

# **Agilent U2722A/U2723A USB Modular Source Measure Units**

## **Programmer's Reference**



**Agilent Technologies**

# Notices

© Agilent Technologies, Inc. 2008 - 2009

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 Agilent Technologies, Inc. as governed by United States and international copyright laws.

## Manual Part Number

U2722-90022

## Edition

Second Edition, May 22, 2009

Printed in Malaysia

Agilent Technologies, Inc.  
5301 Stevens Creek Blvd.  
Santa Clara, CA 95052 USA

## Warranty

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

## Technology Licenses

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

## Restricted Rights Legend

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

## Safety Notices

### CAUTION

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

---

### WARNING

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

---

# Table of Contents

- 1 Introduction to Programming**
  - Introduction to the SCPI Language 2
  - SCPI Conventions and Data Formats 3
    - Command Separators 5
    - Querying Parameter Settings 5
    - SCPI Command Terminators 6
    - IEEE-488.2 Common Commands 6
    - Address/Channel List Parameter 7
- 2 SCPI Status Registers**
  - SCPI Status Registers 10
    - Operation Status Registers 12
    - Questionable Status Registers 12
    - Standard Event Register 13
    - Status Byte Register 14
- 3 ABORt Subsystem**
  - ABORt:TRANsient 18
- 4 CONFigure Subsystem**
  - CONFigure:SSI 20
- 5 INITiate Subsystem**
  - INITiate[:IMMediate]:TRANsient 24
- 6 MEASure Subsystem**
  - MEASure[:SCALar]:CURRent[:DC]? 26
  - MEASure[:SCALar]:VOLTage[:DC]? 27

Table of Contents

MEASure:ARRay:CURRent[:DC]?	28
MEASure:ARRay:VOLTage[:DC]?	29
MEASure:TEMPerature?	30
<b>7   OUTPut Subsystem</b>	
OUTPut[:STATe]	32
<b>8   SENSe Subsystem</b>	
SENSe:CURRent[:DC]:APERture?	34
SENSe:CURRent[:DC]:NPLCycles	35
SENSe:SWEEp:POINts	37
SENSe:SWEEp:TINTerval	39
SENSe:VOLTage[:DC]:APERture?	40
SENSe:VOLTage[:DC]:NPLCycles	41
<b>9   SOURce Subsystem</b>	
[SOURce:]CURRent	44
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]	46
[SOURce:]CURRent:LIMit	48
[SOURce:]CURRent:RANGe	49
[SOURce:]MEMory:ARM	51
[SOURce:]MEMory:CONFigure:POINts	52
[SOURce:]MEMory:CURRent:LIMit	54
[SOURce:]MEMory:CURRent:MEASure	55
[SOURce:]MEMory:CURRent:SOURce	56
[SOURce:]MEMory:CURRent:RANGe	57
[SOURce:]MEMory:LIST	58

[SOURce:]MEMory:LIST:CLEar	60
[SOURce:]MEMory:LIST:POINts?	61
[SOURce:]MEMory:LIST:LOAD	62
[SOURce:]MEMory:LIST:READ?	63
[SOURce:]MEMory:LIST:STORe	64
[SOURce:]MEMory:OUTPut	65
[SOURce:]MEMory:LIST:DATA?	66
[SOURce:]MEMory:SOURce:DELay:AUTO	68
[SOURce:]MEMory:SOURce:DELay	69
[SOURce:]MEMory:TRIGger	70
[SOURce:]MEMory:VOLTage:LIMit	71
[SOURce:]MEMory:VOLTage:MEASure	72
[SOURce:]MEMory:VOLTage:SOURce	73
[SOURce:]MEMory:VOLTage:RANGe	74
[SOURce:]VOLTage	75
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]	77
[SOURce:]VOLTage:LIMit	79
[SOURce:]VOLTage:RANGe	80
<b>10 STATus Subsystem</b>	
STATus:OPERation:CONDition?	82
STATus:OPERation:ENABLE	84
STATus:OPERation[:EVENT]?	87
STATus:OPERation:NTRansition	89
STATus:OPERation:PTRansition	91

## Table of Contents

	STATUS:PRESet	93
	STATUS:QUESTionable:CONDition?	94
	STATUS:QUESTionable:ENABLE	95
	STATUS:QUESTionable[:EVENT]?	97
	STATUS:QUESTionable:NTRansition	99
	STATUS:QUESTionable:PTRansition	101
<b>11</b>	<b>SYSTem Subsystem</b>	
	SYSTem:CDEscription?	104
	SYSTem:CHANnel[:COUNT]?	105
	SYSTem:ERRor?	106
	SYSTem:LFRequency	107
	SYSTem:VERSion?	108
<b>12</b>	<b>TRIGger Subsystem</b>	
	TRIGger:SOURce	110
<b>13</b>	<b>IEEE-488.2 Common Commands</b>	
	*CAL?	112
	*CLS	113
	*ESE/*ESE?	114
	*ESR?	116
	*IDN?	118
	*OPC/OPC?	119
	*RST	120
	*SRE/SRE?	121
	*STB?	124

\*TST? 126

\*WAI 127

14 Error Messages

Error Messages 130

Error List 130

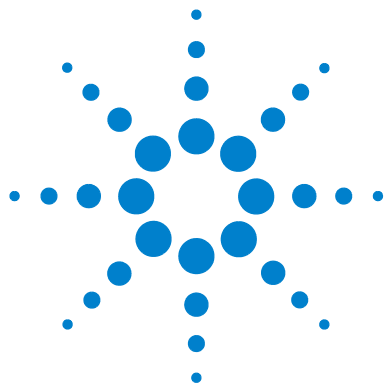
15 Default Factory Settings

Factory Default Settings 138

D  
R  
A  
F  
T  
  
I  
N  
D  
I  
C  
A  
T  
O  
R

DRAFT INDICATOR





# 1

## Introduction to Programming

Introduction to the SCPI Language	2
SCPI Conventions and Data Formats	3
Command Separators	5
Querying Parameter Settings	5
SCPI Command Terminators	6
IEEE-488.2 Common Commands	6
Address/Channel List Parameter	7

This chapter introduces the remote programming basics of the U2722A/U2723A USB modular source measure unit. The 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 are based on a hierarchical structure, also known as a tree system. In this system, associated commands are grouped together under a common node or root, thus forming subsystems. A portion of the SOURce subsystem is shown below to illustrate the tree system.

```
SOURce:
    VOLTage:
        RANGE <CPD>, (<(@1) | (@2) | (@3)>)
        RANGE? (<(@1) | (@2) | (@3)>)

    CURRent:
        LIMit <CPD>, (<(@1) | (@2) | (@3)>)
        LIMit? (<(@1) | (@2) | (@3)>)
```

SOURce is the root keyword of the command, VOLTage and CURRent are second-level keywords, and RANGE and LIMit are third-level keywords. A colon ( : ) separates a command keyword from a lower-level keyword.

For example, in the above syntax statement, VOLT and VOLTAGE are both acceptable forms. You can use upper- or lower-case letters. Therefore, VOLTAGE, volt, and Volt are all acceptable. Other forms, such as VOL and VOLTAG, are not valid and will generate an error.

## SCPI Conventions and Data Formats

The following SCPI conventions are used throughout this guide.

<b>Angle brackets</b> < >	Items within angle brackets are parameter abbreviations. For example, <NR1> indicates a specific form of numerical data.
<b>Vertical bar</b> 	Vertical bars separate alternative parameters. For example, VOLT CURR indicates that either "VOLT" or "CURR" can be used as a parameter.
<b>Square brackets</b> [ ]	Items within square brackets are optional. The representation [SOURce:]VOLTage means that SOURce: may be omitted.
<b>Parenthesis</b> ( )	Items within parentheses are used in place of the usual parameter types to specify a channel list. The notation (@1:3) specifies a channel list that consists of channels 1, 2, and 3. The notation (@1,3) specifies a channel list that includes only channels 1 and 3.
<b>Braces</b> { }	Braces indicate parameters that may be repeated zero or more times. It is used especially for showing arrays. The notation <A>{,<B>} shows that parameter "A" must be entered, while parameter "B" may be omitted or may be entered one or more times.

Data programmed or queried from the instrument is ASCII. The data may be numerical or character string.

<b>&lt;NR1&gt;</b>	Digits with an implied decimal point assumed at the right of the least-significant digit. Example: 273
<b>&lt;NR2&gt;</b>	Digits with an explicit decimal point. Example: 27.3
<b>&lt;NR3&gt;</b>	Digits with an explicit decimal point and an exponent. Example: 2.73E+02
<b>&lt;NRf&gt;</b>	Extended format that includes <NR1>, <NR2>, and <NR3>. Examples: 273 27.3 2.73E+02
<b>&lt;NRf+&gt;</b>	Expanded decimal format that includes <NRf> and MIN, MAX. Examples: 273 27.3 2.73E+02 MAX  MIN and MAX are the minimum and maximum limit values that are implicit in the range specification for the parameter.
<b>&lt;Bool&gt;</b>	Boolean Data. Can be numeric (0, 1), or named (OFF, ON).
<b>&lt;SPD&gt;</b>	String Program Data. Programs string parameters enclosed in single or double quotes.
<b>&lt;CPD&gt;</b>	Character Program Data. Programs discrete parameters. Accepts both the short form and long form.
<b>&lt;SRD&gt;</b>	String Response Data. Returns string parameters enclosed in single or double quotes.
<b>&lt;CRD&gt;</b>	Character Response Data. Returns discrete parameters. Only the short form of the parameter is returned.
<b>&lt;AARD&gt;</b>	Arbitrary ASCII Response Data. Permits the return of an undelimited 7-bit ASCII. This data type has an implied message terminator.
<b>&lt;Block&gt;</b>	Arbitrary Block Response Data. Permits the return of definite length and indefinite length arbitrary response data. This data type has an implied message terminator.

## Command Separators

A colon ( : ) is used to separate a command keyword from a lower-level keyword. You must insert a blank space to separate a parameter from a command keyword. If a command requires more than one parameter, you must separate adjacent parameters using a comma as shown below.

```
SOUR:VOLT:RANG R2V, (@1)
```

A semicolon ( ; ) is used to separate commands within the same subsystem, and can also minimize typing. For example, sending the following command string.

```
SENS:SWE:POIN 1000, (@1); TINT 10, (@1)
```

... is the same as sending the following two commands.

```
SENS:SWE:POIN 1000, (@1)
```

```
SENS:SWE:TINT 10, (@1)
```

Use a colon and semicolon to link commands from different subsystems. For example, in the following command string, an error is generated if you do not use both the colon and semicolon.

```
SOUR:VOLT:RANG R2V, (@1);:SENS:SWE:POIN 1000, (@1)
```

### NOTE

Colons and semicolons can be used to link multiple commands together, from the same subsystem or from different subsystems, in a single command string. However, you are limited to a maximum of 3000 characters (combined) for each command string.

## Querying Parameter Settings

You can query the current value of most parameters by adding a question mark ( ? ) to the command. For example, the following command sets the current range to 10 mA for output 1.

```
SOUR:CURRE:RANG R10mA, (@1)
```

You can then query the current range setting by sending:

```
SOUR:CURRE:RANG? (@1)
```

Typical Response: R10mA

## SCPI Command Terminators

A command string sent to the instrument must terminate with a <new line> (<NL>) character. The IEEE-488 End-Of-Identify (EOI) message is interpreted as a <NL> character and can be used to terminate a command string in place of a <NL> character. A <carriage return> followed by a <NL> is also accepted. Command string termination will always reset the current SCPI command path to the root level.

## IEEE-488.2 Common Commands

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

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

## Address/Channel List Parameter

There are two ways to specify the U2722A/U2723A address or channel list parameter. To specify a single channel, use the following syntax.

(@<ch>)

For example, CURR 0.02, (@1), where (@1) specifies channel 1.

To specify a single address, use the following syntax.

(@<address>)

For example, CONF:SSI SLAV, (@1), where (@1) specifies address 1.

To specify more than one address, the following syntax must be used.

(@<address>[, <address>])

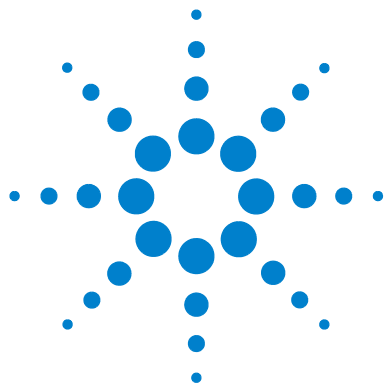
### NOTE

- The command CONFfigure:SSI is the only U2722A/U2723A command that uses the range of address list.
- When adding a channel list parameter to a query, you must include a space character between the query indicator (?) and channel list parameter. Otherwise error **–103**, **"Invalid separator"** will occur.

