Keysight N778-C Series Polarization Instruments

N7781C Polarization Analyzer
N7785C Synchronous Scrambler
N7786C Polarization Synthesizer
N7788C Optical Component Analyzer

Programming Guide

KEYSIGHT
TECHNOLOGIES

Notices

Manual Part Number

N7780-90C02

Edition

Edition 2.0, October 2021

Keysight Technologies Deutschland GmbH
Herrenberger Strasse 130,
71034 Böblingen, Germany

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.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement

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

Warranty

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

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.

Safety Summary

The following general safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or with specific warnings or operating instructions in the product manuals violates safety standards of design, manufacture, and intended use of the instrument. Keysight Technologies assumes no liability for the customer's failure to comply with these requirements. Product manuals are provided on the Web. Go to www.keysight.com and type in your product number in the Search field at the top of the page.

General

This product is a Protection Class 1 instrument (provided with a protective earth terminal) and has been manufactured and tested according to international safety standards. The protective features of this product may be impaired if it is used in a manner not specified in the operation instructions.

All Light Emitting Diodes (LEDs) used in this product are Class 1 LEDs as per IEC 60825-1:2014.

Environment Conditions

This instrument is intended for indoor use in an Overvoltage Category II, pollution degree 2 environment. It is designed to operate at a maximum relative humidity of 85% RH, non-condensing and at altitudes of up to 2000 meters. Refer to the specifications tables for the AC mains voltage requirements and ambient operating temperature range.

Temperature

The instrument should be protected from temperature extremes and changes in temperature that may cause condensation within it.

The operating temperature is from 5 °C to +40 °C.

The storage temperature is from -40 °C to +70 °C.

Before Applying Power

Verify that all safety precautions are taken. The power cable inlet of the instrument serves as a device to disconnect from the mains in case of hazard. The instrument must be positioned so that the operator can easily access the power cable inlet. When the instrument is rack mounted the rack must be provided with an easily accessible mains switch.

Ground the Instrument

To minimize shock hazard, the instrument chassis and cover must be connected to an electrical protective earth ground. The instrument must be connected to the AC power mains through a grounded power cable, with the ground wire firmly connected to an electrical ground (safety ground) at the power outlet. Any interruption of the protective (grounding) conductor or disconnection of the protective earth terminal will cause a potential shock hazard that could result in personal injury.

Do Not Operate in an Explosive Atmosphere

Do not operate the instrument in the presence of flammable gases or fumes.

Do Not Remove the Instrument Cover

Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made only by qualified personnel.

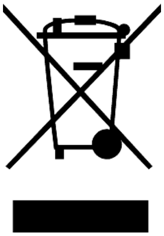
Instruments that appear damaged or defective should be made inoperative and secured against unintended operation until they can be repaired by qualified service personnel.

Instrument Markings

Instrument Marking	Description
⚠	The instruction manual symbol. The product is marked with this warning symbol when it is necessary for the user to refer to the instructions in the manual.
⏻	Standby supply. Unit is not completely disconnected from AC mains when switch is off.
ICES/NMB-001 CE ISM GRP 1-A	The CE mark is a registered trademark of the European Community.
SA ® C US	The CSA mark with the 'c' and 'us' subscript indicates the instrument is certified to the applicable Canadian and United States of America standards respectively.
✓	The RCM mark is a registered trademark of the Australian Communications and Media Authority
KC R-R-Kst- 3E 18526	This symbol is a South Korean Class A EMC Declaration, with the product identification code "R-R-Kst-3E18526". R - Identification of authorization prefix. R - Identification of basic certification information. Kst - Identification of applicant's information 3E18526 - Product identification. This is a Class A instrument suitable for professional use and in electromagnetic environment outside of the home.
♲	The recycling symbol indicates the general ease with which the instrument can be recycled.
40	China Restricted Substance Product Label. The EPUP (environmental protection use period) number in the center indicates the time period during which no hazardous or toxic substances or elements are expected to leak or deteriorate during normal use and generally reflects the expected useful life of the product.

