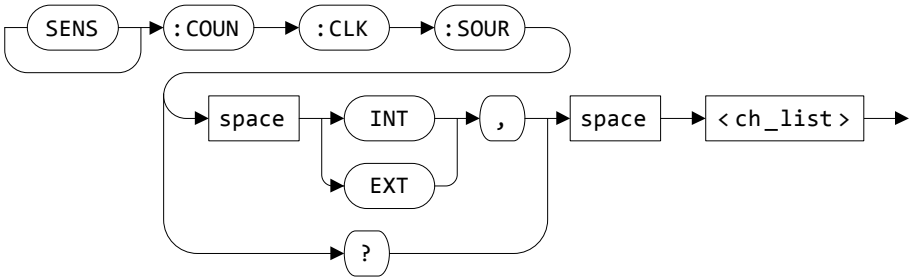
Example

This example illustrates how the `[SENSe:]COUNter:GATE:CONTrol` command is used to enable the gate on counter channel 301.

→ `*CLS; *RST`	*These commands clears and resets the instrument to its default settings.*
→ `COUN:FUNC TOT,(@301)`	*This command selects the totalizer measurement for channel 301.*
→ `COUN:TOT:INIT (@301)`	*This command starts the totalizer measurement on channel 301 with the instrument default values.*
→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301. The input signals are now allowed to pass; the totalizing process can begin.*
→ `MEAS:COUN:DATA? (@301)`	*This query returns the totalized value of the input signals.*
← `64838`	*The totalized value thus far is returned.*
→ `COUN:GATE:CONT DIS (@301)`	*This command disables the gate of channel 301. The totalizing process is halted.*
→ `MEAS:COUN:DATA? (@301)`	
← `64838`	*The totalizing process is halted because the gate has been disabled.*
→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301. The input signals are now allowed to pass again; the totalizing process is resumed.*
→ `MEAS:COUN:DATA? (@301)`	
← `76434`	*The totalized value thus far is returned.*

[SENSe:]COUNter:CLocK:SOURce

Syntax



`[SENSe:]COUNter:CLocK:SOURce <mode>, <ch_list>`

This command is used to set the clock source for the counter channel(s) specified in the `<ch_list>` parameter.

The valid options are:

- INTernal: Selects the DAQ device internal counter clock frequency (12 MHz) as the clock source.
- EXTernal: Selects the COUNT301_CLK or COUNT302_CLK pin as the counter clock source.

NOTE

- The clock frequency coming in from the COUNT301_CLK or COUNT302_CLK pin must be set in the DAQ device via the `[SENSe:]COUNter:CLocK:EXTernal` command. For example, if a 10 MHz clock frequency is expected from the COUNT301_CLK or COUNT302_CLK pin, you must send the `[SENSe:]COUNter:CLocK:EXTernal 10000` command to the DAQ device.
- See "[SENSe:]COUNter:CLocK:INTernal?" on page 139 for more information on how to return the internal counter clock frequency.
- See "[SENSe:]COUNter:CLocK:EXTernal" on page 140 for more information on how to set the external counter clock frequency.

`[SENSe:]COUNter:CLocK:SOURce? <ch_list>`

This query returns a string value that represents the clock source of each counter channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	INTernal|EXTernal	INTernal
ch_list	Channel list	(@301)|(@302)	-

See also

"[SENSe:]COUNter:CLocK:INTernal?" on page 139

"[SENSe:]COUNter:CLocK:EXTernal" on page 140

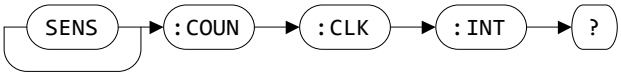
Example

This example illustrates how the `[SENSe:]COUNter:CLocK:SOURce` command is used to set the clock source of counter channel 301 to an external clock source.

→ `COUN:CLK:SOUR EXT,(@301)`	*This command sets the instrument counter clock source to external. A 10 MHz clock frequency signal is supplied to the COUNT301_CLK pin.*
→ `COUN:CLK:EXT 10000,(@301)`	*Since we are expecting a 10 MHz clock frequency from the COUNT301_CLK pin, use this command to set the external counter clock frequency to 10 MHz.*
→ `COUN:CLK:SOUR? (@301)`	*This query returns the counter clock source for channel 301.*
← `EXT`	

[SENSe:]COUNter:CLocK:INTernal?

Syntax



`[SENSe:]COUNter:CLocK:INTernal?`

This query returns a string value that represents the internal clock frequency. The value returned is expressed in kilohertz (kHz).

Supported models

U2531A, U2541A, U2542A

See also

"[SENSe:]COUNter:CLocK:SOURce" on page 137

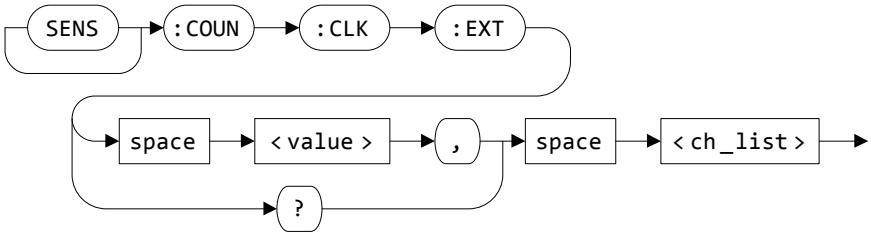
Example

This example illustrates how the **`[SENSe:]COUNter:CLocK:INTernal?`** query is used to return the internal clock frequency.

→ `COUN:CLK:INT?`	*This query returns the internal clock frequency.*
← `12000`	*The internal timebase frequency is returned – 12 MHz. This value is used when you set the counter clock source to the instrument internal clock source.*

[SENSe:]COUNter:CLocK:EXTernal

Syntax



`[SENSe:]COUNter:CLocK:EXTernal <value>, <ch_list>`

This command is used to set the external clock frequency for the counter channel(s) specified in the `<ch_list>` parameter. The frequency set in the `<value>` parameter must be the same as the frequency value supplied to the COUNT301_CLK or COUNT302_CLK pin when the counter clock source is set to `EXTernal`. The clock frequency is expressed in units of kilohertz (kHz).

NOTE See "[SENSe:]COUNter:CLocK:SOURce" on page 137 for more information on selecting a general purpose digital counter clock source.

`[SENSe:]COUNter:CLocK:EXTernal? <ch_list>`

This query returns a string value that represents the external clock frequency of each counter channel specified. Multiple responses are separated by commas. The value returned is expressed in kilohertz (kHz).

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
value	Numeric	1000 kHz (1 MHz) to 10000 kHz (10 MHz)	0
ch_list	Channel list	(@301)|(@302)	-

See also

"[SENSe:]COUNter:CLocK:SOURce" on page 137

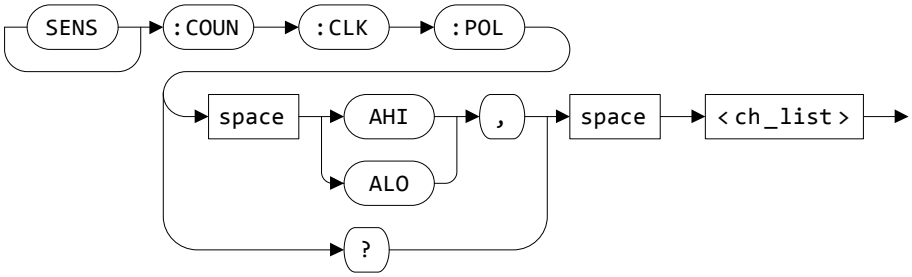
Example

This example illustrates how the `[SENSe:]COUNter:CLocK:EXTernal` command is used to set the external clock frequency of counter channel 301.

→ `COUN:CLK:SOUR EXT,(@301)`	*This command sets the instrument counter clock source to the COUNT301_CLK pin. A 10 MHz clock frequency signal is supplied to the COUNT301_CLK pin.*
→ `COUN:CLK:EXT 10000,(@301)`	*Since we are expecting a 10 MHz clock frequency from the COUNT301_CLK pin, use this command to set the external counter clock frequency to 10 MHz.*
→ `COUN:CLK:EXT? (@301)`	*This query returns the external timebase frequency of channel 301.*
← `10000`	*The external timebase frequency is returned in kHz.*

[SENSe:]COUNter:CLocK:POLarity

Syntax



`[SENSe:]COUNter:CLocK:POLarity <mode>, <ch_list>`

This command is used to set the clock polarity for the counter channel(s) specified in the `<ch_list>` parameter.

The valid options are:

- AHI: Active high polarity
- ALO: Active low polarity

`[SENSe:]COUNter:CLocK:POLarity? <ch_list>`

This query returns a string value that represents the clock polarity of each counter channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	AHI|ALO	AHI
ch_list	Channel list	(@301)|(@302)	-

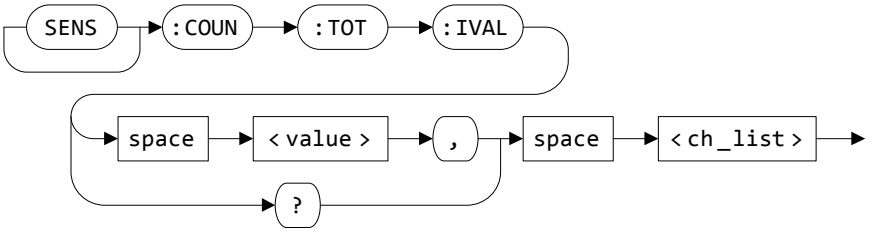
Example

This example illustrates how the `[SENSe:]COUNter:CLocK:POLarity` command is used to change the clock polarity on counter channel 301.

→ `COUN:FUNC TOT,(@301)`	*This command selects the totalizer measurement for channel 301.*
→ `COUN:GATE:SOUR INT,(@301)`	*This command selects an internal gate source for the totalizer measurement on channel 301.*
→ `COUN:GATE:POL ALO,(@301)`	*This command sets an active low logic polarity at the gate input line of channel 301.*
→ `COUN:CLK:POL ALO,(@301)`	*This command sets an active low clock polarity for the totalizer measurement on channel 301. This implies that the instrument will count on the falling edge.*
→ `COUN:CLK:SOUR INT,(@301)`	*This command selects an internal clock source for the totalizer measurement on channel 301.*
→ `COUN:TOT:IVAL 0,(@301)`	*This command sets an initial counter value of 0.*
→ `COUN:TOT:UDOW:DIR UP,(@301)`	*This command sets a counting up direction for the totalizer measurement on channel 301.*
→ `COUN:TOT:INIT (@301)`	*This command starts the totalizer measurement on channel 301.*
→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301. The input signals are now allowed to pass.*
→ `MEAS:COUN:DATA? (@301)`	*This query returns the totalized value of the input signals.*
← `64838`	*The totalized value thus far is returned.*

[SENSe:]COUNter:TOTalize:IVALue

Syntax



`[SENSe:]COUNter:TOTalize:IVALue <value>, <ch_list>`

This command sets the initial counter value for the counter channel(s) specified in the `<ch_list>` parameter.

`[SENSe:]COUNter:TOTalize:IVALue? <ch_list>`

This query returns a numeric value that represents the initial counter value of each counter channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
value	Numeric	0 to 2147483647 ($2^{31} - 1$)	0
ch_list	Channel list	(@301)|(@302)	-

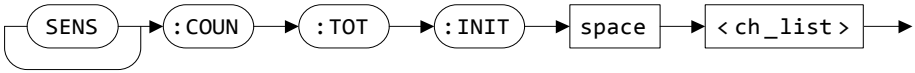
Example

This example illustrates how the `[SENSe:]COUNter:TOTalize:IVALue` command is used to set an initial counter value on counter channel 301.

→ `COUN:FUNC TOT,(@301)`	*This command selects the totalizer measurement for channel 301.*
→ `COUN:GATE:SOUR INT,(@301)`	*This command selects an internal gate source for the totalizer measurement on channel 301.*
→ `COUN:GATE:POL ALO,(@301)`	*This command sets an active low logic polarity at the gate input line of channel 301.*
→ `COUN:CLK:POL AHI,(@301)`	*This command sets an active high clock polarity for the totalizer measurement on channel 301.*
→ `COUN:CLK:SOUR INT,(@301)`	*This command selects an internal clock source for the totalizer measurement on channel 301.*
→ `COUN:TOT:IVAL 1000,(@301)`	*This command sets an initial counter value of 1000.*
→ `COUN:TOT:UDOW:DIR UP,(@301)`	*This command sets a counting up direction for the totalizer measurement on channel 301.*
→ `COUN:TOT:INIT (@301)`	*This command starts the totalizer measurement on channel 301.*
→ `MEAS:COUN:DATA? (@301)`	*This query returns the totalized value of the input signals.*
← `1000`	*The totalized value thus far is returned. Since the general purpose digital counter gate has not been enabled the totalized value thus far is the initial counter value.*
→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301. The input signals are now allowed to pass.*
→ `MEAS:COUN:DATA? (@301)`	
← `65838`	*The totalized value thus far is returned.*

[SENSe:]COUNter:TOTalize:INITiate

Syntax



`[SENSe:]COUNter:TOTalize:INITiate <ch_list>`

This command initiates a totalizer measurement on the counter channel(s) specified in the `<ch_list>` parameter.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	(@301)|(@302)	-

Example

This example illustrates how the `[SENSe:]COUNter:TOTalize:INITiate` command is used to initiate the totalizer measurement on counter channel 301.