## **1 Introduction to Programming**

### SCPI Conventions and Data Formats





## 2 SCPI Status Registers

SCPI Status Registers	10
Operation Status Registers	12
Questionable Status Registers	12
Standard Event Register	13
Status Byte Register	14

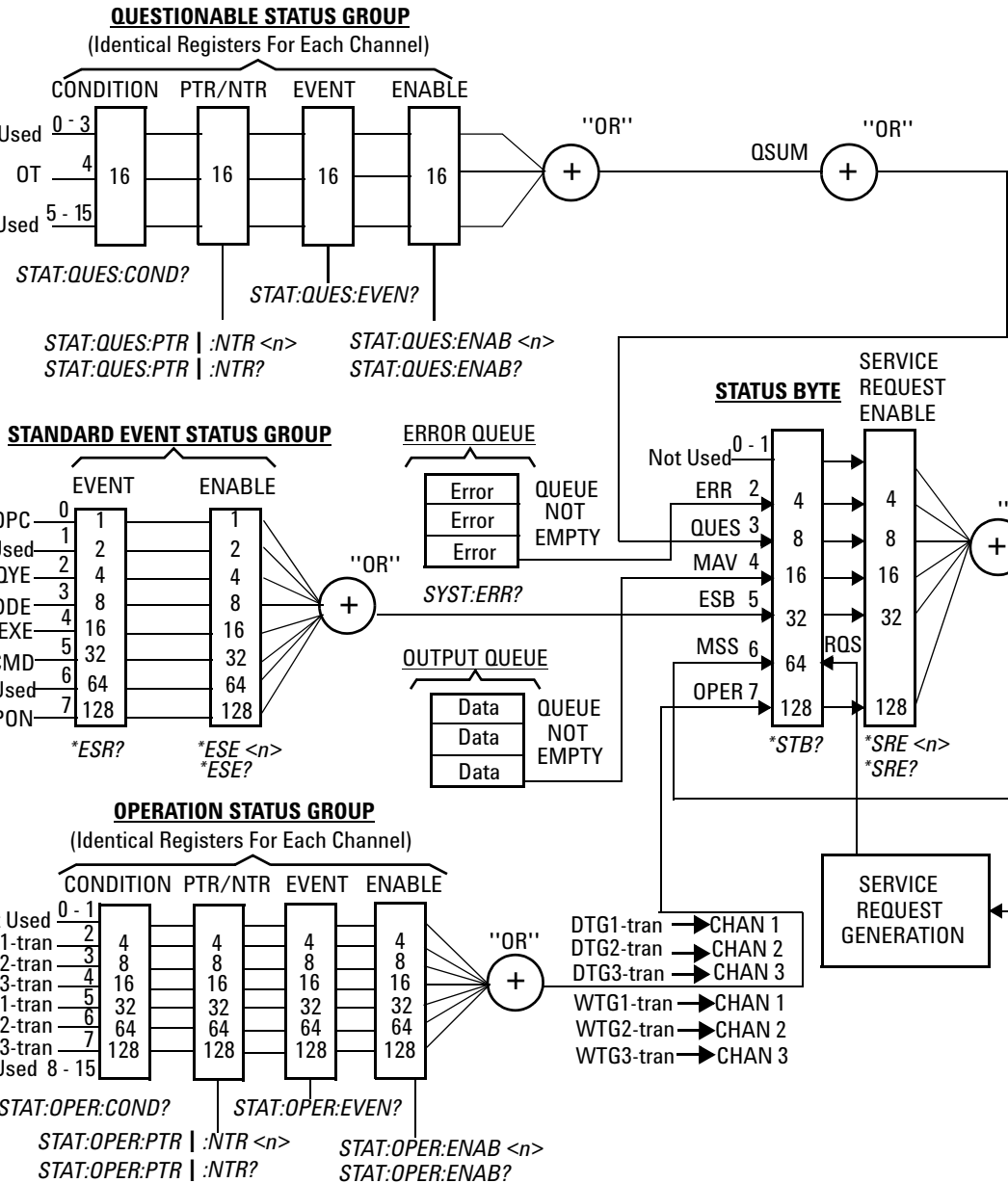
This chapter explains the SCPI status registers that record various instrument conditions of the U2722A/U2723A USB modular source measure unit.



## SCPI Status Registers

Status register programming lets you determine the operating condition of the instrument at any time. The instrument has three groups of status registers; Operation, Questionable, and Standard Event. The Operation and Questionable Status groups each consists of Condition, Enable, Event registers, and Negative-Transition (NTR) and Positive-Transition (PTR) filters. The Status subsystem is also programmed using Common commands. Common commands control additional status functions such as the Service Request Enable and the Status Byte registers.

The U2722A/U2723A uses the Operation, Questionable, Standard Event, and Status Byte register groups to record a variety of instrument conditions. The relationship between various registers in the U2722A/U2723A SCPI status system is shown in [Figure 2-1](#).



### Figure 2-1 Status system diagram

## Operation Status Registers

These registers record signals that occur during normal operation. The group consists of a Condition, PTR/NTR, Event, and Enable register. The outputs of the Operation Status register group are logically-ORed into the OPERation summary bit (7) of the Status Byte register.

### Bit Definitions: Operation Status Registers

Bit number	Decimal value	Definition
0 to 1 Not Used	0	0 is returned.
2 DTG1-tran	4	The transient system has received a trigger signal and operation is running in channel 1
3 DTG2-tran	8	The transient system has received a trigger signal and operation is running in channel 2
4 DTG3-tran	16	The transient system has received a trigger signal and operation is running in channel 3
5 WTG1-tran	32	The transient system is waiting for a trigger in channel 1.
6 WTG2-tran	64	The transient system is waiting for a trigger in channel 2.
7 WTG3-tran	128	The transient system is waiting for a trigger in channel 3.
8 to 15 Not Used	0	0 is returned.

All bits in the Operation Enable register will be cleared when you execute the [STATus:PRESet](#) command. This command presets the Operation Status registers.

## Questionable Status Registers

These registers record signals that indicate abnormal operation. The group consists of the same register types as the Operation Status group. The outputs of the Questionable Status group are logically-ORed into the QUEStionable summary bit (3) of the Status Byte register.

### Bit Definitions: Questionable Status Registers

Bit number	Decimal value	Definition
0 to 3 Not Used	0	0 is returned.
4 OT	16	The over-temperature protection has tripped.
5 to 15 Not Used	0	0 is returned.

- Once a bit is set, it remains set until cleared by reading the event register or the clear status (\*CLS) command.
- The [STATus:PRESet](#) command will clear all bits in the enable register. This command presets the Questionable Status registers.

## Standard Event Register

The Standard Event register reports the following types of instrument events: power-on detected, command syntax errors, command execution errors, self-test or calibration errors, query errors, or when an \*OPC command is executed. All of these conditions can be reported in the Standard Event summary bit through 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 commands prior to and including *OPC have been executed.
1 Not Used	2	Always zero.
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-Dependent Error	8	A self-test or calibration error occurred. For a complete listing of the error messages, refer to <a href="#">Chapter 14, "Error Messages"</a> on page 130.
4 Execution Error	16	An execution error occurred (an error in the –200 range has been generated).
5 Command Error	32	A command syntax error occurred (an error in the –100 range has been generated).

## 2 SCPI Status Registers

### SCPI Status Registers

Bit number	Decimal value	Definition
6 Not Used	64	Always zero.
7 Power On	128	Power has been turned off and on since the last time the event register was read or cleared.

The Standard Event register is cleared when:

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

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

## Status Byte Register

The Status Byte summary register reports conditions from the other status registers. Query data that is waiting in the U2722A/U2723A's output buffer is immediately reported via message available bit (bit 4). Bits in the summary register are not latched. Clearing an event register will clear the corresponding bits in the Status Byte summary register. The message available bit will be cleared once all the messages in the output buffer, including any pending queries, have been read.

### Bit Definitions: Status Byte Register

Bit number	Decimal value	Definition
0 Not Used	1	Always zero.
1 Not Used	2	Always zero.
2 Error Queue	4	There is at least one error code in the error queue.
3 Questionable Status Event	8	One or more bits are set in the Questionable Event register (bits must be enabled in the enable register).
4 Message Available	16	Data is available in the instrument's output buffer.
5 Event Status Byte summary	32	One or more bits are set in the Standard Event register (bits must be enabled in the enable register).

Bit number	Decimal value	Definition
6 Master Status summary (Request for Service)	64	One or more bits are set in the Status Byte register (bits must be enabled in the enable register). Also used to indicate a request for service.
7 Operation Status Event	128	One or more bits are set in the Operation Event register (bits must be enabled in the enable register).

The Status Byte summary register will be cleared when:

- you execute the clear status (\*CLS) command
- querying the Standard Event register (\*ESR? command will clear only bit 4 in the summary register)

The Status Byte Enable register is cleared when you execute the \*SRE 0 command.

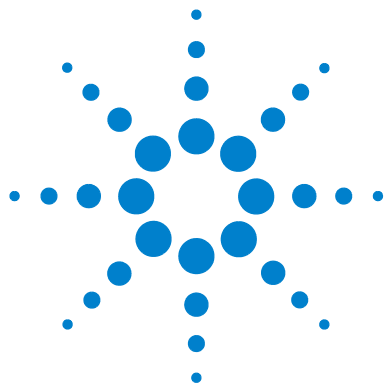
#### NOTE

Please refer to [Chapter 13, “IEEE-488.2 Common Commands”](#) on page 111 for more details of the common IEEE commands mentioned above.

## **2 SCPI Status Registers**

SCPI Status Registers





## 3 ABORt Subsystem

ABORt:TRANsient 18

This chapter describes the ABORt command subsystem which is used to cancel any triggered actions.



# ABORt:TRANsient

**Syntax**

ABORt:TRANsient (@<ch>)

This command cancels any transient trigger actions and returns the transient trigger state back to idle. It also resets the WTG transient bits in the Operation Condition Status register.

**Parameter**

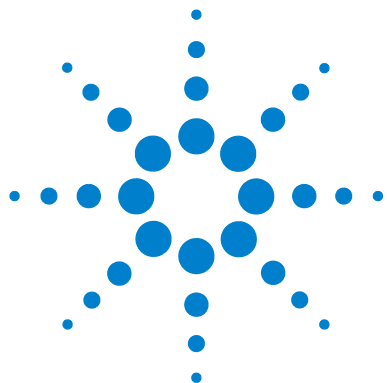
Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

**Remark**

These commands are executed at power-on and upon execution of \*RST.

**Example**

ABOR:TRAN (@1)



## 4 CONFigure Subsystem

CONFigure:SSI 20

The CONFigure:SSI command from the CONFigure subsystem handles the synchronization status when the U2722A/U2723A is used in the U2781A modular instrument chassis.



# CONFigure:SSI

### Syntax

CONFigure:SSI <status>, (@<address\_list>)

This command sets the synchronization status when used in the U2781A modular instrument chassis.

CONFigure:SSI?

This query command returns a string value, which represents the synchronization status.

**NOTE**

The U2722A/U2723A is not designed to function as a master in synchronization.

### Parameters

Item	Type	Range of values	Default value
<address_list>	NR1	0 through 7	Required parameter
<status>	CPD	<ul style="list-style-type: none"><li>• NONE: None</li><li>• SLAVE: To receive triggering source from a master</li></ul>	NONE

**NOTE**

- This command is only operational when it is used in a U2781A modular instrument chassis.
- Each U2722A/U2723A occupies two slots in the U2781A.

### Remark

This command will force all reference clock sources to SSI (Synchronous Simultaneous Interface).

### Returned Query Format

<CRD>, <NR1>

### **Examples**

The following command sets the synchronization status of the U2722A/U2723A to slave.

```
CONFigure:SSI SLAV, (@1)
```

The following query returns the synchronization status of U2722A/U2723A.

```
CONFigure:SSI?
```

Typical Response : SLAV, 1

## **4 CONFigure Subsystem**

CONFigure:SSI



## 5 INITiate Subsystem

INITiate[:IMMediate]:TRANsient 24

This chapter discusses the INITiate command subsystem for the U2722A/U2723A.