Compliance and Environmental Information

Table 1 Compliance and Environmental Information

Safety Symbol	Description
	This product complies with WEEE Directive (2002/96/EC) marking requirements. The affixed label indicates that you must not discard this electrical/electronic product in domestic household waste. Product Category: With reference to the equipment types in WEEE Directive Annex I, this product is classed as a “Monitoring and Control instrumentation” product. Do not dispose in domestic household waste. To return unwanted products, contact your local Keysight office, or see http://about.keysight.com/en/companyinfo/environment/takeback.shtml for more information.

Declaration of Conformity

Declarations of Conformity for this product and for the Keysight products may be downloaded from the Web. Go to http://www.keysight.com/go/conformity.

You can then search by product number to find the latest Declaration of Conformity.

Contents

1 Introduction to Programming

This chapter provides general information on how to control your instrument remotely.

Descriptions for the actual commands for the instruments are given in the following chapters. The information in these chapters is specific to the N778-C polarization instruments.

Message Queues

The instrument exchanges messages using an input and an output queue. Error messages are kept in a separate error queue.

How the Input Queue Works

The input queue is a FIFO queue (first-in first-out). Incoming bytes are stored in the input queue. The parser starts if the LF character is received.

Clearing the Input Queue

Switching the power off, or sending a Device Interface Clear signal, causes commands that are in the input queue, but have not been executed to be lost.

The Output Queue

The output queue contains responses to query messages. The instrument transmits any data from the output queue when a controller addresses the instrument as a talker.

Each response message ends with a LF ($0A_{16}$). If no query is received, or if the query has an error, the output queue remains empty.

The Message Available bit (MAV, bit 4) is set in the Status Byte register whenever there is data in the output queue.

The Error Queue

The error queue is 30 errors long. It is a FIFO queue (first-in first-out). That is, the first error read is the oldest error to have occurred. For example:

1. If no error has occurred, the error queue contains:
   + 0, "No error"
2. After a command such as wav:pow, the error queue now contains:
   + 0, "No error"
   -113, "Undefined header"
3. If the command is immediately repeated, the error queue now contains:
   + 0, "No error"
   -113, "Undefined header"
   -113, "Undefined header"

If more than 29 errors are put into the queue, the message:

-350, "Queue overflow"

is placed as the last message in the queue.

Programming and Syntax Diagram Conventions

A program message is a message containing commands or queries that you send to the instruments. The following are a few points about program messages:

- You can use either upper-case or lower-case characters.
- You can send several commands in a single message. Each command must be separated from the next one by a semicolon (;).
- A command message is ended by a line feed character (LF).
- You can use any valid number/unit combination.

  In other words, 1500NM,1.5UM and 1.5E-6M are all equivalent.

  If you do not specify a unit, then the default unit is assumed. The default unit for the commands are given with command description in the next chapter.

Short Form and Long Form

The instrument accepts messages in short or long forms.

For example, the message

```
:STATUS:OPERATION:ENABLE 768
```

is in long form.

The short form of this message is

```
:STAT:OPER:ENAB 768
```

In this manual, the messages are written in a combination of upper and lower case. Upper case characters are used for the short form of the message.

For example, the above command would be written

```
:STATus:OPERation:ENABle
```

The first colon can be left out for the first command or query in your message. That is, the example given above could also be sent as

```
STAT:OPER:ENAB 768
```

Command and Query Syntax

All characters not between angled brackets must be sent exactly as shown.

The characters between angled brackets (<...>) indicate the kind of data that you should send, or that you get in a response. You do not type the angled brackets in the actual message.

Descriptions of these items follow the syntax description. The following types of data are most commonly used:

string	is ascii data. A string is contained between double quotes ("...") or single quotes ('...').
value	is numeric data in integer (12), decimal (34.5) or exponential format (67.8E-9).
wsp	is a white space.

Other kinds of data are described as required.

The characters between square brackets ([...]) show optional information that you can include with the message.

The bar (|) shows an either-or choice of data, for example, *a*|*b* means either *a* or *b*, but not both simultaneously.

Extra spaces are ignored, so spaces can be inserted to improve readability.

Units

Where units are given with a command, usually only the base units are specified. The full sets of units are given in the table below.