→ `*CLS; *RST`	*These commands clears and resets the instrument to its default settings.*
→ `COUN:FUNC TOT,(@301)`	*This command selects the totalizer measurement for channel 301.*
→ `COUN:TOT:INIT (@301)`	*This command starts the totalizer measurement on channel 301 with the instrument default values.*
→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301. The input signals are now allowed to pass.*

[SENSe:]COUNter:TOTalize:CLEar

Syntax

`[SENSe:]COUNter:TOTalize:CLEar <ch_list>`

This command clears the totalized value thus far on the counter channel(s) specified in the `<ch_list>` parameter.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
ch_list	Channel list	(@301)|(@302)	-

Remarks

This command clears the counter to zero. Sending this command will not return the value of the general purpose digital counter to the initalized counter value. For more information on setting an initial counter value see "[SENSe:]COUNter:TOTalize:IVALue" on page 144.

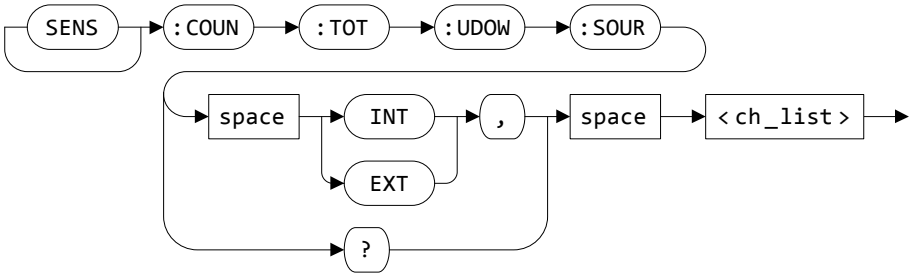
Example

This example illustrates how the `[SENSe:]COUNter:TOTalize:CLEar` command is used to clear the counter values of counter channel 301.

→ `*CLS; *RST`	*These commands clears and resets the instrument to its default settings.*
→ `COUN:FUNC TOT,(@301)`	*This command selects the totalizer measurement for channel 301.*
→ `COUN:TOT:INIT (@301)`	*This command starts the totalizer measurement on channel 301 with the instrument default values.*
→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301.*
→ `MEAS:COUN:DATA? (@301)`	*This query returns the totalized value of the input signals.*
← `64838`	*The totalized value thus far is returned.*
→ `COUN:GATE:CONT DIS (@301)`	*This command disables the gate of channel 301.*
→ `COUN:ABOR (@301)`	*This command aborts the current totalizer measurement in progress.*
→ `MEAS:COUN:DATA? (@301)`	
← `75643`	*The returned value “75643” is the last acquired value before the totalizer measurement is aborted.*
→ `SENS:COUN:TOT:CLE (@301)`	*This command clears all the acquired counter values in the output buffer.*
→ `MEAS:COUN:DATA? (@301)`	
← `0`	

[SENSe:]COUNter:TOTalize:UDOWn:SOURce

Syntax



`[SENSe:]COUNter:TOTalize:UDOWn:SOURce <mode>, <ch_list>`

This command is used to set the up/down counting source for the counter channel(s) specified in the `<ch_list>` parameter.

The valid options are:

- INTernal: The up/down counting source is controlled over a remote interface using SCPI commands.
- EXTernal: The up/down counting source is controlled by the COUNT301_UPDOWN or COUNT302_UPDOWN pins.

`[SENSe:]COUNter:TOTalize:UDOWn:SOURce? <ch_list>`

This query returns a string value that indicates the up/down counting source of each counter channel specified. Multiple responses are separated by commas.

Supported model

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	INTernal|EXTernal	INTernal
ch_list	Channel list	(@301)|(@302)	-

Example

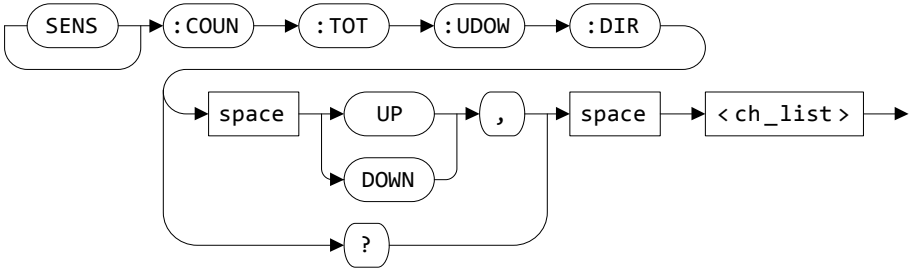
This example illustrates how the `[SENSe:]COUNter:TOTalize:UDOWn:SOURce` command is used to set the COUNT301_UPDOWN pin as the up/down counting source on counter channel 301.

→ `COUN:FUNC TOT,(@301)`	*This command selects the totalizer measurement for channel 301.*
→ `COUN:GATE:SOUR INT,(@301)`	*This command selects an internal gate source for the totalizer measurement on channel 301.*
→ `COUN:GATE:POL AHI,(@301)`	*This command sets an active high logic polarity at the gate input line of channel 301.*
→ `COUN:CLK:POL AHI,(@301)`	*This command sets an active high clock polarity for the totalizer measurement on channel 301.*
→ `COUN:CLK:SOUR INT,(@301)`	*This command selects an internal clock source for the totalizer measurement on channel 301.*
→ `COUN:TOT:IVAL 0,(@301)`	*This command sets an initial counter value of 0.*
→ `COUN:TOT:UDOW:SOUR EXT, (@301)`	*This command selects an external up/down counting source for the totalizer measurement on channel 301. This implies that the input signal from the COUNT301_UPDOWN pin will be used as the up/down counting source.*
→ `COUN:TOT:UDOW:DIR UP,(@301)`	*This command sets a counting up direction for the totalizer measurement on channel 301.*
→ `COUN:TOT:INIT (@301)`	*This command starts the totalizer measurement on channel 301.*

→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301. The input signals are now allowed to pass.*
→ `MEAS:COUN:DATA? (@301)`	*This query returns the totalized value of the input signals.*
← `64838`	*The totalized value thus far is returned.*

[SENSe:]COUNter:TOTalize:UDOWn:DIRection

Syntax



`[SENSe:]COUNter:TOTalize:UDOWn:DIRection <mode>, <ch_list>`

This command is used to set the direction of the up/down counting process for the counter channel(s) specified in the `<ch_list>` parameter.

The valid options are:

- UP: Counting up direction
- DOWN: Counting down direction

`[SENSe:]COUNter:TOTalize:UDOWn:DIRection? <ch_list>`

This query returns a string value that indicates the direction of the counting process of each counter channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	UP|DOWN	UP
ch_list	Channel list	(@301)|(@302)	-

Example

This example illustrates how the `[SENSe:]COUNter:TOTalize:UDOWn:DIRection` command is used to set a counting down direction for the totalizer measurement on counter channel 301.

→ `COUN:FUNC TOT,(@301)`	*This command selects the totalizer measurement for channel 301.*
→ `COUN:GATE:SOUR INT,(@301)`	*This command selects an internal gate source for the totalizer measurement on channel 301.*
→ `COUN:GATE:POL AHI,(@301)`	*This command sets an active high logic polarity at the gate input line of channel 301.*
→ `COUN:CLK:POL AHI,(@301)`	*This command sets an active high clock polarity for the totalizer measurement on channel 301.*
→ `COUN:CLK:SOUR INT,(@301)`	*This command selects an internal clock source for the totalizer measurement on channel 301.*
→ `COUN:TOT:IVAL 0,(@301)`	*This command sets an initial counter value of 0.*
→ `COUN:TOT:UDOW:DIR DOWN,(@301)`	*This command sets a counting down direction for the totalizer measurement on channel 301.*
→ `COUN:TOT:INIT (@301)`	*This command starts the totalizer measurement on channel 301.*
→ `COUN:GATE:CONT ENAB,(@301)`	*This command enables the gate of channel 301. The input signals are now allowed to pass.*
→ `MEAS:COUN:DATA? (@301)`	*This query returns the totalized value of the input signals.*
← `6483843`	*The totalized value thus far is returned.*
→ `MEAS:COUN:DATA? (@301)`	
← `4348901`	*The totalized value thus far is returned.*

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

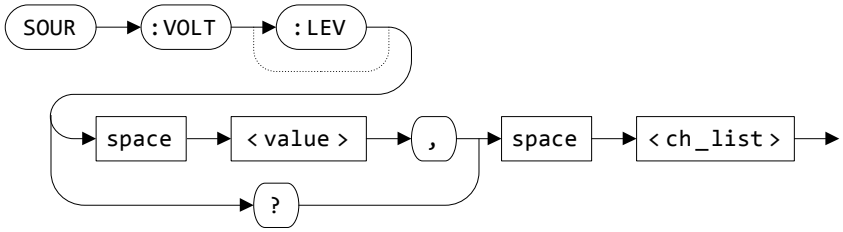
10 SOURce Subsystem

SOURce:VOLTage[:LEVel] 156
SOURce:VOLTage:POLarity 158
SOURce:VOLTage:RSouRCe 160
SOURce:VOLTage:RVOLtage 162
SOURce:DIGital:DATA 164
SOURce:DIGital:DATA:BIT 166

This chapter describes the SOURce commands used to program the U2500A Series DAQ device over a remote interface. Use the SOURce commands to change the AO channel settings prior to making a single voltage output. You can also output digital data to the DIO channels.

SOURce:VOLTage[:LEVel]

Syntax



`SOURce:VOLTage[:LEVel] <value>, <ch_list>`

This command is used to output a single voltage level to the AO channel(s) specified in the `<ch_list>` parameter. The voltage set is supplied as soon as the command is executed.

NOTE **This command receives a voltage value, scales it to the proper binary value and writes a binary value to the designated AO channel. After setting the desired voltage level, the command closes the corresponding output relays and enable the outputs from the specified channel(s). You do not need to send the `OUTPut ON` command to enable the output.**

`SOURce:VOLTage[:LEVel]? <ch_list>`

This query returns a numeric value that represents the output voltage level of each analog channel specified. Multiple responses are separated by commas. The value returned is expressed in volts (V).

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
value	Numeric	–10 V to +10 V (12-bits resolution)	0 V
ch_list	Channel list	(@201)|(@202)	-

Remarks

If the output has already been enabled (via the **OUTPut ON**) command, sending this command will generate an error. You must first disable the output by sending the **OUTPut OFF** command.

See also

"SOURce:VOLTage:POLarity" on page 158

"SOURce:VOLTage:RSouRCe" on page 160

"SOURce:VOLTage:RVOLtage" on page 162

Example

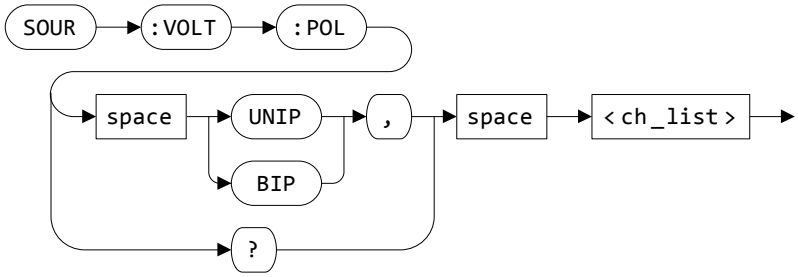
This example illustrates how the **SOURce:VOLTage[:LEVel]** command is used to output a 5 V DC power supply for channel 201.

→ `SOUR:VOLT 5,(@201)`	*This command outputs a 5 V DC power supply from channel 201.*
→ `SOUR:VOLT? (@201)`	*This query returns the current output voltage level from channel 201.*

← `5`

SOURce:VOLTage:POLarity

Syntax



`SOURce:VOLTage:POLarity <mode>, <ch_list>`

This command is used to set the polarity of the AO channel(s) specified in the `<ch_list>` parameter. This command is applicable for single voltage output operations.

The valid options are:

– UNIPolar

– BIPolar

`SOURce:VOLTage:POLarity? <ch_list>`

This query returns a string value that represents the polarity of each AO channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	UNIPolar|BIPolar	BIPolar
ch_list	Channel list	(@201)|(@202)	-

Remarks

This command will force reset the output voltage level to zero if the voltage output is set to a negative voltage level.