# INITiate[:IMMediate]:TRANsient

**Syntax**

```
INITiate[:IMMediate]:TRANsient (@<ch>)
```

This command enables the transient trigger system. When a transient or output trigger is initiated, an event on a selected trigger source causes the specified triggering action to occur. If the trigger system is not initiated, all triggers are ignored.

**Parameter**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

**Example**

```
INIT:TRAN (@1)
```

**See Also**

[ABORt:TRANsient](#)





## 6 MEASure Subsystem

MEASure[:SCALar]:CURRent[:DC]? 26

MEASure[:SCALar]:VOLTage[:DC]? 27

MEASure:ARRay:CURRent[:DC]? 28

MEASure:ARRay:VOLTage[:DC]? 29

MEASure:TEMPerature? 30

The MEASure command subsystem is a group of instructions used to acquire data from the U2722A/U2723A.



# MEASure[:SCALar]:CURRent[:DC]?

**Syntax**

MEASure[:SCALar]:CURRent[:DC]? (@<ch>)

Queries the current measured across the current sense resistor inside the U2722A/U2723A.

**Parameter**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

**Returned Query Format**

<NR3>

The reading is in the form of +9.99999999E+10 when output is set to OFF.

**Example**

MEAS:CURR? (@1)

Typical Response: +9.99999999E+10

**See Also**

[MEASure\[:SCALar\]:VOLTage\[:DC\]?](#)

[MEASure:ARRay:CURRent\[:DC\]?](#)

# MEASure[:SCALar]:VOLTage[:DC]?

## Syntax

MEASure[:SCALar]:VOLTage[:DC]? (@<ch>)

Queries the voltage measured at the sense terminals of the U2722A/U2723A for the specified channel.

## Parameter

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

## Returned Query Format

<NR3>

The reading is in the form of +9.99999999E+10 when output is set to OFF.

## Example

MEAS:VOLT? (@1)

Typical Response: +9.99999999E+10

## See also

[MEASure\[:SCALar\]:CURREnt\[:DC\]?](#)

[MEASure:ARRay:VOLTage\[:DC\]?](#)

# MEASure:ARRay:CURRent[:DC]?

## Syntax

MEASure:ARRay:CURRent[:DC]? (@<ch>)

This query initiates and triggers a measurement and returns an array containing the digitized output current in amperes. The sampling rate is set by [SENSe:SWEEp:TINTerval](#) whereas the number of points is set by [SENSe:SWEEp:POINTs](#).

## Parameter

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

## Returned Query Format

<NR3>{ , <NR3> }

The reading is in the form of +9.99999999E+10 when output is set to OFF. Array values responses are separated by commas.

## Example

MEAS:ARR:CURR? (@1)

## See Also

[MEASure:ARRay:VOLTage\[:DC\]?](#)

[SENSe:SWEEp:POINTs](#)

[SENSe:SWEEp:TINTerval](#)

# MEASure:ARRay:VOLTage[:DC]?

## Syntax

MEASure:ARRay:VOLTage[:DC]? (@<ch>)

This query initiates and triggers a measurement and returns an array containing the digitized output voltage in volts. The sampling rate is set by [SENSe:SWEep:TINTerval](#) whereas the returned number of points is set by [SENSe:SWEep:POINTs](#).

## Parameter

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

## Returned Query Format

<NR3>{ , <NR3> }

The reading is in the form of +9.99999999E+10 when output is set to OFF.  
Array values responses are separated by commas.

## Example

MEAS:ARR:VOLT? (@1)

## See Also

[MEASure:ARRay:CURREnt\[:DC\]?](#)

[SENSe:SWEep:POINTs](#)

[SENSe:SWEep:TINTerval](#)

## MEASure:TEMPerature?

### Syntax

MEASure:TEMPerature?

Queries the temperature measured at the measurement board in degree Celsius (°C).

### Returned Query Format

<NR2>

### Example

MEAS:TEMP?



## 7 OUTPut Subsystem

OUTPut[:STATe] 32

The OUTPut command subsystem is used to switch on or off the U2722A/U2723A output.



# OUTPut[:STATe]

## Syntax

OUTPut[:STATe] <mode>, (@<ch>)

This enables or disables the output of the given channel(s).

OUTPut[:STATe]? (@<ch>)

This query command returns a string value representing the status of the output.

## Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<mode>	Bool	<ul style="list-style-type: none"><li>• OFF or 0: Disables the output</li><li>• ON or 1: Enables the output</li></ul>	Required parameter

## Returned Query Format

<NR1>

The query command returns 0 if the output is OFF, and 1 if the output is ON.

## Examples

The following command turns the output of channel 2 on.

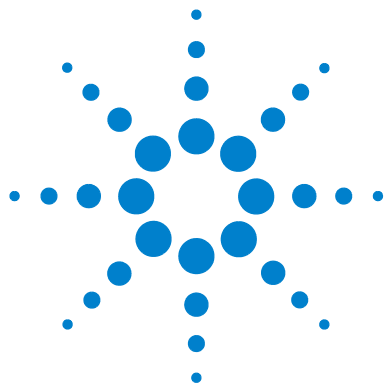
OUTP 1, (@2)

The following query returns the output state of channel 1.

OUTP? (@1)

Typical Response: +1





## 8 SENSe Subsystem

SENSe:CURRent[:DC]:APERture?	34
SENSe:CURRent[:DC]:NPLCycles	35
SENSe:SWEep:POINts	37
SENSe:SWEep:TINterval	39
SENSe:VOLTage[:DC]:APERture?	40
SENSe:VOLTage[:DC]:NPLCycles	41

The SENSe command subsystem directly affects the device specific settings used to make measurements.



# SENSe:CURRent[:DC]:APERture?

**Syntax**

```
SENSe:CURRent[:DC]:APERture? (@<ch>)
```

This will query the sampling time for a single current measurement point. The parameter has a unit of seconds.

**Parameter**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

**Remark**

The aperture value is derived from the following equation.

$$Aperture = \frac{NPLCycles}{Line\ frequency}$$

**Returned Query Format**

```
<NR3>
```

The reading is in the form of +1.000000E+01.

**Example**

```
SENS:CURR:APER? (@2)
```

Typical Response: +2.000000E-02

**See Also**

[SENSe:CURRent\[:DC\]:NPLCycles](#)

[SENSe:VOLTage\[:DC\]:APERture?](#)

[SENSe:VOLTage\[:DC\]:NPLCycles](#)

[SYSTem:LFRequency](#)

# SENSe:CURRent[:DC]:NPLCycles

## Syntax

SENSe:CURRent[:DC]:NPLCycles <cycles>, (@<ch>)

This command sets the number of power line cycles taken for the current measurement. The value is coupled to APERTure by the equation:

*APERTure = NPLCycles / selected line frequency.*

SENSe:CURRent[:DC]:NPLCycles? (@<ch>)

Query the number of power line cycles taken for the current measurement in the defined channels.

## Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<cycles>	NR1	0 to 255	Required parameter

## Returned Query Format

<NR1>

## Examples

SENS:CURR:NPLC 10, (@1)

SENS:CURR:NPLC? (@1)

## 8 SENSE Subsystem

SENSe:CURRent[:DC]:NPLCycles

### See Also

SENSe:CURRent[:DC]:APERture?

SENSe:VOLTage[:DC]:NPLCycles

# SENSe:SWEep:POINts

## Syntax

SENSe:SWEep:POINts <points>, (@<ch>)

This command defines the number of points in a measurement on models that have measurement controls. Programmed values can range from 1 to 4096 (4K).

SENSe:SWEep:POINts? (@<ch>)

This will query the number of points set.

## Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<points>	NR1	1 to 4096 (4K)	Required parameter

## Returned Query Format

<NR1>

The query command returns the number of points for each channel specified.

## Examples

The following command specifies 1000 measurement points on channel 1.

SENS:SWE:POIN 1000, (@1)

The following query returns the number of measurement points set on channel 2.

SENS:SWE:POIN? (@2)

**See Also**

[SENSe:SWEep:TINterval](#)

## SENSe:SWEep:TINterval

### Syntax

SENSe:SWEep:TINterval <interval>, (@<ch>)

This command defines the time period between samples in milliseconds on models that have measurement controls. Programmed values can range from 1 to 32767.

SENSe:SWEep:TINterval? (@<ch>)

Queries the time interval set.

### Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<interval>	NR1	1 to 32767	Required parameter

### Returned Query Format

<NR1>

The query command returns the time interval for each channel specified. The returned value is the time period in milliseconds.

### Examples

The following command specifies a time interval of 10 milliseconds between measurement points on channel 1.

```
SENS:SWE:TINT 10, (@1)
```

The following query returns the measurement time interval for channel 2.

```
SENS:SWE:TINT? (@2)
```

### See Also

[SENSe:SWEep:POINTs](#)

# SENSe:VOLTage[:DC]:APERture?

**Syntax**

```
SENSe:VOLTage[:DC]:APERture? (@<ch>)
```

This will query the sampling time for a single voltage measurement point. The parameter has a unit of seconds.

**Parameter**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

**Remark**

The aperture value is derived from the following equation.

$$Aperture = \frac{NPLCycles}{Line\ frequency}$$

**Returned Query Format**

```
<NR3>
```

The reading is in the form +1.000000E+01.

**Example**

```
SENS:VOLT:APER? (@2)  
Typical Response: +2.000000E-02
```

**See Also**

- [SENSe:CURRent\[:DC\]:APERture?](#)
- [SENSe:CURRent\[:DC\]:NPLCycles](#)
- [SENSe:VOLTage\[:DC\]:NPLCycles](#)
- [SYSTem:LFRequency](#)



# SENSe:VOLTage[:DC]:NPLCycles

## Syntax

SENSe:VOLTage[:DC]:NPLCycles <cycles>, (@<ch>)

This command sets the number of power line cycles taken for the voltage measurement. The value is coupled to APERTure by the equation:

*APERTure = NPLCycles/selected line frequency.*

SENSe:VOLTage[:DC]:NPLCycles? (@<ch>)

Query the number of power line cycles taken for the voltage measurement in the defined channels.

## Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<cycles>	NR1	0 to 255	Required parameter

## Returned Query Format

<NR1>

## Examples

SENS:VOLT:NPLC 10, (@1)

SENS:VOLT:NPLC? (@1)

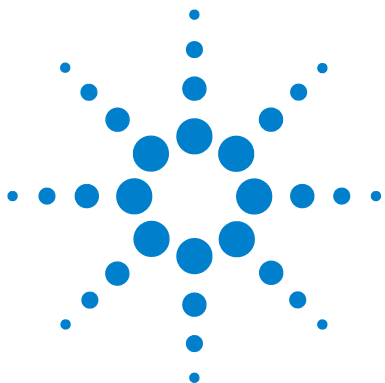
## See Also

[SENSe:CURREnt\[:DC\]:APERTure?](#)

[SENSe:CURREnt\[:DC\]:NPLCycles](#)

## 8 SENSE Subsystem

SENSe:VOLTage[:DC]:NPLCycles



## 9 SOURce Subsystem

[SOURce:]CURRent	44
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]	46
[SOURce:]CURRent:LIMit	48
[SOURce:]CURRent:RANGe	49
[SOURce:]MEMory:ARM	51
[SOURce:]MEMory:CONFigure:POINts	52
[SOURce:]MEMory:CURRent:LIMit	54
[SOURce:]MEMory:CURRent:MEASure	55
[SOURce:]MEMory:CURRent:SOURce	56
[SOURce:]MEMory:CURRent:RANGe	57
[SOURce:]MEMory:LIST	58
[SOURce:]MEMory:LIST:CLEar	60
[SOURce:]MEMory:LIST:POINts?	61
[SOURce:]MEMory:LIST:LOAD	62
[SOURce:]MEMory:LIST:READ?	63
[SOURce:]MEMory:LIST:STORe	64
[SOURce:]MEMory:OUTPut	65
[SOURce:]MEMory:LIST:DATA?	66
[SOURce:]MEMory:SOURce:DELay:AUTO	68
[SOURce:]MEMory:SOURce:DELay	69
[SOURce:]MEMory:TRIGger	70
[SOURce:]MEMory:VOLTage:LIMit	71
[SOURce:]MEMory:VOLTage:MEASure	72
[SOURce:]MEMory:VOLTage:SOURce	73
[SOURce:]MEMory:VOLTage:RANGe	74
[SOURce:]VOLTage	75
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]	77
[SOURce:]VOLTage:LIMit	79
[SOURce:]VOLTage:RANGe	80



# [SOURce:]CURRent

## Syntax

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <current>,  
(@<ch>)

Sets the actual current magnitude of the unswept output signal in terms of the present operating units. The units are set to the default value, or alternately to a different value under the UNIT subsystem. The AMPLitude may be used to specify the level for either a time varying or non-time varying signal.