Table 2 Units and allowed Mnemonics

Unit	Default	Allowed Mnemonics
meters	M	PM, NM, UM, MM, M
decibel	DB	MDB, DB
second	S	NS, US, MS, S
decibel/1mW	DBM	MDBM, DBM
Hertz	HZ	HZ, KHZ, MHZ, GHZ, THZ
Watt	Watt	PW, NW, UW, MW, Watt
meters per second	M/S	NM/S, UM/S, MM/S, M/S

Data Types

With the commands you give parameters to the instrument and receive response values from the instrument. Unless explicitly specified these data are given in ASCII format. The following types of data are used:

- *Boolean* data may only have the values 0 or 1.
- *Integer* range is given for each individual command.
- *Float* variables may be given in decimal or exponential writing (0.123 or 123E-3).
  All *Float* values conform to the 32 bit IEEE Standard, that is, all *Float* values are returned as 32-bit real values.
- A *string* is contained between double quotes ("...") or single quotes (‘...’). When the instrument returns a string, it is always included in " ".
- When a *register* value is given or returned (for example *ESE), the *decimal* values for the single bits are added. For example, a value of nine means that bit 0 and bit 3 are set.
- Larger blocks of data are given as *Binary Blocks*, preceded by “#<*H*><*Len*><*Block*>”, <*H*> represents the number of digits, <*Len*> represents the number of bytes, and <*Block*> is the data block. For example, for a *Binary Block* with 1 digit and 6 bytes this is: *#16TRACES*. The block represents an array of numbers. Each number has the byte ordering least significant byte first, also called LSBfirst, little-endian or Intel byte ordering.

NOTE **Note that within your program, calculations with wavelengths may require double-precision 64-bit floats to provide the desired resolution.**

Common Commands

The IEEE 488.2 standard has a list of reserved commands, called common commands. Some of these commands must be implemented by any instrument using the standard, others are optional.

Your instrument implements all the necessary commands, and some optional ones. This section describes the implemented commands.

Common Command Summary

Table 3 on page -15 provides a summary of the common commands.

Table 3 Common Command Summary

Command	Parameter	Function	Page
*CLS		Clear Status Command	page 28
*ESE		Standard Event Status Enable Command	page 28
*ESE?		Standard Event Status Enable Query	page 29
*ESR?		Standard Event Status Register Query	page 29
*IDN?		Identification Query	page 30
*OPC		Operation Complete Command	page 30
*OPC?		Operation Complete Query	page 31
*OPT?		Options Query	page 31
*RST		Reset Command	page 31
*STB?		Read Status Byte Query	page 32
*TST?		Self Test Query	page 32
*WAI		Wait Command	page 33

NOTE

These commands are described in more detail in IEEE-Common Commands on page 28.

Common Status Information

There are three registers for the status information. Two of these are status-registers and one is an enable-registers. These registers conform to the IEEE Standard 488.2-1987. You can find further descriptions of these registers under ***ESE, *ESR?**, and ***STB?**.

Figure 1 shows how the Standard Event Status Enable Mask (SESEM) and the Standard Event Status Register (SESR) determine the Event Status Bit (ESB) of the Status Byte.