See also

"SOURce:VOLTage[:LEVel]" on page 156

"SOURce:VOLTage:RSouRCe" on page 160

"SOURce:VOLTage:RVOLtage" on page 162

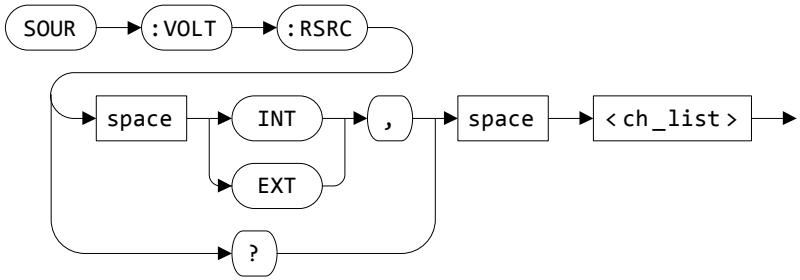
Example

This example illustrates how the `SOURce:VOLTage[:LEVel]` command is used to output a 7.5 V DC power supply for AO channel 201. The `SOURce:VOLTage:POLarity` command is used to change the polarity of AO channel 201.

→ `SOUR:VOLT:POL UNIP,(@201)`	*This command changes the polarity of channel 201 to unipolar. This limits the output level to a positive voltage level.*
→ `SOUR:VOLT:RSRC EXT,(@201)`	*This command changes the reference voltage source of channel 201 to an external source.*
→ `SOUR:VOLT:RVOL 5`	*This command selects an external reference voltage value of 5 V.*
→ `SOUR:VOLT 7.5,(@201)`	*This command outputs a 7.5 V DC power supply from channel 201.*

SOURce:VOLTage:RSouRCe

Syntax



`SOURce:VOLTage:RSouRCe <mode>, <ch_list>`

This command is used to set the reference voltage source for the AO channel(s) specified in the `<ch_list>` parameter. This command is applicable for single voltage output operations.

The valid options are:

- INTernal: Selects the internal on-board reference voltage as the reference voltage source.
- EXTernal: Selects the AO_EXT_REF pin as the reference voltage source.

NOTE

- **The reference voltage coming in from the AO_EXT_REF pin must be set in the DAQ device via the `SOURce:VOLTage:RVOLtage` command. For example, if a 5 V reference voltage is expected from the AO_EXT_REF pin, you must send the `SOURce:VOLTage:RVOLtage 5` command to the DAQ device.**
- **See "SOURce:VOLTage:RVOLtage" on page 162 for more information on how to set the external reference voltage value.**

SOURce:VOLTage:RSouRCe? <ch_list>

This query returns a string value that represents the reference voltage source of each AO channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
mode	Discrete	INTernal|EXTernal	INTernal
ch_list	Channel list	(@201)|(@202)	-

See also

"SOURce:VOLTage[:LEVel]" on page 156

"SOURce:VOLTage:POLarity" on page 158

"SOURce:VOLTage:RVOLtage" on page 162

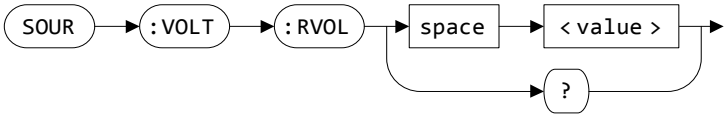
Example

This example illustrates how the **SOURce:VOLTage[:LEVel]** command is used to output a 7.5 V DC power supply for AO channel 201. The **SOURce:VOLTage:RSouRCe** command is used to change the reference voltage source of AO channel 201.

→ `SOUR:VOLT:POL UNIP,(@201)`	*This command changes the polarity of channel 201 to unipolar.*
→ `SOUR:VOLT:RSRC EXT,(@201)`	*This command sets the instrument reference voltage source to the AO_EXT_REF pin. A 5 V voltage source is supplied to the AO_EXT_REF pin.*
→ `SOUR:VOLT:RVOL 5`	*Since we are expecting a 5 V voltage source from the AO_EXT_REF pin, this command is used to set the external reference voltage to 5 V.*
→ `SOUR:VOLT 7.5,(@201)`	*This command outputs a 7.5 V DC power supply from channel 201.*

SOURce:VOLTage:RVOLtage

Syntax



`SOURce:VOLTage:RVOLtage <value>`

This command is used to set the external reference voltage value. The reference voltage set in the `<value>` parameter must be the same as the voltage value supplied to the AO_EXT_REF pin when the reference voltage source is set to `EXTernal`. The reference voltage value is expressed in units of volts (V).

NOTE See "SOURce:VOLTage:RSouRCe" on page 160 for more information on selecting a reference voltage source.

`SOURce:VOLTage:RVOLtage?`

This query returns a string value that represents the external reference voltage value. The value returned is expressed in volts (V).

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	0 < value ≤ 10	10 V

Remarks

The reference voltage value cannot be set to zero or a negative value.

See also

"SOURce:VOLTage[:LEVel]" on page 156

"SOURce:VOLTage:POLarity" on page 158

"SOURce:VOLTage:RVOLtage" on page 162

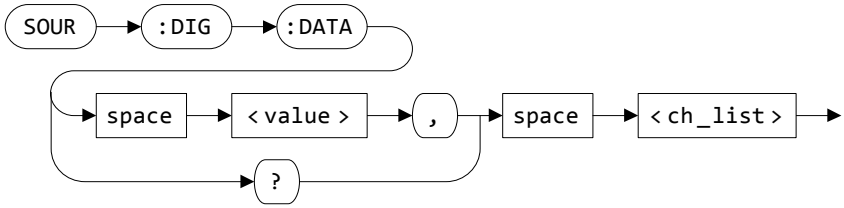
Example

This example illustrates how the **SOURce:VOLTage[:LEVel]** command is used to output a 7.5 V DC power supply for AO channel 201. The **SOURce:VOLTage:RVOLtage** command is used to change the reference voltage level.

→ `SOUR:VOLT:POL UNIP,(@201)`	*This command changes the polarity of channel 201 to unipolar.*
→ `SOUR:VOLT:RSRC EXT,(@201)`	*This command sets the instrument reference voltage source to the AO_EXT_REF pin. A 5 V voltage source is supplied to the AO_EXT_REF pin.*
→ `SOUR:VOLT:RVOL 5`	*Since we are expecting a 5 V voltage source from the AO_EXT_REF pin, this command is used to set the external reference voltage to 5 V.*
→ `SOUR:VOLT 7.5,(@201)`	*This command outputs a 7.5 V DC power supply from channel 201.*

SOURce:DIGital:DATA

Syntax



`SOURce:DIGital:DATA <value>, <ch_list>`

This command is used to output a 4-bit or 8-bit digital pattern to the DIO channel(s) specified in the `<ch_list>` parameter. The specified DIO channels have to be configured as `OUTPut`, prior to sourcing any digital data.

NOTE To configure the DIO channel for output operation, see "CONFigure:DIGital:DIRection" on page 52.

`SOURce:DIGital:DATA? <ch_list>`

This query returns a numeric value that represents the digital data of each DIO channel specified. Multiple responses are separated by commas.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
value	Numeric	– Channel 501 and 502: 0 to 255 – Channel 503 and 504: 0 to 15	-
ch_list	Channel list	(@501) to (@504)[a]	-

[a] Port A: Channel 501 (8-bits)
Port B: Channel 502 (8-bits)
Port C: Channel 503 (4-bits)
Port D: Channel 504 (4-bits)

Remarks

Prior to using this command, the DIO channel has to be configured as **`OUTPut`**; else a system error will be generated. Using this command or query right after a system reset (**`*RST`**) will also generate a system error. This is because after a system reset, the DIO channel by default is set to **`INPut`**.

See also

"CONFigure:DIGital:DIRection" on page 52

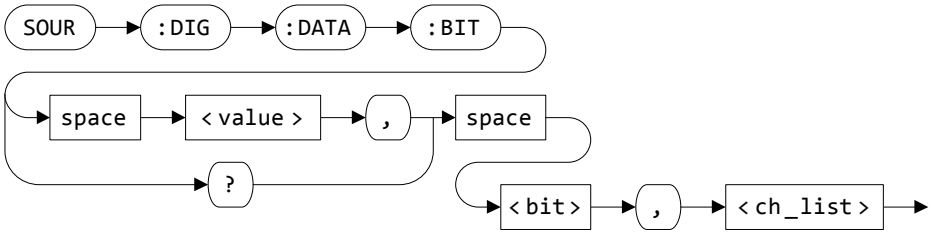
Example

This example illustrates how the **`SOURce:DIGital:DATA`** command is used to output a digital data value of 256 for DIO channel 501. The **`CONFigure:DIGital:DIRection`** command is used to configure the DIO channel for output operations.

→ `CONF:DIG:DIR OUTP,(@501)`	*This command configures channel 501 for output operations.*
→ `SOUR:DIG:DATA 256,(@501)`	*This command outputs a digital data value of 256 through channel 501.*
→ `SOUR:DIG:DATA? (@501)`	*This query returns the digital data value in channel 501.*
← `256`	

SOURce:DIGital:DATA:BIT

Syntax



`SOURce:DIGital:DATA:BIT <value>, <bit>, <ch_list>`

This command sets or clears individual bits on the DIO channel(s) specified in the `<ch_list>` parameter. The specified DIO channels have to be configured as `OUTPut`, prior to sourcing any digital data.

NOTE Once a channel is configured as `OUTPut`, all bits within that channel are set as output. To configure the DIO channel for output operation, see "CONFigure:DIGital:DIRection" on page 52.

`SOURce:DIGital:DATA:BIT? <bit>, <ch_list>`

This query returns a boolean value of 1 (logic high) or 0 (logic low) for the specified bit within the specified channel.

Supported models

U2531A, U2541A, U2542A

Parameters

Item	Type	Range of values	Default value
value	Boolean	0 | 1	0
bit	Numeric	– Channel 501 and 502: 0 to 7 – Channel 503 and 504: 0 to 3	-
ch_list	Channel list	(@501) to (@504)[a]	-

[a] Port A: Channel 501 (8-bits)
Port B: Channel 502 (8-bits)
Port C: Channel 503 (4-bits)
Port D: Channel 504 (4-bits)

Remarks

Prior to using this command, the DIO channel has to be configured as `OUTPut`; else a system error will be generated. Using this command or query right after a system reset (`*RST`) will also generate a system error. This is because after a system reset, the DIO channel by default is set to `INPut`.

See also

"CONFigure:DIGital:DIRection" on page 52

Example

This example illustrates how the `SOURce:DIGital:DATA:BIT` command is used to set bit 4 of DIO channel 501 to a logic high state. The `CONFigure:DIGital:DIRection` command is used to configure the DIO channel for output operations.

→ `CONF:DIG:DIR OUTP,(@501)`	*This command configures channel 501 for output operations.*
→ `SOUR:DIG:DATA:BIT 1,4,(@501)`	*This command sets bit 4 of channel 501 to 1 (logic high).*
→ `SOUR:DIG:DATA:BIT? 4,(@501)`	*This query returns the logic state of bit 4 in channel 501.*

← `1`

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

11 SYSTem Subsystem

SYSTem:CDEScription? 170
SYSTem:ERRor? 171

This section describes the SYSTem commands used to program the U2500A Series DAQ device over a remote interface. The SYSTem:ERRor? query can be used to return the error stored in the DAQ device error queue while the SYSTem:CDEScription? query can be used to return the slot and chassis number of a DAQ device when slotted into a U2781A USB modular instrument chassis.

SYSTem:CDEScription?

Syntax

`SYSTem:CDEScription?`

This command is used when the U2300A Series DAQ device is slotted into a U2781A USB modular instrument chassis. It identifies and returns the slot and chassis number that the module is slotted into.

Supported models

U2531A, U2541A, U2542A

Remarks

This command is only applicable when the DAQ device is used in the U2781A USB modular instrument chassis. If the DAQ device is in standalone mode, using this command will return the value "`7,0`".

NOTE For more information on identifying multiple modules or chassis connected to the same PC host, see "Identifying Modules Location" in the *U2781A USB Modular Instrument Chassis User's Guide*.

Example

This example illustrates how the `SYSTem:CDEScription?` query is used to identify and return the geographical address of the DAQ device.

→ `SYST:CDES?`	*This query returns the slot and chassis number that the DAQ device is plugged into.*
← `5,2`	*The returned value "5,2" indicates that the DAQ device is slotted into slot number 5 of chassis number 2.*

SYSTem:ERRor?

Syntax

`SYSTem:ERRor?`

This query returns the next error number and its corresponding message string from the DAQ device error queue.

Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored. Reading this error will clear this error allowing the next stored error to be read (if there are other errors stored). Once you have read all of the interface-specific errors, the errors in the global error queue will be retrieved.

NOTE See Chapter 16, “Error Messages” on page 231 for a list of all error numbers and their respective message string.

Supported models

U2531A, U2541A, U2542A

Remarks

- If more than 10 errors have occurred, the last error stored in the queue (the most recent error) is replaced with error number `-350,"Queue overflow"`. No additional errors are stored until you remove the previously stored errors from the queue.
- If no errors have occurred when you read the error queue, the instrument responds with `+0,"No error"`.
- Error conditions are also summarized in the Status Byte Register.
- The interface-specific and global error queues are cleared by the clear status (`*CLS`) command and when the power is cycled. The error queue will not be cleared by a factory reset (`*RST`) command.

See also

"*CLS" on page 216

"*RST" on page 224

"*SRE" on page 226

Example

This example illustrates how the **SYSTem:ERRor?** query is used to read the errors stored in the DAQ device error queue.