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]? (@<ch>)

Queries the sourced current.

## Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<current>	NRf	The maximum value is dependent on the current range set	Required parameter

## Returned Query Format

<NR3>

## Examples

The following command sets the output current to 0.02 A on channel 1.

CURR 0.02, (@1)

The following query returns the programmed current level on channel 1.

CURR? (@1)

Typical Response: +2.000000E-02

**See Also**

[SOURCE:]CURRENT[:LEVEL]:TRIGGERed[:AMPLitude]

[SOURCE:]CURRENT:RANGE

[SOURCE:]VOLTage

[SOURCE:]VOLTage[:LEVEL]:TRIGGERed[:AMPLitude]

# [SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]

**Syntax**

```
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <current>,  
(@<ch>)
```

This command sets the current trigger level of the specified output channel. Units are in amperes. The triggered level is a stored value that is transferred to the output when an output step is triggered.

```
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]? (@<ch>)
```

Queries the stored value of triggered current.

**Parameters**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<current>	NRf	The maximum value is dependent on the current range set	Required parameter

**Returned Query Format**

```
<NR3>
```

**Examples**

The following command sets the current trigger level to 0.02 A on channel 1.

```
CURR:TRIG 0.02, (@1)
```

The following query returns the current trigger level on channel 1.

```
CURR:TRIG? (@1)
```

Typical Response: +2.000000E-02

[SOURCE:]CURRENT[:LEVEL]:TRIGGERed[:AMPLitude]

### See Also

INITiate[:IMMediate]:TRANsient

[SOURCE:]CURRENT:RANGe

[SOURCE:]VOLTage[:LEVEL]:TRIGGERed[:AMPLitude]

# [SOURce:]CURRent:LIMit

### Syntax

[SOURce:]CURRent:LIMit <value>, (@<ch>)

This command sets the maximum bounds on the output current value. Output current level will be clamped to the limit value if the current level has exceeded the bounds set.

[SOURce:]CURRent:LIMit? (@<ch>)

This will query the current limit setting.

### Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<value>	NRf	The maximum value is dependent on the current range set	Required parameter

### Returned Query Format

<NR3>

### Examples

The following command sets the current limit to 0.8 A on channel 1.

CURR:LIM 0.8, (@1)

The following query returns the programmed current limit on channel 1.

CURR:LIM? (@1)

Typical Response: +8.000000E-01

### See Also

[\[SOURce:\]VOLTage:LIMit](#)



## [SOURce:]CURRent:RANGe

### Syntax

[SOURce:]CURRent:RANGe <range>, (@<ch>)

This command sets the output current range. At \*RST, low current range is selected.

[SOURce:]CURRent:RANGe? (@<ch>)

Queries the selected current range.

### Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<range>	CPD	<ul style="list-style-type: none"> <li>R1uA – 1 <math>\mu</math>A range</li> <li>R10uA – 10 <math>\mu</math>A range</li> <li>R100uA – 100 <math>\mu</math>A range</li> <li>R1mA – 1 mA range</li> <li>R10mA – 10 mA range</li> <li>R120mA – 120 mA range</li> </ul>	R1uA

### Returned Query Format

<CRD>

### Examples

The following command sets the output current range to 1 mA on channel 1.

CURR:RANG R1mA, (@1)

The following query returns the programmed current range on channel 1.

CURR:RANG? (@1)

Typical Response: R1mA

## 9 **SOURce Subsystem**

[SOURce:]CURRent:RANGe

### **See Also**

[SOURce:]VOLTage:RANGe

[SOURCE:]MEMory:ARM

Syntax

```
[SOURCE:]MEMory:ARM (@<ch>)
```

This command arms the specified channel where the commands in the channel active memory list will be executed once the external trigger signal is received by the transient system.

Parameter

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

Remarks

This command is only applicable for the U2723A USB Modular SMU.

Example

```
MEM:ARM (@1)
```

# [SOURce:]MEMory:CONFigure:POINts

## Syntax

[SOURce:]MEMory:CONFigure:POINts <start>,<end>,<loop>,  
(@<ch>)

This command sets the start point, end point, and loop count for the channel active memory list.

[SOURce:]MEMory:CONFigure:POINts?

Queries the start point, end point, and loop count value for the channel active memory list.

## Parameter

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<start>	NR1	1 to 200	Required parameter
<end>	NR1	1 to 200	Required parameter
<loop>	NR1	1 to 1000	Required parameter

## Remarks

This command is only applicable for the U2723A USB Modular SMU. If the set value is larger than the number of commands in the memory list, or the end point is larger than the start point,the input is invalid and error -222 is displayed.

## Returned Query Format

<NR1>,<NR1>,<NR1>

**Example**

```
MEM:CONF:POIN 1,100,1 (@1)
```

```
MEM:CONF:POIN? (@1)
```

**See Also**

[CONFigure:SSI](#)

# [SOURce:]MEMory:CURRent:LIMit

**Syntax**

[SOURce:]MEMory:CURRent:LIMit <value>, (@<ch>)

This command stores the CURRent:LIMit <value> command into the channel active memory list.

**Parameter**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<value>	NRf	The maximum value is dependent on the current range set in the memory list	Required parameter

**Remarks**

This command is only applicable for the U2723A USB Modular SMU.

**Example**

MEM:CURR:LIM 0.12, (@1)

**See Also**

[\[SOURce:\]MEMory:CURRent:MEASure](#)

[\[SOURce:\]MEMory:CURRent:SOURce](#)

# [SOURce:]MEMory:CURRent:MEASure

## Syntax

[SOURce:]MEMory:CURRent:MEASure (@<ch>)

This command stores the MEASure:CURRent? command into the channel active memory list.

## Parameter

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

## Remarks

This command is only applicable for the U2723A USB Modular SMU.

## Example

MEM:CURR:MEAS (@1)

## See Also

[\[SOURce:\]MEMory:CURRent:LIMit](#)

[\[SOURce:\]MEMory:CURRent:SOURce](#)

# [SOURce:]MEMory:CURRent:SOURce

**Syntax**

[SOURce:]MEMory:CURRent:SOURce <value>, (@<ch>)

This command stores the CURRent <value> command into the channel active memory list.

**Parameter**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<value>	NRf	The maximum value is dependent on the current range set in the memory list	Required parameter

**Remarks**

This command is only applicable for the U2723A USB Modular SMU.

**Example**

MEM:CURR:SOUR 0.12, (@1)

**See Also**

[\[SOURce:\]MEMory:CURRent:LIMit](#)

[\[SOURce:\]MEMory:CURRent:MEASure](#)



# [SOURCE:]MEMORY:CURRENT:RANGE

## Syntax

[SOURCE:]MEMORY:CURRENT:RANGE <range>, (@<ch>)

This command stores the CURRENT:RANGE <range> command into the channel active memory list.

## Parameter

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<range>	CPD	<ul style="list-style-type: none"> <li>R1uA – 1 <math>\mu</math>A range</li> <li>R10uA – 10 <math>\mu</math>A range</li> <li>R100uA – 100 <math>\mu</math>A range</li> <li>R1mA – 1 mA range</li> <li>R10mA – 10 mA range</li> <li>R120mA – 120 mA range</li> </ul>	R1uA

## Remarks

This command is only applicable for the U2723A USB Modular SMU.

## Example

MEM:CURR:RANG R120mA, (@1)

## See Also

[\[SOURCE:\]MEMORY:CURRENT:LIMit](#)

[\[SOURCE:\]MEMORY:CURRENT:MEASure](#)

# [SOURce:]MEMory:LIST

### Syntax

[SOURce:]MEMory:LIST <list>, (@<ch>)

This command sets the channel active memory list.

[SOURce:]MEMory:LIST?

Queries the channel active memory list.

### Parameter

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<list>	NR1	<ul style="list-style-type: none"><li>1: Memory list 1 is set as the active memory list</li><li>2: Memory list 2 is set as the active memory list</li></ul>	1

### Remarks

This command is only applicable for the U2723A USB Modular SMU.

### Returned Query Format

<NR1>

### Example

MEM:LIST 1, (@1)

MEM:LIST? (@1)

**See Also**

[SOURCE:]MEMORY:LIST:CLEAR

[SOURCE:]MEMORY:LIST:POINTS?

[SOURCE:]MEMORY:LIST:LOAD

[SOURCE:]MEMORY:LIST:READ?

[SOURCE:]MEMORY:LIST:STORE

# [SOURce:]MEMory:LIST:CLEar

## Syntax

[SOURce:]MEMory:LIST:CLEar (@<ch>)

This command clears the active memory list's command.

## Parameter

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

## Remarks

This command is only applicable for the U2723A USB Modular SMU.

## Example

MEM:LIST:CLE (@1)

## See Also

- [\[SOURce:\]MEMory:CURRent:RANGe](#)
- [\[SOURce:\]MEMory:LIST:POINts?](#)
- [\[SOURce:\]MEMory:LIST:LOAD](#)
- [\[SOURce:\]MEMory:LIST:READ?](#)
- [\[SOURce:\]MEMory:LIST:STORe](#)

[SOURCE:]MEMORY:LIST:POINTS?

Syntax

```
[SOURCE:]MEMORY:LIST:POINTS? (@<ch>)
```

This command queries the total number of command in the channel active memory list.

Parameter

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

Remarks

This command is only applicable for the U2723A USB Modular SMU.

Returned Query Format

```
<NR1>
```

Example

```
MEM:LIST:POIN? (@1)
```

See Also

- [SOURCE:]MEMORY:CURRENT:RANGE
- [SOURCE:]MEMORY:LIST:CLEAR
- [SOURCE:]MEMORY:LIST:LOAD
- [SOURCE:]MEMORY:LIST:READ?
- [SOURCE:]MEMORY:LIST:STORE

# [SOURCE:]MEMORY:LIST:LOAD

**Syntax**

[SOURCE:]MEMORY:LIST:LOAD (@<ch>)

This command loads the channel active memory list from the nonvolatile memory.

**Parameter**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

**Remarks**

This command is only applicable for the U2723A USB Modular SMU.

**Example**

MEM:LIST:LOAD (@1)

**See Also**

- [\[SOURCE:\]MEMORY:CURRENT:RANGE](#)
- [\[SOURCE:\]MEMORY:LIST:CLEAR](#)
- [\[SOURCE:\]MEMORY:LIST:POINTS?](#)
- [\[SOURCE:\]MEMORY:LIST:READ?](#)
- [\[SOURCE:\]MEMORY:LIST:STORE](#)

[SOURCE:]MEMORY:LIST:READ?

Syntax

```
[SOURCE:]MEMORY:LIST:READ? (@<ch>)
```

This command reads the commands from the channel active memory list.

Parameter

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

Remarks

This command is only applicable for the U2723A USB Modular SMU.

Returned Query Format

```
<AARD>
```

Memory list’s command responses are separated by semicolons.

If there are no commands in the memory list, null is returned.

Example

```
MEM:LIST:READ? (@1)
```

See Also

- [SOURCE:]MEMORY:CURRENT:RANGE
- [SOURCE:]MEMORY:LIST:CLEAR
- [SOURCE:]MEMORY:LIST:POINTS?
- [SOURCE:]MEMORY:LIST:LOAD
- [SOURCE:]MEMORY:LIST:STORE

# [SOURce:]MEMory:LIST:STORe

**Syntax**

[SOURce:]MEMory:LIST:STORe (@<ch>)

This command stores the channel active memory list into the nonvolatile memory.

**Parameter**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

**Remarks**

This command is only applicable for the U2723A USB Modular SMU.

**Example**

MEM:LIST:STOR (@1)

**See Also**

- [\[SOURce:\]MEMory:CURRent:RANGe](#)
- [\[SOURce:\]MEMory:LIST:CLEar](#)
- [\[SOURce:\]MEMory:LIST:POINts?](#)
- [\[SOURce:\]MEMory:LIST:LOAD](#)
- [\[SOURce:\]MEMory:LIST:READ?](#)



[SOURce:]MEMory:OUTPut

Syntax

```
[SOURce:]MEMory:OUTPut <mode>, (@<ch>)
```

This command stores the OUTP <mode> command into the channel active memory list.

Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<mode>	Bool	OFF or 0: Disables the output ON or 1: Enables the output	Required parameter

Remarks

This command is only applicable for the U2723A USB Modular SMU.