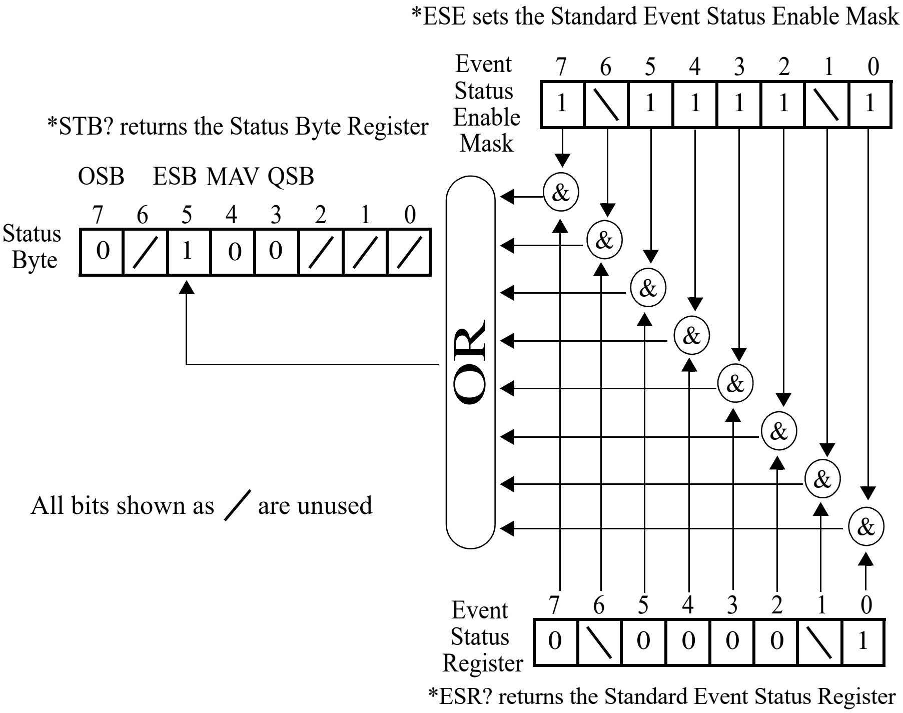


Figure 1 The Event Status Bit

The SESR contains the information about events that are not slot specific.

The SESEM allows you to choose the event that may affect the ESB of the Status Byte. If you set a bit of the SESEM to zero, the corresponding event cannot affect the ESB. The default is for all the bits of the SESEM to be set to 0.

NOTE

Unused bits in any of the registers change to 0 when you read them.

Keysight N778-C Series Polarization Instruments
Programming Guide

2 Command Summary

This chapter lists commands relating to the N778-C series polarization instruments.

Each of these summaries contains a page reference for more detailed information about the particular command later in this manual.

Command Summary

The commands are ordered in a command tree. Every command belongs to a node in this tree.

The root nodes are also called the subsystems. A subsystem contains all commands belonging to a specific topic. In a subsystem there may be further subnodes.

Table 4 on page 20 gives an overview of the command tree. You see the nodes, the subnodes, and the included commands.

Table 4 Command Summary

Command	Page
CONFigure Commands	
:CONFigure:MEASurement:SETTing:ACTual?	Page 52
:CONFigure:MEASurement:SETTing:NUMBer?	Page 52
:CONFigure:MEASurement:SETTing:PRESet	Page 52
:CONFigure:MEASurement:SETTing:CANCel	Page 53
:CONFigure:MEASurement:SETTing:RECall	Page 53
:CONFigure:MEASurement:SETTing:SAVE	Page 53
:CONFigure:MEASurement:SETTing:ERASe	Page 54

Command	Page
Polarization Analyzer Commands	
:POLarimeter:SOP?	Page 58
:POLarimeter:SOP:FETCh?	Page 58
:POLarimeter:POWer?	Page 58
:POLarimeter:POWer:FETCh?	Page 59
:POLarimeter:POWer:UNIT?	Page 59
:POLarimeter:POWer:UNIT	Page 59

Command	Page
:POLarimeter:WAVelength	Page 60
:POLarimeter:WAVelength?	Page 60
:POLarimeter:GAIN	Page 60
:POLarimeter:GAIN?	Page 61
:POLarimeter:AGFLag	Page 61
:POLarimeter:AGFLag?	Page 61
:POLarimeter:STOP	Page 62
:POLarimeter:ZERO?	Page 62
:POLarimeter:SWEep:LOOP	Page 62
:POLarimeter:SWEep:LOOP?	Page 63
:POLarimeter:SWEep:STARt	Page 63
POLarimeter:SWEep:STATe?	Page 63
:POLarimeter:SWEep:GET?	Page 64
:POLarimeter:FUNCtion:RESult?	Page 64
:POLarimeter:SWEep:GET:INDex?	Page 64
:POLarimeter:SWEep:SAMPles	Page 65
:POLarimeter:SWEep:SAMPles:CURRent?	Page 65
:POLarimeter:SWEep:SAMPles?	Page 65
:POLarimeter:SWEep:RATe	Page 66
:POLarimeter:SWEep:RATe?	Page 66
:POLarimeter:SWEep:SRATe	Page 66
:POLarimeter:SWEep:SRATe?	Page 66
:POLarimeter:SWEep:LPRange?	Page 67
:POLarimeter:SWEep:STEP?	Page 68
:POLarimeter:SWEep:STEP	Page 67
:POLarimeter:SWEep:TRIGger:PRE:SAMPles	Page 68
:POLarimeter:SWEep:TRIGger:PRE:SAMPles?	Page 68