→ `SYST:ERR?`	*This command reads the error(s) stored in the DAQ device error queue.*
← `+0,"No error"`	*The returned value "+0" indicates that there are no errors recorded.*
→ `ROUT ENAB 1,(@201)`	*This incomplete command (missing colon) generates a command syntax error when sent.*
→ `SYST:ERR?`	
← `-113,"Undefined header"`	*The returned value "–113" indicates that an invalid command was specified.*

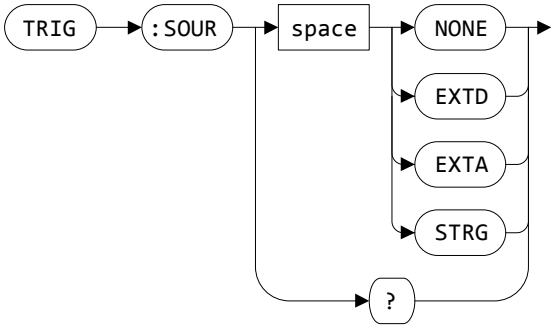
12 TRIGger Subsystem

TRIGger:SOURce 174
TRIGger:TYPe 176
TRIGger:DCouNT 179
TRIGger:ATRiGger:SOURce 182
TRIGger:ATRiGger:CONDition 184
TRIGger:ATRiGger:HTHReshold 187
TRIGger:ATRiGger:LTHReshold 189
TRIGger:DTRiGger:POLarity 191

This chapter describes the TRIGger commands used to program the U2500A Series DAQ device over a remote interface. Use the TRIGger commands to set the AI trigger source and type. You can also configure the analog trigger settings.

TRIGger:SOURce

Syntax



`TRIGger:SOURce <mode>`

This command is used to set the trigger source for input operations.

The valid options are:

- NONE: Immediate triggering.
- EXTD: Selects the external digital trigger (EXTD_AI_TRIG) pin as the triggering source.
- EXTA: Selects the external analog trigger (EXTA_TRIG) pin as the triggering source.
- STRG: Star triggering.

NOTE

- **Star triggering (STRG) is used when the DAQ device is connected into the U2781A USB modular instrument chassis.**
- **For more information on selecting a trigger source, see “Trigger Sources” in the *Keysight U2500A Series USB Simultaneous Sampling Multifunction Data Acquisition User’s Guide*.**

`TRIGger:SOURce?`

This query returns a string value that represents the trigger source selected for input operations.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
mode	Discrete	NONE|EXTD|EXTA|STRG	NONE

Remarks

Only `POST` trigger type is allowed when immediate triggering (`NONE`) is selected. For more information on available trigger types, see "TRIGger:TYPe" on page 176.

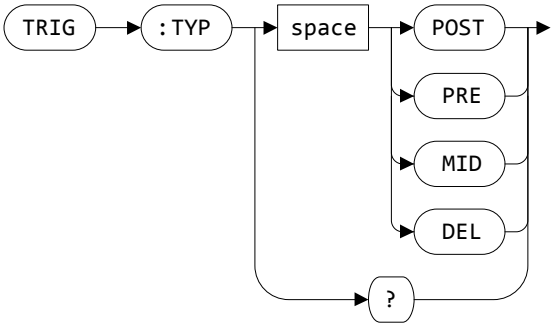
Example

This example illustrates how the `TRIGger:SOURce` command is used to select the EXTD_AI_TRIG pin as the input trigger source.

→ `ROUT:ENAB ON,(@101)`	*This command enables channel 101 for input operations.*
→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `ACQ:POIN 100`	*This command sets the instrument to acquire 100 sample points.*
→ `TRIG:SOUR EXTD`	*This command sets the EXTD_AI_TRIG pin as the input triggering source.*
→ `TRIG:DTRG:POL POS`	*This command sets a positive-edge trigger condition for the digital trigger source.*
→ `TRIG:TYP POST`	*This command set a post-trigger acquisition for the analog input.*
→ `DIG`	*This command starts the single-shot acquisition process when a positive-edge digital signal is received from the EXTD_AI_TRIG pin.*

TRIGger:TYPe

Syntax



`TRIGger:TYPe <mode>`

This command is used to set the trigger type for input operations.

The valid options are:

- POST (Post-trigger): Input is acquired immediately after the trigger condition is met.
- PRE (Pre-trigger): Input is acquired immediately and is stopped when the trigger condition is met.
- MID (Mid-trigger): Input is acquired before and after the trigger condition is met. The sample points acquired before and after the trigger is equally divided.
- DEL (Delay-trigger): Input is acquired when the delay count reaches zero. The delay count starts immediately after the trigger condition is met.

NOTE

- See "TRIGger:DCouNT" on page 179 for more information on how to set a delay counter value.
- For more information on selecting a trigger type, see "Trigger types" in the *Keysight U2500A Series USB Simultaneous Sampling Multifunction Data Acquisition User's Guide*.

`TRIGger:TYPe?`

This query returns a string value that represents the trigger type selected for input operations.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
mode	Discrete	POST|PRE|MID|DEL	POST

See also

"TRIGger:DCouNT" on page 179

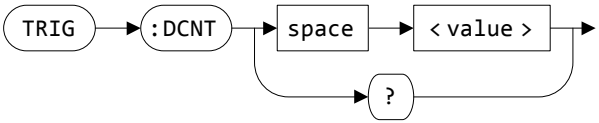
Example

This example illustrates how the **TRIGger:TYPe** command is used to select a delay-trigger acquisition as the input trigger type.

→ `ROUT:ENAB ON,(@101)`	*This command enables channel 101 for input operations.*
→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `ACQ:POIN 100`	*This command sets the instrument to acquire 100 sample points.*
→ `TRIG:SOUR EXTD`	*This command sets the EXTD_AI_TRIG pin as the input triggering source.*
→ `TRIG:DTRG:POL POS`	*This command sets a positive-edge trigger condition for the digital trigger source.*
→ `TRIG:TYP DEL`	*This command sets a delay-trigger acquisition for the analog input.*
→ `TRIG:DCNT 240000000`	*This command sets the delay-trigger acquisition counter value.*
→ `DIG`	*This command starts the single-shot acquisition process once the delay count reaches zero. The delay count starts when a positive-edge digital signal is received from the EXTD_AI_TRIG pin.*

TRIGger:DCouNT

Syntax



`TRIGger:DCouNT <value>`

This command is used to set the delay counter value. When the count reaches zero, the counter stops and the DAQ device starts acquiring data. The count set in the `<value>` parameter will be used when the input trigger type is set to `DEL` (delay-trigger). Set the clock source using the `CONFigure:TIMEbase:SOURce` command.

NOTE

- See "TRIGger:TYPe" on page 176 for more information on selecting an input trigger type.
- See "CONFigure:TIMEbase:SOURce" on page 54 for more information on selecting a timebase/clock source.

`TRIGger:DCouNT?`

This query returns a numeric value that represents the delay count value.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	0 to 2147483647 (31-bits)[a]	0

[a] The delay count parameter is subjected to the following constraints:
Delay count ≥ 0
Delay count < 0x7fffffff

See also

"CONFigure:TIMEbase:SOURce" on page 54

"TRIGger:TYPe" on page 176

Example

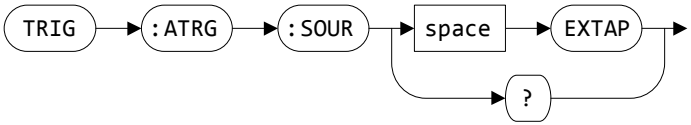
This example illustrates how the **TRIGger:DCouNT** command is used to set a delay counter value of 240000000. This will set a delay time of 5 seconds (240000000/48 MHz).

→ `ROUT:ENAB ON,(@101)`	*This command enables channel 101 for input operations.*
→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `ACQ:POIN 100`	*This command sets the instrument to acquire 100 sample points.*
→ `TRIG:SOUR EXTD`	*This command sets the EXTD_AI_TRIG pin as the input triggering source.*
→ `TRIG:DTRG:POL POS`	*This command sets a positive-edge trigger condition for the digital trigger source.*
→ `TRIG:TYP DEL`	*This command sets a delay-trigger acquisition for the analog input.*
→ `TRIG:DCNT 240000000`	*This command sets the delay-trigger acquisition counter value. This implies that channel 101 will start acquiring data 5 seconds (240000000 / 48 MHz)*[1] *after the trigger condition occurs.*
→ `DIG`	*This command starts the single-shot data acquisition process when all trigger conditions are met.*

[1] Where 48 MHz is the master clock source on board the U2500A Series DAQ device. If you wish to supply your own clock source (with an external clock), see "CONFigure:TIMEbase:SOURce" on page 54 and "CONFigure:TIMEbase:ECLocK" on page 56.

TRIGger:ATRiGger:SOURce

Syntax



`TRIGger:ATRiGger:SOURce <mode>`

This command is used to set the analog trigger source for the AI trigger control.

The only available option is:

EXTAP: Selects the external analog trigger (EXTA_TRIG) pin as the analog triggering source.

NOTE This is an adjunct command for the `TRIGger:SOURce EXTA` command.

`TRIGger:ATRiGger:SOURce?`

This query returns a string value that represents the analog trigger source selected for the AI trigger control.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
mode	Discrete	EXTAP	EXTAP

See also

"TRIGger:SOURce" on page 174

"TRIGger:ATRiGger:CONDition" on page 184

"TRIGger:ATRiGger:HTHReshold" on page 187

"TRIGger:ATRiGger:LTHReshold" on page 189

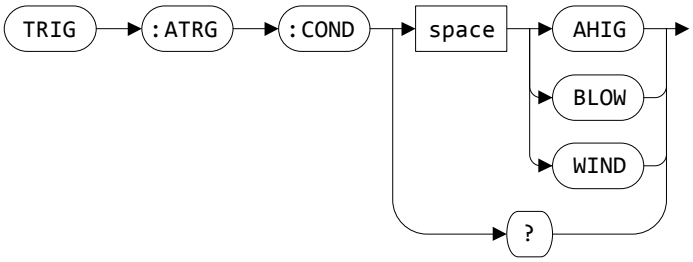
Example

This example illustrates how the **`TRIGger:ATRiGger:SOURce`** command is used to select the external analog trigger (EXTA_TRIG) pin of the DAQ device as the analog input trigger source.

→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `ACQ:POIN 100`	*This command sets the instrument to acquire 100 sample points.*
→ `TRIG:SOUR EXTA`	*This command sets the external analog trigger as the input triggering source.*
→ `TRIG:ATRG:SOUR EXTAP`	*This command sets the EXTA_TRIG pin as the analog trigger control source.*
→ `TRIG:ATRG:COND BLOW`	*This command sets a below-low-level triggering condition.*
→ `TRIG:ATRG:LTHR 5`	*This command sets a 5 V low-threshold.*
→ `TRIG:TYP POST`	*This command set a post-trigger acquisition for the analog input.*
→ `DIG`	*This command starts the single-shot data acquisition process when all trigger conditions are met. If the input signal from the EXTA_TRIG pin falls below 5 V, the analog input will be triggered.*

TRIGger:ATRiGger:CONDition

Syntax



`TRIGger:ATRiGger:CONDition <mode>`

This command is used to set the trigger condition for the AI trigger control.

The valid options are:

- AHIG: Above-High-Level triggering selected. The trigger signal is generated when the analog signal is higher than the high-threshold voltage.
- BLOW: Below-Low-Level triggering selected. The trigger signal is generated when the analog signal is lower than the low-threshold voltage.
- WIND: Window (inside region) triggering selected. The trigger signal is generated when the analog signal falls within the range of the high-threshold and low-threshold voltages.

NOTE Use the `TRIGger:ATRiGger:LTHReshold` and `TRIGger:ATRiGger:HTHReshold` commands to set the low-threshold and high-threshold voltages.

`TRIGger:ATRiGger:CONDition?`

This query returns a string value that represents the trigger condition for the AI trigger control.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
mode	Discrete	AHIG|BLOW|WIND	BLOW

See also

"TRIGger:SOURce" on page 174
"TRIGger:ATRiGger:SOURce" on page 182
"TRIGger:ATRiGger:HTHReshold" on page 187
"TRIGger:ATRiGger:LTHReshold" on page 189

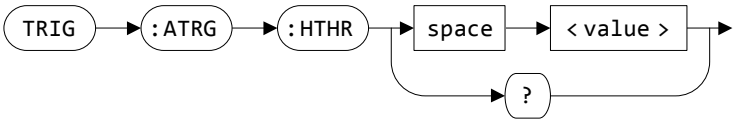
Example

This example illustrates how the `TRIGger:ATRiGger:CONDition` command is used to select a window triggering condition for the AI trigger control.