Example

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

# [SOURce:]MEMory:LIST:DATA?

**Syntax**

```
[SOURce:]MEMory:LIST:DATA? (@<ch>)
```

This command reads the data result after executing the commands in the channel active memory list through a remote or hardware trigger action.

**Parameter**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

**Remarks**

This command is only applicable for the U2723A USB Modular SMU.

**Returned Query Format**

```
<NR3>{,<NR3>}
```

The reading is in the form of +9.99999999E+10 when output is set to OFF or no measurement is made during the memory list's commands execution.

Array values responses are separated by commas.

**Example**

```
MEM:LIST:DATA? (@1)
```

**See Also**

[SOURCE:]CURRENT:RANGE

[SOURCE:]MEMORY:LIST:CLEAR

[SOURCE:]MEMORY:LIST:POINTS?

[SOURCE:]MEMORY:LIST:LOAD

[SOURCE:]MEMORY:LIST:READ?

[SOURCE:]MEMORY:LIST:STORE

# [SOURce:]MEMory:SOURce:DElay:AUTO

**Syntax**

[SOURce:]MEMory:SOURce:DElay:AUTO <mode>, (@<ch>)

Store this command to turn on or off auto source delay in channel active memory list's commands execution.

**Parameters**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<mode>	Bool	<ul style="list-style-type: none"><li>• OFF or 0: Disables the auto source delay</li><li>• ON or 1: Enables the auto source delay</li></ul>	Required parameter

**Remarks**

This command is only applicable for the U2723A USB Modular SMU.

**Example**

MEM:SOUR:DEL:AUTO ON, (@1)

[SOURce:]MEMory:SOURce:DElay

Syntax

[SOURce:]MEMory:SOURce:DElay <mode>,<value>, (@<ch>)

Store this command to set source delay time in ms for channel active memory list.

Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<mode>	CPD	<ul style="list-style-type: none"> <li>GLOBAL: Set once and the following source delay will follow this timing</li> <li>SINGLE: Affect the next subsequent source action in the memory list.</li> </ul>	GLOB
<value>	NRf	0 to 65535	0

Remarks

This command is only applicable for the U2723A USB Modular SMU.

Example

MEM:SOUR:DEL:SING,5,(@1)

See Also

[\[SOURce:\]MEMory:SOURce:DElay:AUTO](#)

# [SOURce:]MEMory:TRIGger

**Syntax**

```
[SOURce:]MEMory:TRIGger (@<ch>)
```

This command executes the active memory list commands for the specified channel.

**Parameters**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

**Remarks**

This command is only applicable for the U2723A USB Modular SMU.

**Example**

```
MEM:TRIG (@1)
```

[SOURce:]MEMory:VOLTage:LIMit

Syntax

[SOURce:]MEMory:VOLTage:LIMit <value>, (@<ch>)

This command stores the VOLTage:LIMit <value> command into the channel active memory list.

Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<value>	NRf	The maximum value is dependent on the voltage range set in the memory list	Required parameter

Remarks

This command is only applicable for the U2723A USB Modular SMU.

Example

MEM:VOLT:LIM 20, (@1)

See Also

[\[SOURce:\]MEMory:VOLTage:MEASure](#)

[\[SOURce:\]MEMory:VOLTage:RANGe](#)

# [SOURce:]MEMory:VOLTage:MEASure

**Syntax**

[SOURce:]MEMory:VOLTage:MEASure (@<ch>)

This command stores the MEASure:VOLTage? command into the channel active memory list.

**Parameters**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter

**Remarks**

This command is only applicable for the U2723A USB Modular SMU.

**Example**

MEM:VOLT:MEAS (@1)

**See Also**

[\[SOURce:\]MEMory:VOLTage:LIMit](#)

[\[SOURce:\]MEMory:VOLTage:RANGe](#)



[SOURce:]MEMory:VOLTage:SOURce

Syntax

```
[SOURce:]MEMory:VOLTage:SOURce <value>, (@<ch>)
```

This command stores the VOLTage <value> command into the channel active memory list.

Parameter

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<value>	NRf	The maximum value is dependent on the voltage range set in the memory list	Required parameter

Remarks

This command is only applicable for the U2723A USB Modular SMU.

Example

```
MEM:VOLT:SOUR 20, (@1)
```

See Also

- [SOURce:]MEMory:VOLTage:LIMit
- [SOURce:]MEMory:VOLTage:MEASure

# [SOURce:]MEMory:VOLTage:RANGe

## Syntax

[SOURce:]MEMory:VOLTage:RANGe <range>, (@<ch>)

This command stores the VOLTage:RANGe <range> command into the channel active memory list.

## Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<range>	CPD	<ul style="list-style-type: none"><li>R2V – 2 V range</li><li>R20V – 20 V range</li></ul>	Required parameter

## Remarks

This command is only applicable for the U2723A USB Modular SMU.

## Example

MEM:VOLT:RANG R20V, (@1)

## See Also

[\[SOURce:\]MEMory:VOLTage:LIMit](#)

[\[SOURce:\]MEMory:VOLTage:MEASure](#)

# **[SOURce:]VOLTage**

## **Syntax**

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <voltage>,  
(@<ch>)

Sets the actual voltage magnitude of the unswept output signal in terms of the present operating units. The units are set to the default value, or alternately to a different value under the UNIT subsystem. The AMPLitude may be used to specify the level for either a time varying or non-time varying signal.

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? (@<ch>)

Queries the sourced voltage.

## **Parameters**

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<voltage>	NRf	The maximum value is dependent on the voltage range set	Required parameter

## **Returned Query Format**

<NR3>

## **Examples**

The following command sets the output voltage to 5 V on channel 1.

VOLT 5, (@1)

The following query returns the programmed voltage level on channel 1.

VOLT? (@1)

Typical Response: +5.000000E+00

## 9 **SOURce Subsystem**

[SOURce:]VOLTage

### **See Also**

[SOURce:]CURRent

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]

## [SOURCE:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]

### Syntax

```
[SOURCE:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <voltage>,  
(@<ch>)
```

This command sets the voltage trigger level of the specified output channel. Units are in voltage. The triggered level is a stored value that is transferred to the output when an output step is triggered.

```
[SOURCE:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]? (@<ch>)
```

Query the stored value of triggered voltage.

### Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<voltage>	NRf	The maximum value is dependent on the voltage range set	Required parameter

### Returned Query Format

```
<NR3>
```

### Examples

The following command sets the voltage trigger level to 0.02 V on channel 1.

```
VOLT:TRIG 0.02, (@1)
```

The following query returns the voltage trigger level on channel 1.

```
VOLT:TRIG? (@1)
```

Typical Response: +2.000000E-02

## 9 **SOURce Subsystem**

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]

### **See Also**

INITiate[:IMMediate]:TRANsient

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]

[SOURce:]CURRent:RANGe

## [SOURce:]VOLTage:LIMit

### Syntax

[SOURce:]VOLTage:LIMit <value>, (@<ch>)

This command sets the maximum bounds on the output voltage value. Output voltage level will be clamped to the limit value if the voltage level has exceeded the bounds set.

[SOURce:]VOLTage:LIMit? (@<ch>)

This will query the set voltage limit.

### Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<value>	NRf	The maximum value is dependent on the voltage range set	Required parameter

### Returned Query Format

<NR3>

### Examples

The following command sets the voltage limit to 0.5 V on channel 1.

VOLT:LIM 0.5, (@1)

The following query returns the programmed voltage limit on channel 1.

VOLT:LIM? (@1)

Typical Response: +5.000000E-01

### See Also

[\[SOURce:\]CURRent:LIMit](#)

# [SOURce:]VOLTage:RANGe

### Syntax

[SOURce:]VOLTage:RANGe <range>, (@<ch>)

This command sets the output voltage range. At \*RST, low voltage range is selected.

[SOURce:]VOLTage:RANGe? (@<ch>)

Queries the selected voltage range.

### Parameters

Item	Type	Range of values	Default value
<ch>	NR1	1 through 3	Required parameter
<range>	CPD	<ul style="list-style-type: none"><li>• R2V – 2 V range</li><li>• R20V – 20 V range</li></ul>	Required parameter

### Returned Query Format

<CRD>

### Examples

The following command sets the output voltage range to 20 V on channel 1.

VOLT:RANG R20V, (@1)

The following query returns the programmed voltage range on channel 1.

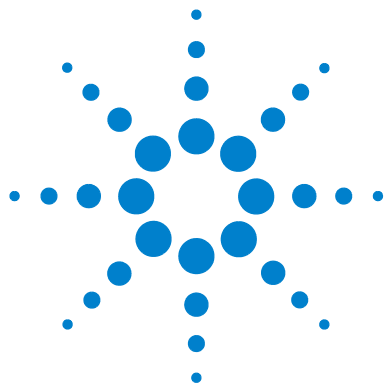
VOLT:RANG? (@1)

Typical Response: R20V

### See Also

[\[SOURce:\]CURRent:RANGe](#)





## 10 STATus Subsystem

STATus:OPERation:CONDition?	82
STATus:OPERation:ENABle	84
STATus:OPERation[:EVENT]?	87
STATus:OPERation:NTRansition	89
STATus:OPERation:PTRansition	91
STATus:PRESet	93
STATus:QUEStionable:CONDition?	94
STATus:QUEStionable:ENABle	95
STATus:QUEStionable[:EVENT]?	97
STATus:QUEStionable:NTRansition	99
STATus:QUEStionable:PTRansition	101

The STATus command subsystem lets you determine the operating condition of the U2722A/U2723A at any time.



# STATus:OPERation:CONDition?

## Syntax

STATus:OPERation:CONDition?

This query command returns the value of the Operation Status Condition register. The Condition register is a read-only register, which holds the live (unlatched) operational status of the instrument. Reading the Operation Condition register does not clear it.

## NOTE

For more information on the SCPI Status System, refer to [Chapter 2, "Status system diagram"](#) on page 11.

## Remarks

The following table lists the bit definitions for the Operation Status registers.

Bit number	Decimal value	Definition
0 to 1 Not Used	0	0 is returned.
2 DTG1-tran	4	The transient system has received a trigger signal and operation is running in channel 1.
3 DTG2-tran	8	The transient system has received a trigger signal and operation is running in channel 2.
4 DTG3-tran	16	The transient system has received a trigger signal and operation is running in channel 3.
5 WTG1-tran	32	The transient system is waiting for a trigger in channel 1.
6 WTG2-tran	64	The transient system is waiting for a trigger in channel 2.
7 WTG3-tran	128	The transient system is waiting for a trigger in channel 3.
8 to 15 Not Used	0	0 is returned.

## Returned Query Format

<NR1>

The returned value is the binary-weighted sum of all bits set in the register. For example, if bit 5 (decimal value = 32) and bit 6 (decimal value = 64) are set, this query command will return "+96" (32 + 64).

**Example**

The following query command reads the Operation Status Condition register.

STAT:OPER:COND?

Typical Response: +32

# STATus:OPERation:ENABle

## Syntax

STATus:OPERation:ENABle <value>

This command and its query set and read the value of the Operation Status Enable register. The enable register is a mask for enabling specific bits from the Operation Event register to set the operation summary bit (OPER) of the Status Byte register. This bit (bit 7) is the logical OR of all the Operation Event register bits that are enabled by the Operation Status Enable register.

STATus:OPERation:ENABle?

This query the value of the Operation Status Enable register

**NOTE**

For more information on the SCPI Status System, refer to [Chapter 2, “Status system diagram”](#) on page 11.

## Parameters

Item	Type	Range of values	Default value
<value>	NR1	A decimal value which corresponds to the binary-weighted sum of the bits in the register (refer to the table below).	Preset = 0

For example, to enable bit 5 (decimal value = 32) and bit 6 (decimal value = 64), the corresponding decimal value would be 96 (32 + 64).

## Remarks

- The following table lists the bit definitions for the Operation Status registers.

Bit number	Decimal value	Definition
0 to 1 Not Used	0	0 is returned.
2 DTG1-tran	4	The transient system has received a trigger signal and operation is running in channel 1.
3 DTG2-tran	8	The transient system has received a trigger signal and operation is running in channel 2.
4 DTG3-tran	16	The transient system has received a trigger signal and operation is running in channel 3.
5 WTG1-tran	32	The transient system is waiting for a trigger in channel 1.
6 WTG2-tran	64	The transient system is waiting for a trigger in channel 2.
7 WTG3-tran	128	The transient system is waiting for a trigger in channel 3.
8 to 15 Not Used	0	0 is returned.