Command	Page
:POLarimeter:SWEep:TRIGger:POST:SAMPles	Page 69
:POLarimeter:SWEep:TRIGger:POST:SAMPles?	Page 69
:POLarimeter:SWEep:CONTinuous:SACCuracy	Page 69
:POLarimeter:SWEep:CONTinuous:SACCuracy?	Page 70
:POLarimeter:TRIGger:INPut	Page 70
:POLarimeter:TRIGger:DELay?	Page 71
:POLarimeter:TRIGger:DELay	Page 70
:POLarimeter:TRIGger:OFFSet?	Page 71
:POLarimeter:TRIGger:OFFSet	Page 71
:POLarimeter:TRIGger:INPut?	Page 72
:POLarimeter:TRIGger:OUTPut	Page 72
:POLarimeter:TRIGger:OUTPut?	Page 72

Command	Page
Polarization Synchronous Scrambler Commands	
:PCONtroller:STOP	Page 73
:PCONtroller:WPLAtes	Page 73
:PCONtroller:WPLAtes?	Page 73
:PCONtroller:SEQuence:LENGth	Page 74
:PCONtroller:SEQuence:LENGth?	Page 74
:PCONtroller:SEQuence:RRATe	Page 74
:PCONtroller:SEQuence:RRATe?	Page 75
:PCONtroller:SEQuence:SMODe	Page 75
:PCONtroller:SEQuence:SMODe?	Page 76
:PCONtroller:SEQuence:DCOMpensation	Page 76
:PCONtroller:SEQuence:DCOMpensation?	Page 76

Command	Page
:PCONtroller:SEQuence:HOLDoff	Page 77
:PCONtroller:SEQuence:HOLDoff?	Page 77
:PCONtroller:SEQuence	Page 77
:PCONtroller:SEQuence?	Page 78
:PCONtroller:SEQuence:SEQVoltage	Page 78
:PCONtroller:SEQuence:SEQVoltage?	Page 79
:PCONtroller:SCRambler:ENABle	Page 79
:PCONtroller:SCRambler:ENABle?	Page 79
:PCONtroller:STAGe<stage>:DEGree	Page 80
:PCONtroller:STAGe<stage>:DEGree?	Page 80
:PCONtroller:STARt	Page 80
:PCONtroller:REPetition	Page 81
:PCONtroller:LOAD:DATA?	Page 81
:PCONtroller:SAVE:ARB?	Page 82
:PCONtroller:LOAD:ARB?	Page 82
:PCONtroller:GEN:RANDom?	Page 82
:PCONtroller:GEN:SCRAmble?	Page 83
:PCONtroller:SWITch	Page 83
:PCONtroller:SWITch?	Page 83

Command	Page
Polarization Synthesizer / Polarization Component Analyzer Commands	
:STABilizer:STABilize	Page 85
:STABilizer:STABilize?	Page 85
:STABilizer:SOP	Page 86
:STABilizer:SOP?	Page 86

Command	Page
SYSTem Commands	
:SYSTem:DATE	Page 34
:SYSTem:DATE?	Page 34
:SYSTem:HELP:HEADers?	Page 34
:SYSTem:HELP:ERRors?	Page 35
:SYSTem:TIME	Page 35
:SYSTem:PRESet	Page 36
:SYSTem:TIME?	Page 36
:SYSTem:ERRor[:NEXT]?	Page 36
:SYSTem:ERRor:COUNt?	Page 36
:SYSTem:VERSion?	Page 36
:SYSTem:REBoot	Page 37
:SYSTem:COMMunicate:ETHernet:AUTOip:ENABle?	Page 38
:SYSTem:COMMunicate:ETHernet:AUTOip:ENABle	Page 39
:SYSTem:COMMunicate:ETHernet:CANCel	Page 39
:SYSTem:COMMunicate:ETHernet:DGATeway	Page 39
:SYSTem:COMMunicate:ETHernet:DGATeway?	Page 39
:SYSTem:COMMunicate:ETHernet:DGATeway:CURRent?	Page 40
:SYSTem:COMMunicate:ETHernet:DHCP:ENABle?	Page 40
:SYSTem:COMMunicate:ETHernet:DHCP:ENABle	Page 40
:SYSTem:COMMunicate:ETHernet:DOMainname?	Page 41
:SYSTem:COMMunicate:ETHernet:DOMainname	Page 41
:SYSTem:COMMunicate:ETHernet:DOMainname:CURRent?	Page 41
:SYSTem:COMMunicate:ETHernet:HOSTname	Page 41
:SYSTem:COMMunicate:ETHernet:HOSTname?	Page 42
:SYSTem:COMMunicate:ETHernet:HOSTname:CURRent?	Page 42