→ `ROUT:ENAB ON,(@101)`	*This command enables channel 101 for input operations.*
→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `ACQ:POIN 100`	*This command sets the instrument to acquire 100 sample points.*
→ `TRIG:SOUR EXTA`	*This command sets the external analog trigger as the input triggering source.*
→ `TRIG:ATRG:SOUR EXTAP`	*This command sets the EXTA_TRIG pin as the analog trigger control source.*
→ `TRIG:ATRG:COND WIND`	*This command sets a window triggering condition.*
→ `TRIG:ATRG:HTHR 5`	*This command sets a 5 V high-threshold.*
→ `TRIG:ATRG:LTHR 1`	*This command sets a 1 V low-threshold.*
→ `TRIG:TYP POST`	*This command set a post-trigger acquisition for the analog input.*
→ `DIG`	*This command starts the single-shot data acquisition process when all trigger conditions are met. If the input signal from the EXTA_TRIG pin falls between 5 V and 1 V, the analog input will be triggered.*

TRIGger:ATRiGger:HTHReshold

Syntax



`TRIGger:ATRiGger:HTHReshold <value>`

This command sets the high-threshold voltage of the AI trigger control.

`TRIGger:ATRiGger:HTHReshold?`

This query returns a numeric value that represents the high-threshold voltage for the AI trigger control.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	–10 V to 10 V	0 V

Remarks

The high-threshold voltage value must be larger than the low-threshold voltage value.

See also

"TRIGger:SOURce" on page 174

"TRIGger:ATRiGger:SOURce" on page 182

"TRIGger:ATRiGger:CONDition" on page 184

"TRIGger:ATRiGger:LTHReshold" on page 189

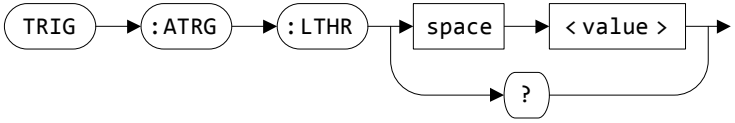
Example

This example illustrates how the `TRIGger:ATRiGger:HTHReshold` command is used to set a 5 V high-threshold for the AI trigger control.

→ `ROUT:ENAB ON,(@101)`	*This command enables channel 101 for input operations.*
→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `ACQ:POIN 100`	*This command sets the instrument to acquire 100 sample points.*
→ `TRIG:SOUR EXTA`	*This command sets the external analog trigger as the input triggering source.*
→ `TRIG:ATRG:SOUR EXTAP`	*This command sets the EXTA_TRIG pin as the analog trigger control source.*
→ `TRIG:ATRG:COND AHIG`	*This command sets a above-high-level triggering condition.*
→ `TRIG:ATRG:HTHR 5`	*This command sets a 5 V high-threshold.*
→ `TRIG:TYP POST`	*This command set a post-trigger acquisition for the analog input.*
→ `DIG`	*This command starts the single-shot data acquisition process when all trigger conditions are met. If the input signal from the EXTA_TRIG pin rises above 5 V, the analog input will be triggered.*

TRIGger:ATRiGger:LTHReshold

Syntax



`TRIGger:ATRiGger:LTHReshold <value>`

This command is used to set the low-threshold voltage of the AI trigger control.

`TRIGger:ATRiGger:LTHReshold?`

This query returns a numeric value that represents the low-threshold voltage for AI trigger control.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	–10 V to 10 V	0 V

Remarks

The high-threshold voltage value must be larger than the low-threshold voltage value.

See also

"TRIGger:SOURce" on page 174

"TRIGger:ATRiGger:SOURce" on page 182

"TRIGger:ATRiGger:CONDition" on page 184

"TRIGger:ATRiGger:HTHReshold" on page 187

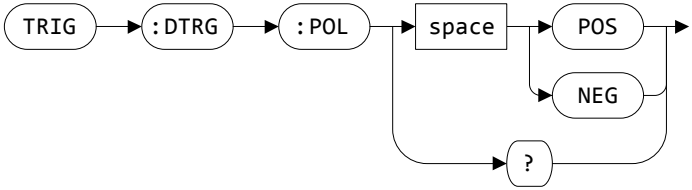
Example

This example illustrates how the `TRIGger:ATRiGger:LTHReshold` command is used to set a 1 V low-threshold for the AI trigger control.

→ `ROUT:ENAB ON,(@101)`	*This command enables channel 101 for input operations.*
→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `ACQ:POIN 100`	*This command sets the instrument to acquire 100 sample points.*
→ `TRIG:SOUR EXTA`	*This command sets the external analog trigger as the input triggering source.*
→ `TRIG:ATRG:SOUR EXTAP`	*This command sets the EXTA_TRIG pin as the analog trigger control source.*
→ `TRIG:ATRG:COND BLOW`	*This command sets a below-low-level triggering condition.*
→ `TRIG:ATRG:LTHR 1`	*This command sets a 1 V low-threshold.*
→ `TRIG:TYP POST`	*This command set a post-trigger acquisition for the analog input.*
→ `DIG`	*This command starts the single-shot data acquisition process when all trigger conditions are met. If the input signal from the EXTA_TRIG pin falls below 1V, the analog input will be triggered.*

TRIGger:DTRiGger:POLarity

Syntax



`TRIGger:DTRiGger:POLarity <mode>`

This command is used to set the polarity of the DIO trigger control.

The valid options are:

- POS (Positive-edge triggering): The trigger signal is generated when a rising edge is detected in the digital signal.
- NEG (Negative-edge triggering): The trigger signal is generated when a falling edge is detected in the digital signal.

`TRIGger:DTRiGger:POLarity?`

This query returns a string value that represents the polarity selected for the DIO trigger control.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
mode	Discrete	POS|NEG	POS

See also

"TRIGger:SOURce" on page 174

"TRIGger:TYPe" on page 176

Example

This example illustrates how the **`TRIGger:DTRiGger:POLarity`** command is used to select a negative-edge trigger condition for the DIO trigger control.

→ `ROUT:ENAB ON,(@101)`	*This command enables channel 101 for input operations.*
→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `ACQ:POIN 100`	*This command sets the instrument to acquire 100 sample points.*
→ `TRIG:SOUR EXTD`	*This command sets the EXTD_AI_TRIG pin as the input triggering source.*
→ `TRIG:DTRG:POL NEG`	*This command sets a negative-edge trigger condition for the digital trigger source.*
→ `TRIG:TYP POST`	*This command set a post-trigger acquisition for the analog input.*
→ `DIG`	*This command starts the single-shot data acquisition process when a negative-edge digital signal is received from the EXTD_AO_TRIG pin.*

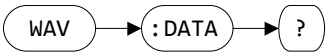
13 WAVeform Subsystem

WAVeform:DATA? 194
WAVeform:POINts 196
WAVeform:STATus? 199
WAVeform:COMPlete? 201

This chapter describes the WAVeform commands used to program the U2500A Series DAQ device over a remote interface. Use the WAVeform:DATA? query to return the raw data acquired by the input channels. You can also set the acquisition count for the continuous acquisition process.

WAVeform:DATA?

Syntax



WAVeform:DATA?

This command is used to return the raw data from the enabled AI channels (as set by the **ROUTe:ENABle** command). The values are returned in the IEEE-488.2 binary block format. See "Using the IEEE-488.2 binary block format" on page 213 for more information.

As an example, a returned value of **#800000200<byte1><byte2>... <byte199><byte200>** indicates that there is a total of 200 bytes of returned data. Each acquisition point is made up of two bytes. The first byte is the least-significant bit (LSB) and the second byte is the most-significant bit (MSB).

The example above returns a total of 100 acquisition points.

Supported models

U2531A, U2541A, U2542A

Remarks

- A maximum of 8 Msa is returned for the single-shot acquisition process and 4 Msa for the continuous acquisition process.
- The length of the header returned is fixed at 10 characters (this includes the characters "#" and "8".

See also

"ACQuire:SRATe" on page 30

"ACQuire:POINts" on page 32

"WAVeform:POINts" on page 196

"DIGitize" on page 204

"RUN" on page 206

"STOP" on page 208

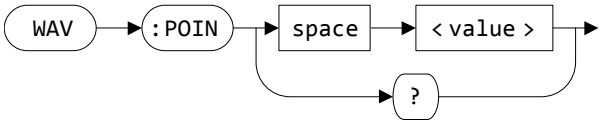
Example

This example illustrates how the **WAVeform:DATA?** query is used to return 1000 sample points of raw data per data block from the instrument buffer.

→ ROUT:ENAB ON,(@101)	*This command enables channel 101 for input operations.*
→ ACQ:SRAT 1000	*This command sets the analog input sampling rate to 1 kHz.*
→ WAV:POIN 1000	*This command sets the instrument to acquire 1000 sample points per data block.*
→ RUN	*This command starts the continuous data acquisition process.*
→ WAV:STAT?	*This query returns the acquisition status of the instrument buffer.*
← DATA	*The returned value is DATA, indicating that at least one block of data is completed and ready to be read back.*
→ WAV:DATA?	*This query returns one block of raw data from AI channel 101.*
← #800002000<byte1><byte2>... <byte2000>	*The first block of data is returned.*
→ WAV:DATA?	
← #800002000<byte1><byte2>... <byte2000>	*The second block of data is returned.*
→ STOP	*This command stops the continuous data acquisition process.*

WAVeform:POINts

Syntax



`WAVeform:POINts <value>`

This command is used to set the number of acquisition points for the continuous acquisition process. Each acquisition point is made up of two bytes. Hence, setting the acquisition points to 100 implies that a block of 200 bytes of raw data is transferred to the PC when the `WAVeform:DATA?` query is issued.

`WAVeform:POINts?`

This query returns a numeric value that represents the number of acquisition points set for the continuous acquisition process.

NOTE The continuous acquisition process is initiated by the `RUN` command. Use the `STOP` command to stop the continuous acquisition process.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	Maximum of 4 Msa	500

Remarks

This command performs the same functions as the **ACQuire:POINts** command. The only difference is that the **ACQuire:POINts** command is used for single-shot acquisitions, while the **WAVeform:POINts** command is used for continuous acquisitions.

See also

"ACQuire:SRATe" on page 30

"ACQuire:POINts" on page 32

"WAVeform:DATA?" on page 194

"RUN" on page 206

"STOP" on page 208

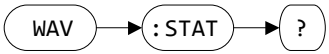
Example

This example illustrates how the **WAVeform:POINts** command is used to acquire 10 sample points of raw data per data block from AI channel 101 and 102.

→ ROUT:ENAB ON,(@101,102)	*This command enables channel 101 and 102 for input operations.*
→ ACQ:SRAT 1000	*This command sets the analog input sampling rate to 1 kHz.*
→ WAV:POIN 10	*This command sets the instrument to acquire 10 sample points per data block.*
→ RUN	*This command starts the continuous acquisition process.*
→ WAV:STAT?	*This query returns the acquisition status of the instrument buffer.*
← DATA	*The returned value is DATA, indicating that at least one block of data is completed and ready to be read back.*
→ WAV:DATA?	*This query returns one block of raw data from both AI channel 101 and channel 102 respectively.*
← #800000040<byte1><byte2>... <byte40>	*The first block of data is returned.*
→ WAV:DATA?	
← #800000040<byte1><byte2>... <byte40>	*The second block of data is returned.*
→ STOP	*This command stops the continuous acquisition process.*

WAVeform:STATus?

Syntax



`WAVeform:STATus?`

This query returns a string value that represents the acquisition status of the instrument buffer.

- EPTY: Empty, indicates that there is no data captured.
- FRAG: Fragment, indicates that the instrument has started to acquire data, but has yet to complete a single block of data.
- DATA: Indicates that at least one block of data is completed and ready to be read back.
- OVER: Indicates that the buffer is full and the acquisition is stopped.

Supported models

U2531A, U2541A, U2542A

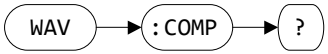
Example

This example illustrates how the `WAVeform:STATus?` query is used to return the acquisition status of the instrument.

→ `ROUT:ENAB ON,(@101:103)`	*This command enables channel 101, 102, and 103 for input operations.*
→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `WAV:POIN 1000`	*This command sets the instrument to acquire 1000 sample points per data block.*
→ `WAV:STAT?`	*This query returns the acquisition status of the instrument buffer.*
← `EPTY`	*The returned value is EPTY, indicating that there is no data captured at the moment.*
→ `RUN`	*This command starts the continuous acquisition process.*
→ `WAV:STAT?`	
← `FRAG`	*The returned value is FRAG, indicating that the acquisition process has begun, but has yet to complete a single data block.*
→ `WAV:STAT?`	
← `DATA`	*The returned value is DATA, indicating that at least one block of data is completed and ready to be read back.*
→ `WAV:DATA?`	*This query returns the completed block of raw data from all three AI channels (101, 102, and 103).*
← `#800006000<byte1><byte2>... <byte6000>`	*The first completed block of data is returned.*
→ `STOP`	*This command stops the continuous acquisition process.*

WAVeform:COMPlete?

Syntax



`WAVeform:COMPlete?`

This query returns a string value that represents the current acquisition status of the instrument.

- YES: Indicates that the acquisition process has been completed (for single-shot acquisition) or stopped.
- NO: Indicates that the instrument is still capturing data.

Supported models

U2531A, U2541A, U2542A

Example

This example illustrates how the `WAVeform:STATus?` query is used to return the acquisition status of the instrument.