- The **STATus:PRESet** command will clear all bits in the enable register.

## Returned Query Format

<NR1>

The returned value is the binary-weighted sum of all bits set in the register. For example, if bit 5 (decimal value = 32) and bit 6 (decimal value = 64) are set, the query command will return "+96".

## Examples

The following command enables bits 5 (decimal value = 32) and 6 (decimal value = 64) in the enable register.

```
STAT:OPER:ENAB 96
```

The following query returns the bits enabled in the register.

```
STAT:OPER:ENAB?
```

Typical Response: +96

## 10 STATUS Subsystem

STATUS:OPERation:ENABle

### See Also

STATUS:OPERation[:EVENT]?

STATUS:PRESet

## STATus:OPERation[:EVENT]?

### Syntax

STATus:OPERation[:EVENT]?

This query returns the value of the Operation Status Event register. The Event register is a read-only register, which stores (latches) all events that are passed by the Operation NTR and/or PTR filter. Reading the Operation Status Event register clears it.

### NOTE

For more information on the SCPI Status System, refer to [Chapter 2, “Status system diagram”](#) on page 11 .

### Remarks

- The following table lists the bit definitions for the Operation Status registers.

Bit number	Decimal value	Definition
0 to 1 Not Used	0	0 is returned.
2 DTG1-tran	4	The transient system has received a trigger signal and operation is running in channel 1.
3 DTG2-tran	8	The transient system has received a trigger signal and operation is running in channel 2.
4 DTG3-tran	16	The transient system has received a trigger signal and operation is running in channel 3.
5 WTG1-tran	32	The transient system is waiting for a trigger in channel 1.
6 WTG2-tran	64	The transient system is waiting for a trigger in channel 2.
7 WTG3-tran	128	The transient system is waiting for a trigger in channel 3.
8 to 15 Not Used	0	0 is returned.

- Once a bit is set, it remains set until cleared by reading the event register or the clear status (\*CLS) command.

### Returned Query Format

<NR1>

## 10 STATus Subsystem

STATus:OPERation[:EVENT]?

The returned value is the binary-weighted sum of all bits set in the register. For example, if bit 5 (decimal value = 32) and bit 6 (decimal value = 64) are set, this query command will return "+96".

### Example

The following query command reads the Operation Status Event register.

STAT:OPER?

Typical Response: +96

### See Also

[STATus:OPERation:ENABle](#)

[\\*CLS](#)



# STATUS:OPERation:NTRansition

## Syntax

STATUS:OPERation:NTRansition <value>

This register serves as a polarity filter between the Operation Condition and Operation Event registers. When a bit in the Operation NTR register is set to 1, then a 1-to-0 transition of the corresponding bit in the Operation Condition register causes that bit in the Operation Event register to be set.

STATUS:OPERation:NTRansition?

These command reads the value of the Operation NTR (Negative-Transition) register.

NOTE

For more information on the SCPI Status System, refer to [Chapter 2, “Status system diagram”](#) on page 11.

## Parameters

Item	Type	Range of values	Default value
<value>	NR1	A decimal value which corresponds to the binary-weighted sum of the bits in the register (refer to the table below).	Preset = 0

For example, to enable bit 5 (decimal value = 32) and bit 6 (decimal value = 64), the corresponding decimal value would be 96 (32 + 64).

## Remarks

- The following table lists the bit definitions for the Operation Status registers.

## 10 STATus Subsystem

STATus:OPERation:NTRansition

Bit number	Decimal value	Definition
0 to 1 Not Used	0	0 is returned.
2 DTG1-tran	4	The transient system has received a trigger signal and operation is running in channel 1.
3 DTG2-tran	8	The transient system has received a trigger signal and operation is running in channel 2.
4 DTG3-tran	16	The transient system has received a trigger signal and operation is running in channel 3.
5 WTG1-tran	32	The transient system is waiting for a trigger in channel 1.
6 WTG2-tran	64	The transient system is waiting for a trigger in channel 2.
7 WTG3-tran	128	The transient system is waiting for a trigger in channel 3.
8 to 15 Not Used	0	0 is returned.

- The [STATus:PRESet](#) command will set all bits in the NTR register to 0.

### Returned Query Format

<NR1>

The returned value is the binary-weighted sum of all bits set in the register.

### Examples

The following command enables bits 5 (decimal value = 32) and 6 (decimal value = 64) in the NTR register.

```
STAT:OPER:NTR 96
```

The following query returns the bits enabled in the register.

```
STAT:OPER:NTR?
```

Typical Response: +96

### See Also

[STATus:OPERation:PTRansition](#)

[STATus:PRESet](#)

# STATUS:OPERation:PTRansition

## Syntax

STATUS:OPERation:PTRansition <value>

This register serves as a polarity filter between the Questionable Condition and Questionable Event registers. When a bit in the Questionable PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Questionable Condition register causes that bit in the Questionable Event register to be set.

STATUS:OPERation:PTRansition?

These reads the value of the Questionable PTR (Positive-Transition) register.

NOTE

For more information on the SCPI Status System, refer to [Chapter 2, “Status system diagram”](#) on page 11.

## Parameters

Item	Type	Range of values	Default value
<value>	NR1	A decimal value which corresponds to the binary-weighted sum of the bits in the register (refer to the table below).	Preset = 252

For example, to enable bit 5 (decimal value = 32) and bit 6 (decimal value = 64), the corresponding decimal value would be 96 (32 + 64).

## Remarks

- The following table lists the bit definitions for the Operation Status registers.

## 10 STATus Subsystem

STATus:OPERation:PTRansition

Bit number	Decimal value	Definition
0 to 1 Not Used	0	0 is returned.
2 DTG1-tran	4	The transient system has received a trigger signal and operation is running in channel 1.
3 DTG2-tran	8	The transient system has received a trigger signal and operation is running in channel 2.
4 DTG3-tran	16	The transient system has received a trigger signal and operation is running in channel 3.
5 WTG1-tran	32	The transient system is waiting for a trigger in channel 1.
6 WTG2-tran	64	The transient system is waiting for a trigger in channel 2.
7 WTG3-tran	128	The transient system is waiting for a trigger in channel 3.
8 to 15 Not Used	0	0 is returned.

- The [STATus:PRESet](#) command will set all bits in the PTR register to 1.

### Returned Query Format

<NR1>

The returned value is the binary-weighted sum of all bits set in the register.

### Examples

The following command enables bits 5 (decimal value = 32) and 6 (decimal value = 64) in the PTR register.

```
STAT:OPER:PTR 96
```

The following query returns the bits enabled in the register.

```
STAT:OPER:PTR?
```

Typical Response: +96

### See Also

[STATus:OPERation:NTRansition](#)

[STATus:PRESet](#)

# STATus:PRESet

## Syntax

STATus:PRESet

This command sets all defined bits in the status system's PTR registers and clears all bits in the NTR and Enable registers.

## Parameters

Operation register	Questionable register	Definition
STAT:OPER:ENAB	STAT:QUES:ENAB	0 – all bits disabled.
STAT:OPER:NTR	STAT:QUES:NTR	0 – all bits disabled.
STAT:OPER:PTR		252 – all defined bits enabled.
	STAT:QUES:PTR	16 – all defined bits enabled.

## Example

The following command presets the Operation and Questionable Status registers.

STAT: PRES

# STATus:QUEStionable:CONDition?

## Syntax

STATus:QUEStionable:CONDition?

This query returns the value of the Questionable Status Condition register. The Condition register is a read-only register, which holds the live (unlatched) questionable status of the instrument. Reading the Questionable Condition register does not clear it.

### NOTE

For more information on the SCPI Status System, refer to [Chapter 2, "Status system diagram"](#) on page 11.

## Remarks

The following table lists the bit definitions for the Questionable Status registers.

Bit number	Decimal value	Definition
0 to 3 Not Used	0	0 is returned.
4 OT	16	The over-temperature protection has tripped.
5 to 15 Not Used	0	0 is returned.

## Returned Query Format

<NR1>

The returned value is the binary-weighted sum of all bits set in the register. For example, if bit 4 is set, this query command will return "+16".

## Example

The following query command reads the Questionable Status Condition register.

STAT:QUES:COND?

Typical Response: +16

## STAtus:QUEStionable:ENABle

### Syntax

STAtus:QUEStionable:ENABle <value>

The Enable register is a mask for enabling specific bits from the Questionable Event register to set the questionable summary bit (QUES) of the Status Byte register. This bit (bit 3) is the logical OR of all the Questionable Event register bits that are enabled by the Questionable Status Enable register.

STAtus:QUEStionable:ENABle?

This command reads the value of the Questionable Status Enable register.

### NOTE

For more information on the SCPI Status System, refer to [Chapter 2, “Status system diagram”](#) on page 11.

### Parameters

Item	Type	Range of values	Default value
<value>	NR1	A decimal value which corresponds to the binary-weighted sum of the bits in the register (refer to the table below).	Preset = 0

For example, to enable bit 4, the corresponding decimal value would be 16.

### Remarks

- The following table lists the bit definitions for the Questionable Status registers.

Bit number	Decimal value	Definition
0 to 3 Not Used	0	0 is returned.
4 OT	16	The over-temperature protection has tripped.
5 to 15 Not Used	0	0 is returned.

## 10 STATus Subsystem

STATus:QUEStionable:ENABle

- The [STATus:PRESet](#) command will clear all bits in the enable register.

### Returned Query Format

<NR1>

The returned value is the binary-weighted sum of all bits set in the register.

### Examples

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

```
STAT:QUES:ENAB 16
```

The following query returns the bits enabled in the register.

```
STAT:QUES:ENAB?
```

Typical Response: +16

### See Also

[STATus:QUEStionable\[:EVENT\]?](#)

[STATus:PRESet](#)



## STATus:QUESTionable[:EVENT]?

### Syntax

STATus:QUESTionable[:EVENT]?

This query returns the value of the Questionable Status Event register. The Event register is a read-only register, which stores (latches) all events that are passed by the Questionable NTR and/or PTR filter. Reading the Questionable Status Event register clears it.

### NOTE

For more information on the SCPI Status System, refer to [Chapter 2, "Status system diagram"](#) on page 11.

### Remarks

- The following table lists the bit definitions for the Questionable Status registers.

Bit number	Decimal value	Definition
0 to 3 Not Used	0	0 is returned.
4 OT	16	The over-temperature protection has tripped.
5 to 15 Not Used	0	0 is returned.

- Once a bit is set, it remains set until cleared by reading the event register or the clear status (\*CLS) command.

### Returned Query Format

<NR1>

The returned value is the binary-weighted sum of all bits set in the register. For example, if bit 4 is set, this query command will return "+16".

## 10 STATus Subsystem

STATus:QUEStionable[:EVENT]?

### Example

The following command reads the Questionable Status Event register.

STAT:QUES?

Typical Response: +0

### See Also

[STATus:QUEStionable:ENABLE](#)

[\\*CLS](#)

## STATus:QUEStionable:NTRansition

### Syntax

STATus:QUEStionable:NTRansition <value>

This register serves as a polarity filter between the Question Condition and Question Event registers. When a bit in the Question NTR register is set to 1, then a 1-to-0 transition of the corresponding bit in the Question Condition register causes that bit in the Question Event register to be set.

STATus:QUEStionable:NTRansition?

These reads the value of the Question NTR (Negative-Transition) register.

### NOTE

For more information on the SCPI Status System, refer to [Chapter 2, “Status system diagram”](#) on page 11.

### Parameter

Item	Type	Range of values	Default value
<value>	NR1	A decimal value which corresponds to the binary-weighted sum of the bits in the register (refer to the table below).	Preset = 0

For example, to enable bit 4, the corresponding decimal value would be 16.

### Remarks

- The following table lists the bit definitions for the Questionable Status registers.

Bit number	Decimal value	Definition
0 to 3 Not Used	0	0 is returned.
4 OT	16	The over-temperature protection has tripped.
5 to 15 Not Used	0	0 is returned.

## 10 STATus Subsystem

STATus:QUEStionable:NTRansition

- The [STATus:PRESet](#) command will set all bits in the NTR register to 0.

### Returned Query Format

<NR1>

The returned value is the binary-weighted sum of all bits set in the register.

### Examples

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

```
STAT:QUES:NTR 16
```

The following query returns the bits enabled in the register.

```
STAT:QUES:NTR?
```

Typical Response: +16

### See Also

[STATus:QUEStionable:PTRansition](#)

[STATus:PRESet](#)

# STATUS:QUESTIONABLE:PTRansition

## Syntax

STATUS:QUESTIONABLE:PTRansition <value>

This register serves as a polarity filter between the Questionable Condition and Questionable Event registers. When a bit in the Questionable PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Questionable Condition register causes that bit in the Questionable Event register to be set.