Command	Page
:SYSTem:COMMunicate:ETHernet:NSERver?	Page 42
:SYSTem:COMMunicate:ETHernet:NSERver	Page 43
:SYSTem:COMMunicate:ETHernet:NSERver:CURRent?	Page 43
:SYSTem:COMMunicate:ETHernet:IDN	Page 43
:SYSTem:COMMunicate:ETHernet:IPADdress	Page 43
:SYSTem:COMMunicate:ETHernet:IPADdress?	Page 44
:SYSTem:COMMunicate:ETHernet:IPADdress:CURRent?	Page 44
:SYSTem:COMMunicate:ETHernet:MACaddress?	Page 44
:SYSTem:COMMunicate:ETHernet:NTP:ENABle?	Page 45
:SYSTem:COMMunicate:ETHernet:NTP:ENABle	Page 45
:SYSTem:COMMunicate:ETHernet:NTP:SERVer?	Page 45
:SYSTem:COMMunicate:ETHernet:NTP:SERVer	Page 45
:SYSTem:COMMunicate:ETHernet:DESCription?	Page 46
:SYSTem:COMMunicate:ETHernet:DESCription	Page 46
:SYSTem:COMMunicate:ETHernet:WHITEList:ENABle?	Page 46
:SYSTem:COMMunicate:ETHernet:WHITEList:ENABle	Page 47
:SYSTem:COMMunicate:ETHernet:WHITEList#:IPADdress?	Page 47
:SYSTem:COMMunicate:ETHernet:WHITEList#:IPADdress	Page 47
:SYSTem:COMMunicate:ETHernet:WHITEList#:SMASk?	Page 47
:SYSTem:COMMunicate:ETHernet:WHITEList#:SMASk	Page 48
::SYSTem:COMMunicate:ETHernet:RESet	Page 48
:SYSTem:COMMunicate:ETHernet:RESTart	Page 48
:SYSTem:COMMunicate:ETHernet:SAVE	Page 49
:SYSTem:COMMunicate:ETHernet:SMASk?	Page 49
:SYSTem:COMMunicate:ETHernet:SMASk	Page 49
:SYSTem:COMMunicate:ETHernet:SMASk:CURRent?	Page 50

Command	Page
TRIGger Commands	
:TRIGger	Page 55
:TRIGger:CONFiguration	Page 55
:TRIGger:CONFiguration?	Page 56

3 Instrument Setup and Status

This chapter gives descriptions of commands that you can use when setting up your instrument. The commands are split into the following separate subsytems:

- IEEE specific commands that were introduced in Common Commands on page 15.
- STATus subsystem commands that relate to the status model.
- SYSTem subsystem commands that control the serial interface and internal data.

IEEE-Common Commands

Common Commands on page 15 gave a brief introduction to the IEEE-common commands which can be used with the instruments. This section gives fuller descriptions of each of these commands.

Command:	***CLS**
Syntax:	*CLS
Description:	The Clear Status (*CLS) command clears the status byte by emptying the error queue and clearing all the event registers (SESR) including the Data Questionable Event Register, the Standard Event Status Register, the Standard Operation Status Register and any other registers that are summarized in the status byte.
Parameters:	none
Response:	none
Example:	*CLS

Command:	***ESE**		
Syntax:	*ESE<wsp><value> 0 ≤ *value* ≤ 255		
Description:	The standard Event Status Enable command (*ESE) sets bits in the Standard Event Status Enable Mask (SESEM) that enable the corresponding bits in the standard event status register (SESR). The register is cleared: at power-on, by sending a value of zero. The register is not changed by the *CLS command.		
Parameters:	The bit value for the register (a *16-bit signed integer* value):		
	Bit	Mnemonic	Decimal Value
	7 (MSB)	Power On	128
	6	Not Used	64
	5	Command Error	32
	4	Execution Error	16
	3	Device Dependent Error	8
	2	Query Error	4