→ `ROUT:ENAB ON,(@101)`	*This command enables channel 101 for input operations.*
→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `ACQ:POIN 1000`	*This command sets the instrument to acquire 1000 sample points..*
→ `DIG`	*This command starts the single-shot acquisition process.*
→ `WAV:COMP?`	*This query returns the acquisition status of the instrument.*
← `NO`	*The returned value is NO, indicating that the data is still being captured.*
→ `WAV:COMP?`	
→ `YES`	*The returned value is YES, indicating that the single-shot acquisition process has completed.*
→ `WAV:DATA?`	*This query returns the raw data acquired from AI channel 101.*
← `#800002000<byte1><byte2>...<byte2000>`	*The acquired data block is returned.*

14 Root Commands

DIGitize 204
RUN 206
STOP 208
MODel? 209
SERial? 210
DATA[:USER] 211

This chapter describes the root commands used to program the U2500A Series DAQ device over a remote interface. Use the RUN/DIG command to start a single-shot or continuous acquisition process. You can also send a block of data in the instrument FIFO buffer for AO operations.

DIGitize

Syntax

DIGitize

This command is used to start the single-shot data acquisition process. In single-shot acquisition, a specified number of sample points is acquired and the data is processed once. Use the **ACQuire:POINts** command to specify the number of sample points to be acquired.

Supported models

U2531A, U2541A, U2542A

Remarks

Use the **STOP** command to stop the single-shot acquisition process before the number of points specified in **ACQuire:POINts** is reached.

See also

"ACQuire:SRATe" on page 30

"ACQuire:POINts" on page 32

"WAVeform:DATA?" on page 194

"WAVeform:COMPlete?" on page 201

"STOP" on page 208

Example

This example illustrates how the `DIG` command is used to acquire 1000 sample points of raw data from the AI channel 101.

→ `ROUT:ENAB ON,(@101)`	*This command enables channel 101 for input operations.*
→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `ACQ:POIN 1000`	*This command sets the instrument to acquire 1000 sample points..*
→ `DIG`	*This command starts the single-shot acquisition process.*
→ `WAV:COMP?`	*This query returns the acquisition status of the instrument.*
← `NO`	*The returned value is NO, indicating that the data is still being captured.*
→ `WAV:COMP?`	
→ `YES`	*The returned value is YES, indicating that the single-shot acquisition process has completed.*
→ `WAV:DATA?`	*This query returns the raw data acquired from AI channel 101.*
← `#800002000<byte1><byte2>... <byte1000>`	*The acquired data block is returned.*

RUN

Syntax

RUN

This command is used to start the continuous data acquisition process. In continuous acquisition, data is continuously acquired until a **STOP** command is sent. Use the **WAVeform:POINts** command to specify the number of sample points to be acquired in a single block of data.

Supported models

U2531A, U2541A, U2542A

Remarks

Use the **STOP** command to terminate the continuous acquisition process.

See also

"ACQuire:SRATe" on page 30

"WAVeform:POINts" on page 196

"WAVeform:DATA?" on page 194

"WAVeform:STATus?" on page 199

"STOP" on page 208

Example

This example illustrates how the `RUN` command is used to continuously acquire 200 sample points of raw data from AI channel 101 until the `STOP` command is used.

→ `ROUT:ENAB ON,(@101)`	*This command enables channel 101 for input operations.*
→ `WAV:POIN 200`	*This command sets 200 sample points to be acquired.*
→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `WAV:STAT?`	*This query returns the acquisition status of the instrument.*
← `EPTY`	*The returned value is EPTY, indicating that no data have been captured.*
→ `RUN`	*This command starts the continuous data acquisition process.*
→ `WAV:STAT?`	
← `DATA`	*The returned value is DATA, indicating that at least a block of data is completed and ready to be read back.*
→ `STOP`	*This command stops the continuous data acquisition process.*
→ `WAV:DATA?`	*This query returns the raw data of the AI channels that are enabled by the ROUTe:ENABle command.*
← `#800003990<byte1><byte2>... <byte3990>`	*The total acquired data is returned.*

STOP

Syntax

`STOP`

This command is used to stop the asynchronous analog input operation. Use this command to terminate the continuous acquisition process (initiated by the `RUN` command) or to prematurely stop the single-shot acquisition process (initiated by the `DIG` command).

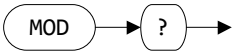
Example

This example illustrates how the `STOP` command is used to stop the continuous acquisition process from capturing anymore data.

→ `ROUT:ENAB ON,(@101)`	*This command enables channel 101 for input operations.*
→ `WAV:POIN 200`	*This command sets 200 sample points to be acquired.*
→ `ACQ:SRAT 1000`	*This command sets the analog input sampling rate to 1 kHz.*
→ `RUN`	*This command starts the continuous data acquisition process.*
→ `WAV:COMP?`	*This query returns the current acquisition status of the continuous acquisition process.*
← `NO`	*The returned value is NO, indicating that data is still being captured.*
→ `STOP`	*This command stops the continuous data acquisition process.*
→ `WAV:COMP?`	
← `YES`	*The returned value is YES, indicating that the continuous acquisition process has completed.*

MODel?

Syntax



`MODel?`

This query returns a string value that represents the model of the connected DAQ device. The returned string can be any of the supported models in the U2500A Series USB simultaneous sampling multifunction data acquisition devices family.

Supported models

U2531A, U2541A, U2542A

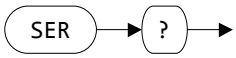
Example

→ `MOD?`	*This query returns the model of the connected DAQ device.*

← `U2531A`

SERial?

Syntax



`SERial?`

This query returns a string value that represents the serial number of the connected DAQ device.

Supported models

U2531A, U2541A, U2542A

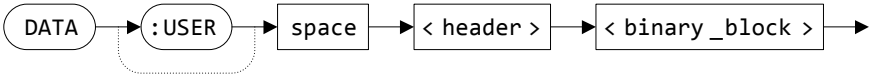
Example

→ `SER?`	*This query returns the serial number of the connected DAQ device.*

← `TW47222003`

DATA[:USER]

Syntax



DATA[:USER] <header><binary_block>

This command is used to set a user-defined pattern onto the instrument FIFO buffer. At least 4 bytes (2 waveform points) of data per channel is required for AO function. If both AO channels are enabled, then a minimum length of 8 bytes (4 waveform points) is necessary.

Supported models

U2531A, U2541A, U2542A

Remarks

- The **<header>** and **<binary_block>** value must be sent in the IEEE-488.2 binary block format. See "Using the IEEE-488.2 binary block format" on page 213 for more information.
- This command cannot be executed while the analog output is enabled (initiated by the **OUTPut ON** command).
- This is an adjunct command for the **APPLy:USER** command.

See also

"APPLy:USER" on page 47

"OUTPut" on page 80

"OUTPut:WAVeform:SRATe" on page 84

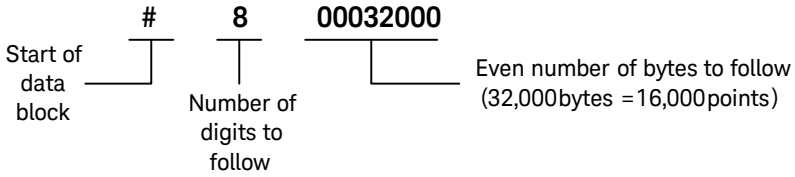
Example

This example illustrates how the `DATA[:USER]` command is used with the `APPLy:USER` command to output an arbitrary waveform via channel 201.

→ `ROUT:ENAB ON,(@201)`	*This command enables output operations for channel 201.*
→ `OUTP:WAV:ITER 0`	*This command causes the instrument to generate the waveform indefinitely.*
→ `OUTP:WAV:SRAT 2000`	*This command sets the analog output waveform sampling rate to 2 kHz.*
→ `DATA #800002000cb0fc50fc30fc20fc30fcb1f...<byte2000>`	*This command sets a user defined pattern to the instrument buffer.*
→ `APPL:USER (@201)`	*This command sends the user-defined pattern to channel 201.*
→ `OUTP ON`	*This command generates the user-defined pattern from channel 201.*

Using the IEEE-488.2 binary block format

In the binary block format, a block header precedes the waveform data. The block header has the following format:



The DAQ device represents binary data as 14-bit (U2531A) or 16-bit (U2541A/42A) integers, which is sent as two bytes. Therefore, the total number of bytes is always twice the number of data points in the waveform (and must always be an even number). Any number of data that is not divisible by two will be truncated. In other words if you input 5-bytes of data, only 4-bytes will be retained.

For example, 32,000 bytes are required to download a waveform with 16,000 points. The least-significant byte (LSB) of each data point is sent first in the byte order for binary transfers in block mode.

The length of the block header is fixed in the U2300A Series DAQ device — 10 characters (including the start of data block character “#” and the number of digits to follow character “8”), 20 bytes in size.

NOTE For more information on the binary block format and the raw data conversion, see “A/D data conversion” in the *Keysight U2500A Series USB Simultaneous Sampling Multifunction Data Acquisition User’s Guide*.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

15 IEEE-488.2 Common Commands

*CLS 216
*ESE 217
*ESR? 219
*IDN? 220
*OPC? 221
*RCL 222
*RST 224
*SAV 225
*SRE 226
*STB? 228
*TST? 229
*WAI 230

This chapter gives an introduction to the IEEE-488.2 common commands. It defines a set of common commands that perform functions such as reset, self-test, and status operations.

*CLS

Syntax

`*CLS`

The `*CLS` (CLear Status) is used to clear the event registers in all register groups. This command also clears the error queue.

Supported models

U2531A, U2541A, U2542A

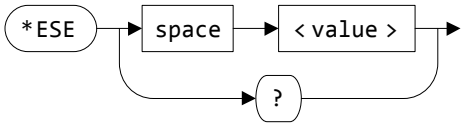
Example

This example illustrates how the `*CLS` command is used to clear the event registers in all register groups and the error queue.

→ `*CLS`	*This command clears the event register bits and the error queue.*

*ESE

Syntax



`*ESE <value>`

The `*ESE` (Event Status Enable) enables bits in the enable register for the Standard Event register group. The selected bits are then reported to bit 5 of the Status Byte register.

Table 15-1 Standard Event register mapping

Bit number	Decimal value	Meaning
0	1	Operation complete
1	Not used	Not used
2	4	Query error
3	8	Device error
4	16	Execution error
5	32	Command error
6	Not used	Not used
7	128	Power-on

`*ESE?`

This query reads the enable register and returns a decimal value that corresponds to the binary-weighted sum of all bits set in the register. For example, if bit 3 (decimal value = 8) and bit 7 (decimal value = 128) are enabled, the query command will return "`+136`".

NOTE For more information on the bit definitions of the Standard Event register and the SCPI status diagram, see "SCPI Status System" on page 25.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
value	Numeric	0 to 255[a]	0

[a] Refer to Table 15-1 for the list of bit definitions for the Standard Event register. To enable more than one bit at a time, use the binary-weighted sum of the intended bits. For example, to set bit 2, 3, and 4, use **28** (4 + 8 + 16) as your value.

Remarks

- An enable register defines which bits in the event register is reported to the Status Byte register group. You can write to or read from an enable register.
- The `*CLS` (clear status) command will not clear the enable register but it will clear all bits in the event register.
- Refer to Figure 1-2 on page 26 for the status system diagram.

See also

"*CLS" on page 216

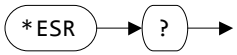
"*ESR?" on page 219

Example

→ `*ESE 16`	*This command enables bit 4 (decimal value = 16) in the enable register. If an execution error occurs, this condition will be reported to the Status Byte register (bit 5 will be set to high).*
→ `*ESE?`	*This query returns which bits are enabled in the register.*
← `+16`	

*ESR?

Syntax



`*ESR?`

The **`*ESR?`** (Event Status Register) query reads the event register for the Standard Event register group and returns a decimal value which corresponds to the binary-weighted sum of all bits set in the event register (Table on page 217). For example, if bit 0 (decimal value = 1) and bit 4 (decimal value = 16) are set (and the corresponding bits are enabled), this query returns the decimal value "**`+17`**".

Supported models

U2531A, U2541A, U2542A

Remarks

- An event register latches the various events from the condition register. There is no buffering in this register; while an event bit is set, subsequent events corresponding to that bit are ignored. This is a read-only register.
- The **`*CLS`** (clear status) command will clear all bits in the event register.
- In order to be reported to the Standard Event register, the corresponding bits in the event register must be enabled using the **`*ESE`** command.
- Refer to Figure 1-2 on page 26 for the status system diagram.

See also

"*CLS" on page 216

"*ESE" on page 217

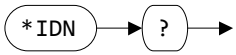
Example

→ **`*ESR?`**	*This command reads the event register (bits 3 and 4 are set).*

← **`+24`**

*IDN?

Syntax



`*IDN?`

This **`*IDN?`** (IDeNtify) query returns the instrument identification string which contains four comma-separated fields. The first field returned is the manufacturer's name, followed by the instrument model number, serial number, and firmware revision code.

```
<Manufacturer>, <Model_no>, <Serial_no>, <A.YYYY.MM.DD>
```

Where, **`YYYY`** represents the year, **`MM`** represents the month, and **`DD`** represents the date of the firmware release.