STATUS:QUESTIONABLE:PTRansition?

These command reads the value of the Questionable PTR (Positive-Transition) register.

NOTE

For more information on the SCPI Status System, refer to [Chapter 2, “Status system diagram”](#) on page 11.

## Parameter

Item	Type	Range of values	Default value
<value>	NR1	A decimal value which corresponds to the binary-weighted sum of the bits in the register (refer to the table below).	Preset = 16

For example, to enable bit 4, the corresponding decimal value would be 16.

## Remarks

- The following table lists the bit definitions for the Questionable Status registers.

Bit number	Decimal value	Definition
0 to 3 Not Used	0	0 is returned.
4 OT	16	The over-temperature protection has tripped.

# 10 STATus Subsystem

STATus:QUEStionable:PTRansition

5 to 15 Not Used	0	0 is returned.
------------------	---	----------------

- The [STATus:PRESet](#) command will set all bits in the PTR register to 1.

## Returned Query Format

<NR1>

The returned value is the binary-weighted sum of all bits set in the register.

## Examples

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

STAT:QUES:PTR 16

The following query returns the bits enabled in the register.

STAT:QUES:PTR?

Typical Response: +16

## See Also

[STATus:QUEStionable:NTRansition](#)

[STATus:PRESet](#)



## 11 SYSTem Subsystem

SYSTem:CDEscription? [104](#)  
SYSTem:CHANnel[:COUNT]? [105](#)  
SYSTem:ERRor? [106](#)  
SYSTem:LFRequency [107](#)  
SYSTem:VERSion? [108](#)

The SYSTem command subsystem is used to return slot description and channel count of the U2722A/U2723A when used in a chassis. This subsystem also returns the error numbers and messages from the U2722A/U2723A as well as sets the line frequency and queries the SCPI version.



## SYSTem:CDEscription?

### Syntax

SYSTem:CDEscription?

This query is used when the SMU device is slotted into modular instrument chassis. It identifies which slot is the module plugged into.

### Remarks

- It is only applicable when the U2722A/U2723A is used in the modular instrument chassis.
- If the U2722A/U2723A is not plugged into the chassis, then the returned query value would be "+7, +0", where "+7" is an invalid slot number and "+0" is the chassis number.
- Please refer to the *U2781A Modular Instrument Chassis User's Guide* for the details.

### Returned Query Format

<NR1>, <NR1>

### Examples

The following shows the query when the U2722A/U2723A is slotted into slot 4 of chassis #1.

SYST:CDES?

Typical Response: +4, +1

Below is the query when the U2722A/U2723A is plugged into slot 2 of chassis #2.

SYST:CDES?

Typical Response: +2, +2



## SYSTem:CHANnel[:COUNT]?

### Syntax

SYSTem:CHANnel[:COUNT]?

This query returns the number of output channel.

### Returned Query Format

<NR1>

### Example

The following query returns the number of output channels.

SYST:CHAN?

Typical Response: +3

### See Also

[SYSTem:CDEscription?](#)

## SYSTem:ERRor?

### Syntax

SYSTem:ERRor?

This query returns the next error number and its corresponding message string from the error queue. Please refer to [Chapter 14, “Error Messages”](#) on page 130 for the error codes.

### Remarks

- Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored.
- Errors are cleared as you read them. When you have read all the errors from the error queue, the power indicator turns off and the errors are cleared.
- The error queue is cleared by the clear status (\*CLS) command and when power is cycled. The errors are also cleared when you read the error queue.

### Returned Query Format

<NR1>, <SRD>

### Example

SYST:ERR?

Typical Response: -101, "Invalid character"

# SYSTem:LFRrequency

## Syntax

SYSTem:LFRrequency <mode>

This command sets the power line frequency in the measurement. The value is coupled to APERTure by the equation:

$$APERTure = NPLCycles / selected\ line\ frequency$$

SYSTem:LFRrequency?

Queries the power line frequency set.

## Parameter

Item	Type	Range of values	Default value
<mode>	CPD	F50HZ: 50 Hz frequency line	F50HZ
		F60HZ: 60 Hz frequency line	

## Returned Query Format

<CRD>

## Examples

The following command sets the line frequency to 60 Hz.

SYST:LFR F60HZ

The following query returns the line frequency.

SYST:LFR?

Typical Response: F60Hz

## **SYSTem:VERSion?**

### **Syntax**

SYSTem:VERSion?

This command returns the version of the SCPI (Standard Commands for Programmable Instruments) standard in which the instrument complies with.

### **Returned Query Format**

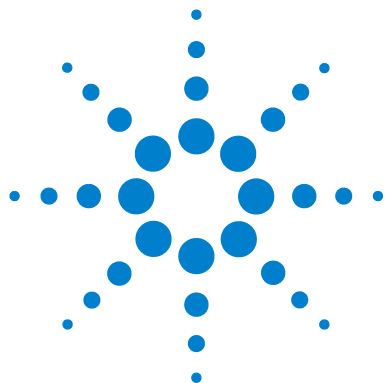
<SRD>

The command returns a string in the form of "YYYY.V", where YYYY represents the year of the version and V represents a version for that year.

### **Example**

SYST:VERS?

Typical Response: "1997.0"



## 12 TRIGger Subsystem

TRIGger:SOURce 110

The chapter describes the TRIGger command subsystem which controls the remote triggering of the instrument. These commands specify the trigger source for the transient and measurement system, and also generate software triggers.



## TRIGger:SOURce

### Syntax

TRIGger:SOURce <NONE|STRG>

Selects the trigger source to NONE or STRG.

TRIGger:SOURce?

Queries the present trigger source.

### Returned Query Format

<CRD>

### Examples

The following command selects the trigger source to NONE.

TRIG:SOUR NONE

The following query returns the trigger source.

TRIG:SOUR?

Typical Response: NONE



## 13 IEEE-488.2 Common Commands

\*CAL? 112  
\*CLS 113  
\*ESE/\*ESE? 114  
\*ESR? 116  
\*IDN? 118  
\*OPC/OPC? 119  
\*RST 120  
\*SRE/SRE? 121  
\*STB? 124  
\*TST? 126  
\*WAI 127

This chapter contains information on the IEEE-488.2 Common (\*) Commands supported by the U2722A/U2723A. It also describes the universal command statements which form the nucleus of the GPIB programming understood by all instruments in the network. When combined with the programming language codes, the commands provide all management and data communication instructions for the system



## **\*CAL?**

### **Syntax**

\*CAL?

This command performs a self-calibration of the instrument and returns a pass/fail indication.

### **Returned Query Format**

<NR1>

The command returns "+0" (calibration test passed) or "+1" (calibration test failed).

### **Example**

\*CAL?

Typical Response: +0



## \*CLS

### Syntax

\*CLS

This command is used to clear the event registers in all register groups and also clears the error queue.

### Example

The following command clears the event register bits.

\*CLS

\*ESE/\*ESE?

# \*ESE/\*ESE?

## Syntax

\*ESE <enable\_value>

This command 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.

\*ESE?

This will query the Standard Event register group.

## Parameter

Item	Type	Range of values	Default value
<enable_value>	NRF	A decimal value that corresponds to the binary-weighted sum of the bits in the register. Refer to the table below.	This is a required parameter.

## Remarks

- The following table lists the bit definitions for the Standard Event register.

Bit number	Decimal value	Definition
0 Operation Complete	1	All commands prior to and including *OPC have been executed.
1 Not Used	2	Always zero.
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-Dependent Error	8	A self-test or calibration error occurred. For a complete listing of the error messages, refer to <a href="#">Chapter 14, “Error Messages”</a> on page 130.
4 Execution Error	16	An execution error occurred (an error in the –200 range has been generated).
5 Command Error	32	A command syntax error occurred (an error in the –100 range has been generated).
6 Not Used	64	Always zero.

7 Power On	128	Power has been turned off and on since the last time the event register was read or cleared.
------------	-----	--

- Use the <enable\_value> parameter to specify which bits will be enabled. The specified decimal value corresponds to the binary-weighted sum of the bits you wish to enable in the register. For example, to enable bit 2 (decimal value = 4), bit 4 (decimal value = 16), and bit 5 (decimal value = 32), the corresponding decimal value would be 52 (4 + 16 + 32).
- The clear status (\*CLS) command will not clear the enable register but it clears all bits in the event register.

Returned Query Format

<NR1>

The query command 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".

Examples

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

\*ESE 16

The following query returns the bits enabled in the register.

\*ESE?

Typical Response: +16

See Also

\*ESR?

**\*ESR?****Syntax**

\*ESR?

This query returns the value of the event register of the Standard Event Status group. Once it is read, it will be cleared automatically. The event register is a read-only register, which stores (latches) all standard events.

**Remarks**

- The following table lists the bit definitions for the Standard Event register.

Bit number	Decimal value	Definition
0 Operation Complete	1	All commands prior to and including *OPC have been executed.
1 Not Used	2	Always zero.
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-Dependent Error	8	A self-test or calibration error occurred. For a complete listing of the error messages, refer to <a href="#">Chapter 14, “Error Messages”</a> on page 130.
4 Execution Error	16	An execution error occurred (an error in the –200 range has been generated).
5 Command Error	32	A command syntax error occurred (an error in the –100 range has been generated).
6 Not Used	64	Always zero.
7 Power On	128	Power has been turned off and on since the last time the event register was read or cleared.

- To be reported to the Standard Event register, the corresponding bits in the event register must be enabled using the [\\*ESE/\\*ESE?](#) command.
- Once a bit is set, it will remain set until cleared by reading the event register or the clear status ([\\*CLS](#)) command.

**Returned Query Format**

<NR1>

The query command 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".

**Example**

The following query returns the bits enabled in the register.

\*ESR?

Typical Response: +16

**See Also**

[\\*ESE/\\*ESE?](#)

## \*IDN?

### Syntax

\*IDN?

This command reads the instrument's identification string which contains four comma-separated fields. The first field is the manufacturer's name, the second is the model number of the instrument, the third is the serial number, and the fourth is the firmware revision code with the measurement firmware version.

### Returned Query Format

<AARD>

The command returns a string with the following format.

AGILENT TECHNOLOGIES,U2722A/U2723A,<Serial Number>,Rf.ff-c.cc

f.ff = firmware revision number

c.cc = measurement firmware version

### Example

The following query returns the instrument's identification string.

\*IDN?

Typical Response:

AGILENT TECHNOLOGIES,U2722A/U2723A,MY12345678,R1.00-1.00

## \*OPC/OPC?

### Syntax

\*OPC

The command is mainly used for program synchronization. It causes the instrument to set the OPC bit (bit 0) of the Standard Event Status register when the instrument has completed all pending operation sent before the \*OPC command.

\*OPC?

This query returns "1" to the output buffer at the completion of the current operation. \*OPC? command does not suspend processing of commands.

### Remarks

Pending operation have completed when:

- all commands sent before \*OPC, including paralleled commands, have completed. Most commands are sequential and will be completed before the next command is executed. Commands that affect the output voltage, current, or state, relays, and trigger actions are executed in parallel with subsequent commands. \*OPC provides notification for the completion of the parallel commands
- all triggered actions have completed

\*OPC does not prevent processing of subsequent commands, however, the OPC bit will not be set until all pending operation have completed.

### Returned Query Format

<NR1>

The command returns "1" to the output buffer.

### See Also

[\\*ESE/\\*ESE?](#)

[\\*ESR?](#)

## **\*RST**

### **Syntax**

\*RST

This command resets the instrument to the default configuration. Refer to [Chapter 15, “Default Factory Settings”](#) on page 137 for a complete listing of the instrument's default configuration.

### **Example**

The following command resets the instrument.

\*RST



\*SRE/SRE?

Syntax

\*SRE <enable\_value>

This command enables the 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.

\*SRE?

The query command 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 2 (decimal value = 4) and bit 6 (decimal value = 64) are enabled, the query command will return "+68".

Parameter

Item	Type	Range of values	Default value
<enable_value>	NRF	A decimal value that corresponds to the binary-weighted sum of the bits in the register. Refer to the table below.	This is a required parameter.

**Remarks**

- The following table lists the bit definitions for the Status Byte register.

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

- Use the `<enable_value>` parameter to specify which bits will be enabled. The specified decimal value corresponds to the binary-weighted sum of the bits you wish to enable in the register. For example, to enable bit 3 (decimal value = 8) and bit 6 (decimal value = 64), the corresponding decimal value would be 72 (8 + 64).
- The Status Byte Enable register will be cleared when you execute the `*SRE 0` command.

**Returned Query Format**

`<NR1>`

The query command 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".

**Examples**

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

\*SRE 16

The following query returns the bits enabled in the register.

\*SRE?

Typical Response: +16

**See Also**

[\\*STB?](#)

# **\*STB?**

## **Syntax**

\*STB?

Queries the condition register for the Status Byte register group. This query command is similar to a Serial Poll but it is processed like any other instrument command. This is a read-only register and the bits will not be cleared once they are read.

## **Remarks**

- The following table lists the bit definitions for the Status Byte register.

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

- Use the <enable\_value> parameter to specify which bits will be enabled. The specified decimal value corresponds to the binary-weighted sum of the bits you wish to enable in the register. For example, to enable bit 3 (decimal value = 8) and bit 6 (decimal value = 64), the corresponding decimal value would be 72 (8 + 64).
- The Status Byte Enable register will be cleared once you execute the \*SRE 0 command.

**Returned Query Format**

<NR1>

The command will return "1" to the output buffer.

**Example**

The following command reads the condition register (bits 3 and 4 are set).

\*STB?

Typical Response: +24

**See Also**

[\\*CLS](#)

`*TST?`

## **\*TST?**

### **Syntax**

`*TST?`

This query command performs a self-test of the instrument and returns a pass/fail indication.

### **Remarks**

Please refer to *the U2722A/U2723A USB Modular Source Measure Units User's Guide* for the self-test return codes.

### **Returned Query Format**

`<NR1>`

The command returns "+0" (all tests passed) or a decimal value which is larger than 0 (one or more tests failed).

### **Example**

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

`*TST?`

Typical Response: +0

## \*WAI

### Syntax

\*WAI

This command configures the instrument to wait for the completion of all pending operation before executing any additional commands over the interface.

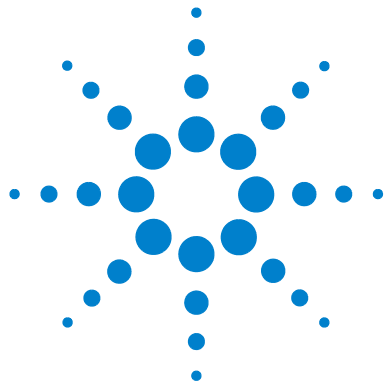
### See Also

[\\*OPC/OPC?](#)

**13 IEEE-488.2 Common Commands**

\*WAI





## 14 Error Messages

Error Messages [130](#)

Error List [130](#)

The U2722A/U2723A SCPI command errors are summarized in this chapter.



# Error Messages

Error messages are created once a command error or an erroneous condition has been detected.

- Errors are retrieved in first-in, first-out (FIFO) order.
- Errors are cleared as you read them.
- No additional errors will be stored until you remove the errors from the queue. If no errors occur when you read the error queue, the instrument will respond with 0, "No error".
- [SYSTem:ERRor?](#) will read and clear one error from the queue.

## Error List

The table below contains the device-dependent errors.

Table 14-1 Device-dependent errors

Error	Device-dependent errors (these errors set the Standard Event Status register bit #3)
0	<b>No error</b> This is the response to the <a href="#">SYSTem:ERRor?</a> query when there are no errors.
+101	<b>Calibration state is off</b> Calibration is not enabled. The instrument will not accept calibration commands.
+103	<b>Calibration voltage or current range is incorrect</b> Calibration voltage and current ranges are set incorrectly. Calibration cannot be executed.
+104	<b>Bad sequence of calibration commands</b> Calibration commands have not been entered in the proper sequence.
+120	<b>Configuration Multiple slave not allowed</b> Multiple slaves are not allowed in the configuration.
+121	<b>Configuration master not allowed</b> Status master is not allowed in the configuration.
+246	<b>Master Controller Busy</b> The channel is busy. All input commands will be ignored.

**Table 14-1** Device-dependent errors (continued)

+249	<b>Over Temperature Occurred</b> Over temperature has been detected. Please remove the load or switch off the device.
+254	<b>Memory List Full</b> The active memory lists is full. No commands can be inserted anymore.

The following table shows the list of command errors.

**Table 14-2** Command errors

Error	Command errors (these errors set the Standard Event Status register bit #5)
–100	<b>Command error</b> Generic syntax error.
–101	<b>Invalid character</b> 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	<b>Syntax error</b> 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	<b>Invalid separator</b> An invalid separator was found in the command string. Check for proper usage of , ; :
–104	<b>Data type error</b> 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	<b>GET not allowed</b> A group execute trigger is not allowed in a command string.

**Table 14-2** Command errors (continued)

Error	Command errors (these errors set the Standard Event Status register bit #5)
–108	<b>Parameter not allowed</b> More parameters were received than expected for the command. You may have entered an extra parameter, or you have added a parameter to a command that does not accept a parameter.
–109	<b>Missing parameter</b> Fewer parameters were received than expected for the command. You may have omitted one or more parameters that are required for this command.
–110	<b>Command header error</b> An error was detected in the header.
–111	<b>Header separator error</b> A character that was not a valid header separator was found in the command string.
–112	<b>Program mnemonic too long</b> The header contains more than 12 characters.
–113	<b>Undefined header</b> 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	<b>Header suffix out of range</b> The value of the numeric suffix is not valid.
–120	<b>Numeric data error</b> Generic numeric data error.
–121	<b>Invalid character in number</b> An invalid character was found in the number specified for a parameter value. Example: *ESE #2
–123	<b>Exponent too large</b> A numeric parameter was found where its exponent was larger than 32,000.
–124	<b>Too many digits</b> The mantissa of a numeric parameter contained more than 255 digits, excluding leading zeros.
–128	<b>Numeric data not allowed</b> A numeric parameter was received but a character string was expected.

**Table 14-2** Command errors (continued)

Error	Command errors
–130	<b>Suffix error</b> Generic suffix error.
–131	<b>Invalid suffix</b> A suffix was incorrectly specified for a numeric parameter.
–134	<b>Suffix too long</b> The suffix contains more than 12 characters.
–138	<b>Suffix not allowed</b> A suffix is not supported for this command.
–140	<b>Character data error</b> Generic character data error.
–141	<b>Invalid character data</b> Either the character data element contains an invalid character, or the element is not valid.
–144	<b>Character data too long</b> The character data element contains more than 12 characters.
–148	<b>Character data not allowed</b> A discrete parameter was received, but a string or numeric parameter was expected.
–150	<b>String data error</b> Generic string data error.
–151	<b>Invalid string data</b> An invalid character string was received. Check that the string is enclosed in quotation marks.
–158	<b>String data not allowed</b> A character string was received, but is not allowed for this command.
–160	<b>Block data error</b> Generic block data error.
–161	<b>Invalid block data</b> The number of data bytes sent does not match the number of bytes specified in the header.
–168	<b>Block data not allowed</b> Data was sent in arbitrary block format but is not allowed for this command.

**Table 14-2** Command errors (continued)

Error	Command errors
–170	<b>Expression error</b> Generic expression error.
–171	<b>Invalid expression</b> The expression data element was invalid.
–178	<b>Expression data not allowed</b> Expression data element was sent but is not allowed for this command.

The execution errors are listed in the table below.

**Table 14-3** Execution errors

Error	Execution errors (these errors set the Standard Event Status register bit #4)
–200	<b>Execution error</b> Generic syntax error.
–210	<b>Trigger error</b> Generic trigger error.
–211	<b>Trigger ignored</b> A Group Execute Trigger (GET) was received but the trigger was ignored.
–220	<b>Parameter error</b> A data element related error occurred.
–221	<b>Settings conflict</b> A data element could not be executed because of the present instrument state.
–222	<b>Data out of range</b> A data element could not be executed because the value was outside the valid range.
–223	<b>Too much data</b> A data element was received that contains more data than the instrument can handle.
–224	<b>Illegal parameter value</b> An exact value was expected but not received.

**Table 14-3** Execution errors (continued)

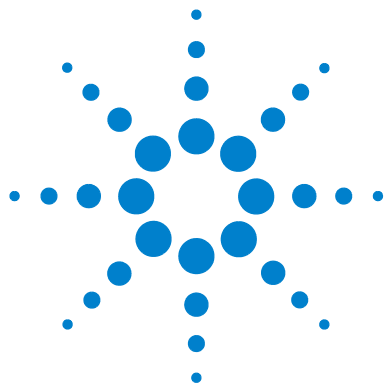
Error	Execution errors (these errors set the Standard Event Status register bit #4)
-225	<b>Out of memory</b> The device has insufficient memory to perform the requested operation.
-230	<b>Data corrupt or stale</b> Possibly an invalid data. A new reading was started but not completed.
-231	<b>Data questionable</b> The measurement accuracy is in question.
-240	<b>Hardware error</b> The command could not be executed due to a hardware problem with the instrument.
-241	<b>Hardware missing</b> The command could not be executed due to missing hardware, such as an option.
-260	<b>Expression error</b> An expression program data element related error occurred.
-261	<b>Math error in expression</b> An expression program data element could not be executed due to a math error.

The list of query errors is shown in the following table.

**Table 14-4** Query errors

Error	Query errors (these errors set the Standard Event Status register bit #2)
–400	<b>Query error</b> Generic error query.
–410	<b>Query INTERRUPTED</b> A condition causing an interrupted query error occurred.
–420	<b>Query UNTERMINATED</b> A condition causing an unterminated query error occurred.
–430	<b>Query DEADLOCKED</b> A condition causing a deadlocked query error occurred.
–440	<b>Query UNTERMINATED after indefinite response</b> A query was received in the same program message after a query indicating an indefinite response was executed.





## 15 Default Factory Settings

Factory Default Settings [138](#)

The default factory settings are listed in this chapter.

### Remarks

- These settings are used when the U2722A/U2723A modular SMU is powered on.
- They are set once the `*RST` command is sent to the U2722A/U2723A.



## Factory Default Settings

The following table shows the list of the factory default settings.

**Table 15-1** List of factory default settings

SCPI Commands	Default value
CONFigure:SSI	NONE, 0
*ESE	0
*OPC	0
OUTPut	0
SENSe:CURRent[:DC]:NPLCycles	0
SENSe:SWEep:POINts	1024
SENSe:SWEep:TINterval	1
SENSe:VOLTage[:DC]:NPLCycles	0
[SOURce:]CURRent	0
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]	0
[SOURce:]CURRent:LIMit	0.0000001
[SOURce:]CURRent:RANGe	R1uA
[SOURce:]MEMory:CONFigure:POINts	1,1,1
[SOURce:]MEMory:LIST	1
[SOURce:]VOLTage	0
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]	0
[SOURce:]VOLTage:LIMit	0.2
[SOURce:]VOLTage:RANGe	R2V
STATus:OPERation:ENABle	0
STATus:OPERation:NTRansition	0
STATus:OPERation:PTRansition	252
STATus:QUEStionable:ENABle	0

**Table 15-1** List of factory default settings (continued)

SCPI Commands	Default value
STATus:QUESTionable:NTRansition	0
STATus:QUESTionable:PTRansition	16
*SRE	0
SYSTem:LFRequency	F50HZ
TRIGger:SOURce	NONE

**15    Default Factory Settings**

Factory Default Settings

**www.agilent.com**

### **Contact us**

To obtain service, warranty or technical support assistance, contact us at the following phone numbers:

United States:

(tel) 800 829 4444 (fax) 800 829 4433

Canada:

(tel) 877 894 4414 (fax) 800 746 4866

China:

(tel) 800 810 0189 (fax) 800 820 2816

Europe:

(tel) 31 20 547 2111

Japan:

(tel) (81) 426 56 7832 (fax) (81) 426 56

7840

Korea:

(tel) (080) 769 0800 (fax) (080) 769 0900

Latin America:

(tel) (305) 269 7500

Taiwan:

(tel) 0800 047 866 (fax) 0800 286 331

Other Asia Pacific Countries:

(tel) (65) 6375 8100 (fax) (65) 6755 0042

Or visit Agilent worldwide web at:

[www.agilent.com/find/assist](http://www.agilent.com/find/assist)

Product specifications and descriptions in this document subject to change without notice.

© Agilent Technologies, Inc., 2008 - 2009

Second Edition, May 22, 2009

U2722-90022



**Agilent Technologies**