Supported models

U2531A, U2541A, U2542A

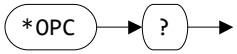
Example

→ `*IDN?`	*This query returns the instrument identification string.*

```
← Keysight Technologies,
  U2531A,TW12345678,
  A.2008.11.04
```

*OPC?

Syntax



`*OPC`

The `*OPC` (OPeration Complete) command sets the “Operation complete” bit (bit 0) in the Standard Event register at the completion of the current operation.

`*OPC?`

This query returns the value “+1” to the output buffer at the completion of the current operation.

NOTE For more information on the bit definitions of the Standard Event register and the SCPI status diagram, see “SCPI Status System” on page 25.

Supported models

U2531A, U2541A, U2542A

Remarks

- The purpose of this command is to synchronize your application with the instrument.
- Note the difference between the `*OPC` command and the `*OPC?` query. The latter returns “1” to the output buffer at the completion of the current operation.

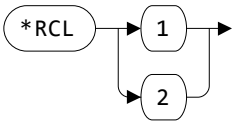
Example

→ `*OPC?`	*This query returns the value “+1” to the output buffer at the completion of the current operation.*

← `1`

*RCL

Syntax



`*RCL {1|2}`

The `*RCL` (ReCaLl) command recalls the instrument state stored in the specified storage location. It is not possible to recall the instrument state from a storage location that is empty. When shipped from the factory, storage locations 1 and 2 are empty.

Use the `*SAV` command to store the current instrument state.

Supported models

U2531A, U2541A, U2542A

Parameter

Item	Type	Range of values	Default value
state	Discrete	1|2	-

Remarks

- The U2300A Series DAQ devices have two storage locations in the nonvolatile memory to store instrument states. You can only recall a state from a location that contains a previously stored state.
- Before recalling a stored state, the DAQ device performs the equivalent of a factory reset (`*RST` command).
- A Factory Reset (`*RST` command) does not affect the configurations stored in the memory. Once a state is stored, it remains until it is overwritten.

See also

"*SAV" on page 225

"*RST" on page 224

Example

→ `*RCL 1`	*This command recalls the instrument state previously stored in location 1.*

*RST

Syntax

`*RST`

The `*RST` (ReSeT) command resets the DAQ device to its factory default state, which is the state when the DAQ device is powered-on for the first time.

NOTE Refer to Chapter 17, "Factory Default Settings" on page 241 for a complete listing of the U2300A Series DAQ device default settings.

Remarks

- The `*RST` command does not clear the event registers and the error queue. To clear the event registers and the error queue, send the `*CLS` command.
- This command does not affect any previously stored instrument states. For more information on saving and recalling previously stored instrument states, see "*SAV" on page 225 and "*RCL" on page 222.

See also

"*CLS" on page 216

"*RCL" on page 222

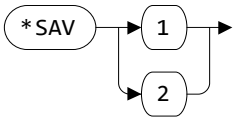
"*SAV" on page 225

Example

→ `*RST`	*This command resets the instrument to its factory default state.*

*SAV

Syntax



`*SAV {1|2}`

The `*SAV` (SAVe) command stores (saves) the current instrument state in the designated storage location. Any state previously stored in the same location is overwritten (with no error generated).

Use the `*RCL` command to recall a stored instrument state.

Parameter

Item	Type	Range of values	Default value
state	Discrete	1|2	-

Remarks

- The U2300A Series DAQ devices have two storage locations in nonvolatile memory to store instrument states. You can store the instrument state in any one of the two locations, but you can only recall a state from a location that contains a previously stored state.
- A Factory Reset (`*RST` command) command does not affect the configurations stored in the memory. Once a state is stored, it remains until it is overwritten.

See also

"*RCL" on page 222

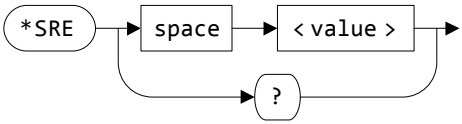
"*RST" on page 224

Example

→ `*SAV 1`	*This command stores the current instrument state in location 1.*

*SRE

Syntax



`*SRE <value>`

The `*SRE` (Service Request Enable) command enables bits in the enable register for the Status Byte register group. Once enabled, the corresponding bits may generate a Request for Service (RQS) in the Status Byte. This RQS event may generate a “call back” to your application as a type of asynchronous interrupt.

Table 15-2 Status Byte register mapping

Bit number	Decimal value	Definition
0	Not used	Not used
1	Not used	Not used
2	4	Error queue
3	Not used	Not used
4	16	Message available
5	32	Standard Event summary
6	64	Master Status summary
7	Not used	Not used

`*SRE?`

This command reads the enable register and returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register. For example, if bit 2 (decimal value = 4) and bit 4 (decimal value = 16) are enabled, the query returns “+20”.

NOTE For more information on the bit definitions of the Status Byte register and the SCPI status diagram, see "SCPI Status System" on page 25.

Parameter

Item	Type	Range of values	Default value
value	Numeric	0 to 255[a]	0

[a] Refer to Table 15-2 for the list of bit definitions for the Status Byte register. To enable more than one bit at a time, use the binary-weighted sum of the intended bits. For example, to set bit 2, 4, and 5, use **52** (4 + 16 + 32) as your value.

Remarks

- An enable register defines which bits in the event register is reported to the Status Byte register group. You can write to or read from an enable register.
- The `*CLS` (clear status) command will not clear the enable register but it does clear all bits in the event register.
- A `*CLS` or `*RST` command does not clear the bits in the Status Byte enable register.
- Refer to Figure 1-2 on page 26 for the status system diagram.

See also

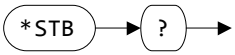
"*STB?" on page 228

Example

→ `*SRE 16`	*This command enables bit 4 (decimal value = 16) in the enable register.*
→ `*SRE?`	*This query returns which bits are enabled in the register.*
← `16`	

*STB?

Syntax



The `*STB?` (STatus Byte) query returns the condition register for the Status Byte register group. This command is similar to a Serial Poll but it is processed like any other instrument command.

NOTE **This is a read-only register and the bits are not cleared when you read the register. For more information on the bit definitions of the Status Byte register and the SCPI status diagram, see "SCPI Status System" on page 25.**

Remarks

- A condition register continuously monitors the state of the instrument. The bits in the condition register are updated in real time and the bits are not latched or buffered.
- This query returns the same results as a Serial Poll but the "Master Status summary" bit (bit 6) is not cleared if a Serial Poll has occurred.
- A Factory Reset (`*RST` command) or power-on cycle clears all bits in the condition register.
- Refer to Figure 1-2 on page 26 for the status system diagram.

See also

"*RST" on page 224

"*SRE" on page 226

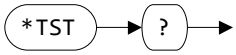
Example

→ `*STB?`	*This command reads the condition register (bits 3 and 5 are set)..*

← `+40`

*TST?

Syntax



`*TST?`

The `*TST?` (TeST) query performs a self-test of the instrument and returns a pass/fail indication. The self-test runs a series of tests and will take a few seconds to complete. If all tests pass, you can have a high confidence that the instrument is operational. The query returns “+0” (all tests passed) or “+1” (one or more tests failed).

Remarks

- If one or more tests fail, “+1” is returned and an error is stored in the error queue. For a complete listing of the error messages related to self-test failures, see Chapter 16, “Error Messages” on page 231.
- If one or more tests fail, see the *Keysight U2500A Series USB Simultaneous Sampling Multifunction Data Acquisition* Service Guide for instructions on obtaining service.
- Following the `*TST?` query, the instrument issues a Factory Reset (*RST command).

See also

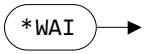
“*RST” on page 224

Example

→ `*TST?`	*This command begins the self-test process and returns a pass/fail indication.*
← `+0`	*The returned value “+0” indicates that the instrument has passed all tests performed.*

*WAI

Syntax



`*WAI`

The **`*WAI`** (WAIt) command configures the instrument output buffer to wait until:

- all pending operations are complete,
- the device clear command is received, or
- power is cycled,

before executing any subsequent commands or queries.

Example

→ `*WAI`	*This command waits until all pending operations are completed.*

16 Error Messages

Error Messages 232

The U2500A Series DAQ device SCPI command errors are summarized in this chapter.

Error Messages

Error messages are created once an erroneous condition is detected.

Errors are retrieved in first-in-first-out (FIFO) order using the `SYSTem:ERRor?` query. The first error returned is the first error that was stored. Reading this error will clear this error allowing the next stored error to be read (if there are other errors stored). Once you have read all of the interface-specific errors, the errors in the global error queue will be retrieved.

If more than 10 errors have occurred, the last error stored in the queue (the most recent error) is replaced with error number: `-350,"Queue overflow"`. No additional errors are stored until you remove the previously stored errors from the queue.

If no errors have occurred when you read the error queue, the instrument responds with the message: `+0,"No error"`.

The interface-specific and global error queues are cleared by the clear status (`*CLS`) command and when the instrument power is cycled. The error queue will not be cleared by a factory reset (`*RST`) command.

See also

"*CLS" on page 216

"*RST" on page 224

"SYSTem:ERRor?" on page 171

SCPI command errors

The following table shows the list of SCPI command errors.

Table 16-1 List of SCPI command errors

Error number	Description
+0	**No error** This is the response to the SYSTem:ERRor? query when there are no errors.
–100	**Command error** Generic syntax error.
–101	**Invalid character** An invalid character was found in the command string. You may have inserted a character such as #, $, or % in a command keyword or within a parameter.
–102	**Syntax error** Invalid syntax was found in the command string. You may have inserted a blank space before or after a colon in the command header, or before a comma.
–103	**Invalid separator** An invalid separator was found in the command string. Check for proper usage of , ; :
–104	**Data type error** The wrong parameter type was found in the command string. You may have specified a number where a string was expected, or vice versa.
–105	**GET not allowed** A Group Execute Trigger (GET) is not allowed within a command string.
–108	**Parameter not allowed** More parameters were received than were expected for the command. You may have entered an extra parameter or added a parameter to a command that does not require a parameter.
–109	**Missing parameter** Fewer parameters were received than were expected for this command. You have omitted one or more parameters that are required for this command.
–110	**Command header error** An error was detected in the header.
–111	**Header separator error** A character that was not a valid header separator was found in the command string.

Table 16-1 List of SCPI command errors (continued)

Error number	Description
–112	**Program mnemonic too long** The header contains more than 12 characters.
–113	**Undefined header** A command was received that is not valid. You may have misspelled the command or it may not be a valid command. If you are using the short form of the command, remember that it may contain up to four letters.
–114	**Header suffix out of range** The value of the numeric suffix is not valid.
–118	**Query not allowed** A query is not supported for this command.
–120	**Numeric data error** Generic numeric data error.
–121	**Invalid character in number** An invalid character was found in the number specified for a parameter value.
–123	**Exponent too large** A numeric parameter was found where its exponent was larger than 32759.
–124	**Too many digits** The mantissa of a numeric parameter contained more than 255 digits, excluding leading zeros.
–128	**Numeric data not allowed** A numeric parameter was received but a character string was expected.
–130	**Suffix error** Generic suffix error.
–131	**Invalid suffix** A suffix was incorrectly specified for a numeric parameter.
–134	**Suffix too long** The suffix contains more than 12 characters.
–138	**Suffix not allowed** A suffix is not supported for this command.
–140	**Character data error** Generic character data error.

Table 16-1 List of SCPI command errors (continued)

Error number	Description
–141	**Invalid character data** Either the character data element contains an invalid character, or the element is not valid.
–144	**Character data too long** The character data element contains more than 12 characters.
–148	**Character data not allowed** A discrete parameter was received, but a string or numeric parameter was expected.
–150	**String data error** Generic string data error.
–151	**Invalid string data** An invalid character string was received. Check that the string is enclosed in quotation marks.
–158	**String data not allowed** A character string was received, but is not allowed for this command.
–160	**Block data error** Generic block data error.
–161	**Invalid block data** The number of data bytes sent does not match the number of bytes specified in the header.
–168	**Block data not allowed** Data was sent in the arbitrary block format but is not allowed for this command.
–170	**Expression error** The U2500A Series devices does not accept mathematical expressions.
–171	**Invalid expression** The U2500A Series devices does not accept mathematical expressions.
–178	**Expression data not allowed** The U2500A Series devices does not accept mathematical expressions.

SCPI execution errors

The following table shows the list of SCPI execution errors.

Table 16-2 List of SCPI execution errors

Error number	Description
–200	**Execution error** Generic syntax error.
–220	**Parameter error** A data element related error occurred.
–221	**Settings conflict** A data element could not be executed because of the present instrument state.
–221	**Settings conflict** The amplitude and offset are out of the reference voltage range.
–221	**Settings conflict** The high threshold is lower than low threshold.
–221	**Settings conflict** The analog trigger level is beyond range because of the analog trigger source.
–221	**Settings conflict** The sampling rate is beyond range because of the number of channels.
–221	**Settings conflict** The acquisition points is beyond range because of the number of channels.
–221	**Settings conflict** The waveform points is beyond range because of the number of channels.
–221	**Settings conflict** Unsupported trigger mode.
–221	**Settings conflict.** Unsupported trigger mode because of the analog trigger source.
–222	**Data out of range** External clock is set above instrument's capability.

Table 16-2 List of SCPI execution errors (continued)

Error number	Description
–223	**Too much data** An Arbitrary waveform was specified that contains more than 65536 waveform points.
–224	**Illegal parameter value** An exact value was expected or an invalid discrete parameter was received.

SCPI device-specific errors

The following table shows the list of SCPI device-specific errors.

Table 16-3 List of SCPI device-specific errors

Error number	Description
–316	**Cal error** A calibration error not defined in greater detail has occured.
–330	**Self-test failed** The U2500A Series DAQ device complete self-test failed from the remote interface (*TST? command). This indicates a problem with the device that is not covered by a specific error message. The device may require service.
–350	**Queue overflow** The error queue is full because more than 20 errors have occurred. No additional errors are stored until you remove the errors from the queue.

SCPI query errors

The following table shows the list of SCPI query errors.

Table 16-4 List of SCPI query errors

Error number	Description
–400	**Query error** Generic error query.
–410	**Query INTERRUPTED** A condition causing an interrupted query error occurred.
–420	**Query UNTERMINATED** A condition causing an unterminated query error occurred.
–430	**Query DEADLOCKED** A condition causing a deadlocked query error occurred.
–440	**Query UNTERMINATED after indefinite response** A query was received in the same program message after a query indicating an indefinite response was executed.

SCPI instrument errors

The following table shows the list of SCPI instrument errors.

Table 16-5 List of SCPI instrument errors

Error number	Description
112	Channel list: channel number out of range
113	Channel list: empty scan list
222	Settings conflict: module type does not match state
223	Settings conflict: trig source changed to IMM
261	Not able to execute while scan initiated
262	Not able to abort scan
263	Not able to execute while instrument is measuring

Table 16-5 List of SCPI instrument errors (continued)

Error number	Description
264	Not a scannable channel
281	Not able to perform on more than one channel
301	Module currently committed to scan
303	Module is not able to perform requested operation
304	Does not exist
305	Not able to perform requested operation
305	Not able to perform requested operation; cannot generate user-defined and pre-defined waveforms at once
305	Not able to perform requested operation; output is running
305	Not able to perform requested operation; output has stopped
305	Not able to perform requested operation; function must be enabled first
305	Not able to perform requested operation; user-defined waveform not set
305	Not able to perform requested operation; sampling rate cannot be 0 with user-defined output
307	Incorrectly configured ref channel
308	Channel not able to perform requested operation
308	Channel not able to perform requested operation: currently in differential mode
309	Incorrectly formatted channel list
311	Not able to specify resolution with Auto range
521	Input buffer overflow
522	Output buffer overflow
531	Insufficient memory
532	Not able to achieve requested resolution

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

17 Factory Default Settings

Factory Default Settings 242
Command Quick Reference 246

The U2500A Series DAQ device SCPI default settings are summarized in this chapter.

Factory Default Settings

The settings in Table 17-1 are used when the U2500A Series DAQ device is powered-on. They are also set once the `*RST` command is sent to the U2500A Series DAQ device.

See also

"*CLS" on page 216

"*RST" on page 224

"SYSTem:ERRor?" on page 171

The following table shows the list of the factory default settings.

Table 17-1 List of factory default settings

Command	Setting	Remarks
ACQuire:SRATe	1000 Hz	AI sampling rate
ACQuire:POINts	500	Single-shot acquisition points
APPLy	SINusoid	AO function
	10 V	Waveform amplitude
	0 V	DC offset
CONFigure:DIGital:DIRection	INPut	Input operations enabled
CONFigure:TIMEbase:SOURce	INTernal	Timebase/clock source
CONFigure:TIMEbase:ECLocK	10000 kHz	External clock frequency
CONFigure:SSI	NONE	Synchronization is disabled
OUTPut	0	Output is disabled
OUTPut:WAVeform:ITERate	0	Infinite iteration
OUTPut:WAVeform:SRATe	0 Hz	Automatic sampling
OUTPut:WAVeform:FREQuency	4000 Hz	AO frequency
OUTPut:TRIGger:SOURce	NONE	Immediate triggering
OUTPut:TRIGger:TYPe	POST	AO trigger type
TRIGger:DCouNT	0	AO trigger delay
OUTPut:TRIGger:ATRiGger:SOURce	EXTAP	AO trigger control source
OUTPut:TRIGger:ATRiGger:CONDition	BLOW	AO trigger control condition
OUTPut:TRIGger:ATRiGger:HTHReshold	0 V	AO trigger control high-threshold
OUTPut:TRIGger:ATRiGger:LTHReshold	0 V	AO trigger control low-threshold
OUTPut:TRIGger:DTRiGger:POLarity	POS	DIO trigger control polarity
ROUTe:CHANnel:RANGe	10 V	AI range
ROUTe:CHANnel:POLarity	BIPolar	AI polarity
ROUTe:CHANnel:RSouRCe	INTernal	AO reference voltage source

Table 17-1 List of factory default settings (continued)

Command	Setting	Remarks
ROUTe:CHANnel:RVOLtage	10 V	AO reference voltage
ROUTe:ENABle	0	AO channels disabled
[SENSe:]VOLTage:RANGe	AUTO	AI range (polling mode)
[SENSe:]VOLTage:POLarity	BIPolar	AI polarity (polling mode)
[SENSe:]VOLTage:AVERage	1	Number of measurement points to average
[SENSe:]COUNter:FUNCtion	TOTalize	Counter function
[SENSe:]COUNter:GATE:SOURce	INTernal	Counter gate source
[SENSe:]COUNter:GATE:POLarity	AHI	Counter gate polarity
[SENSe:]COUNter:GATE:CONTrol	DIS	Counter gate control
[SENSe:]COUNter:CLocK:SOURce	INTernal	Counter clock source
[SENSe:]COUNter:CLocK:EXTernal	0	Counter clock external frequency
[SENSe:]COUNter:CLocK:POLarity	AHI	Counter clock polarity
[SENSe:]COUNter:TOTalize:IVALue	0	Totalizer measurement initial value
[SENSe:]COUNter:TOTalize:UDOWn:SOURce	INTernal	Totalizer measurement up/down source
[SENSe:]COUNter:TOTalize:UDOWn:DIRection	UP	Totalizer measurement up/down direction
SOURce:VOLTage[:LEVel]	0 V	AO supply voltage (single voltage output)
SOURce:VOLTage:POLarity	BIPolar	AO polarity (single voltage output)
SOURce:VOLTage:RSouRCe	INTernal	AO reference voltage source (single voltage output)
SOURce:VOLTage:RVOLtage	10 V	AO reference voltage (single voltage output)
TRIGger:SOURce	NONE	Immediate triggering
TRIGger:TYPe	POST	AI trigger type
TRIGger:DCouNT	0	AI trigger delay
TRIGger:ATRiGger:SOURce	EXTAP	AI trigger control source
TRIGger:ATRiGger:CONDition	BLOW	AI trigger control condition
TRIGger:ATRiGger:HTHReshold	0 V	AI trigger control high-threshold
TRIGger:ATRiGger:LTHReshold	0 V	AI trigger control low-threshold

Table 17-1 List of factory default settings (continued)

Command	Setting	Remarks
TRIGger:DTRiGger:POLarity	POS	DIO trigger control polarity
WAVeform:POINts	500	Continuous acquisition points
*ESE	0	Standard Event enable register
*SRE	0	Status Byte enable register

Command Quick Reference

Root keyword	Second-level	Third-level	Fourth-level	Fifth-level	Page
ACQuire	:SRATe				page 30
	:SRATe?				page 30
	:POINts				page 32
	:POINts?				page 32
APPLy?					page 36
APPLy	:SINusoid				page 37
	:SQUare				page 39
	:SAWTooth				page 41
	:TRIangle				page 43
	:NOISe				page 45
	:USER				page 47
CALibration	:BEGin				page 50
CONFigure	:DIGital	:DIRection			page 52
		:DIRection?			page 52
	:TIMEbase	:SOURce			page 54
		:SOURce?			page 54
		:ECLocK			page 56
		:ECLocK?			page 56
	:SSI				page 58
	:SSI?				page 58
MEASure	[:VOLTage]	[:DC]?			page 62
	:COUNter	:DATA?			page 64
MEASure	:COUNter	:FREQuency?			page 66
		:PERiod?			page 68
		:PWIDth?			page 70

Root keyword	Second-level	Third-level	Fourth-level	Fifth-level	Page
		:TOTalize?			page 72
	:DIGital?				page 74
	:DIGital	:BIT?			page 76
OUTPut?					page 80
OUTPut					page 80
	:WAVeform	:ITERate			page 82
		:ITERate?			page 82
		:SRATe			page 84
		:SRATe?			page 84
		:FREQuency			page 87
		:FREQuency?			page 87
	:TRIGger	:SOURce			page 89
		:SOURce?			page 89
		:TYPe			page 91
		:TYPe?			page 91
		:DCouNT			page 93
		:DCouNT?			page 93
		:ATRiGger	:SOURce		page 95
			:SOURce?		page 95
			:CONDition		page 97
			:CONDition?		page 97
			:HTHReshold		page 100
			:HTHReshold?		page 100
OUTPut	:TRIGger	:ATRiGger	:LTHReshold		page 102
			:LTHReshold?		page 102
		:DTRiGger	:POLarity		page 104
			:POLarity?		page 104

Root keyword	Second-level	Third-level	Fourth-level	Fifth-level	Page
ROUTe	:CHANnel	:RANGe			page 108
		:RANGe?			page 108
		:POLarity			page 110
		:POLarity?			page 110
		:STYPe?			page 112
		:RSouRCe			page 113
		:RSouRCe?			page 113
		:RVOLtage			page 115
		:RVOLtage?			page 115
	:ENABle				page 117
	:ENABle?				page 117
[SENSe:]	VOLTage	:RANGe			page 120
		:RANGe?			page 120
		:POLarity			page 122
		:POLarity?			page 122
		:STYPe?			page 124
		:AVERage			page 125
		:AVERage?			page 125
[SENSe:]	COUNter	:FUNCtion			page 127
		:FUNCtion?			page 127
		:ABORt			page 129
		:GATE	:SOURce		page 131
			:SOURce?		page 131
			:POLarity		page 133
			:POLarity?		page 133
			:CONTrol		page 135
			:CONTrol?		page 135

Root keyword	Second-level	Third-level	Fourth-level	Fifth-level	Page
		:CLocK	:SOURce		page 137
			:SOURce?		page 137
			:INTernal?		page 139
			:EXTernal		page 140
			:EXTernal?		page 140
			:POLarity		page 142
			:POLarity?		page 142
		:TOTalize	:IVALue		page 144
			:IVALue?		page 144
			:INITiate		page 146
			:CLEar		page 147
			:UDOWn	:SOURce	page 149
				:SOURce?	page 149
				:DIRection	page 152
				:DIRection?	page 152
SOURce	:VOLTage	[:LEVel]			page 156
		[:LEVel]?			page 156
		:POLarity			page 158
		:POLarity?			page 158
SOURce	:VOLTage	:RSouRCe			page 160
		:RSouRCe?			page 160
		:RVOLtage			page 162
		:RVOLtage?			page 162
	:DIGital	:DATA			page 164
		:DATA?			page 164
			:BIT		page 166
			:BIT?		page 166

Root keyword	Second-level	Third-level	Fourth-level	Fifth-level	Page
SYSTem	:CDEScription?				page 170
	:ERRor?				page 171
TRIGger	:SOURce				page 174
	:SOURce?				page 174
	:TYPe				page 176
	:TYPe?				page 177
	:DCouNT				page 179
	:DCouNT?				page 179
	:ATRiGger	:SOURce			page 182
		:SOURce?			page 182
		:CONDition			page 184
		:CONDition?			page 184
		:HTHReshold			page 187
		:HTHReshold?			page 187
		:LTHReshold			page 189
		:LTHReshold?			page 189
	:DTRiGger	:POLarity			page 191
		:POLarity?			page 191
WAVeform	:DATA?				page 194
	:POINts				page 196
	:STATus?				page 199
	:COMPlete?				page 201
DIGitize					page 204
RUN					page 206
STOP					page 208
MODel?					page 209
SERial?					page 210

Root keyword	Second-level	Third-level	Fourth-level	Fifth-level	Page
DATA	[:USER]				page 211
*CLS					page 216
*ESE					page 217
*ESE?					page 217
*ESR?					page 219
*OPC?					page 221
*RCL					page 222
*RST					page 224
*SAV					page 225
*SRE					page 226
*SRE?					page 226
*STB?					page 228
*TST					page 229
*WAI					page 230

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

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

© Keysight Technologies 2010 - 2017
Edition 5, June 1, 2017

Printed in Malaysia

U2541-90014
www.keysight.com