	1	Not Used	2
	0 (LSB)	Operation Complete	1
Response:	none		
Example:	*ESE 255		

Command:	***ESE?**
Syntax:	*ESE?
Description:	The standard Event Status Enable query *ESE? returns the contents of the Standard Event Status Enable Mask (see *ESE for information on this register).
Parameters:	none
Response:	The bit value for the register (a *16-bit signed integer* value).
Example:	*ESE? —> 255

Command:	***ESR?**		
Syntax:	*ESR?		
Description:	The standard Event Status Register query *ESR? returns the contents of the Standard Event Status Register. The register is cleared after being read.		
parameters	none		
response	The bit value for the register (a *16-bit signed integer* value):		
	Bit	Mnemonic	Decimal Value
	7 (MSB)	Power On	128
	6	Not used	64
	5	Command Error	32
	4	Execution Error	16
	3	Device Dependent Error	8
	2	Query Error	4

	1	Not used	2
	0 (LSB)	Operation Complete	1
Example:	*ESR? –> +255		

Command:	***IDN?**	
Syntax:	*IDN?	
Description:	The IDeNtification query *IDN? gets the instrument identification over the interface.	
Parameters:	none	
Response:	The identification, for example:	
	MMMMMMMM *mmmm* *ssssssss* *rrrrrrrrrr*	manufacturer, for example Keysight Technologies instrument model number (for example N7785C) serial number firmware revision level
Example:	*IDN? -> Keysight Technologies,N7785C,DE42100168,V1.000	

Command:	***OPC**
Syntax:	*OPC
Description:	Generates the OPC message in the standard event status register when all pending overlapped operations have been completed.
Parameters:	none
Response:	none
Example:	*OPC

Command:	***OPC?**
Syntax:	*OPC?
Description:	The OPeration Complete query *OPC? parses all program message units in the input queue, sets the operation complete bit in the Standard Event Status register, and places an ASCII ’1’ in the output queue, when the contents of the input queue have been processed. Taking advantage of this feature, and using *OPC? in a loop to query until the instrument returns 1, can lead to useful gains in program execution efficiency.
Parameters:	none
Response:	1 is returned if all modules are ready to execute a new operation. 0 is returned if any module is busy.
Example:	*OPC? –> 1

Command:	***OPT?**
Syntax:	*OPT?
Description:	The OPTions query *OPT? returns the options installed in your instrument.
Parameters:	none
Response:	Returns the part number of all installed options, separated by commas. Slots are listed starting with the lowest slot number, that is, slot 1.
Example:	*OPT? -> N7752A-002, N7752A-002, N7752A-001,N7752A-001,

Command:	***RST**
Syntax:	*RST
Description:	The ReSeT command *RST sets the mainframe and all modules to the reset setting (standard setting) stored internally. The instrument is placed in the idle state awaiting a command. The *RST command clears the error queue. The *RST command is equivalent to the *CLS command AND the syst:preset command. The following are not changed: Instrument interface address Service request enable register (SRE) Standard Event Status Enable Mask (SESEM) NOTE: When the saved configurations are needed and should not be erased the preset can be done with conf:meas:sett:pres command.

Parameters:	none
Response:	none
Example:	*RST

Command:	***STB?**		
Syntax:	*STB?		
Description:	The STatus Byte query *STB? returns the contents of the Status Byte register.		
Parameters:	none		
Response:	The bit value for the register (a *16-bit signed integer* value):		
	Bit	Mnemonic	Decimal Value
	7 (MSB)	Operation Status (OSB)	128
	6	Not used	0
	5	Event Status Bit (ESB)	32
	4	Message Available (MAV)	16
	3	Questionable Status (QSB)	8
	2	Error/Event Queue	4
	1	Not used	0
	0	Not used	0
Example:	*STB? –> 128		

Command:	***TST?**
Syntax:	*TST?
Description:	The self-TeST query *TST? makes the instrument perform a self-test and place the results of the test in the output queue. If the self-test fails, the results are also put in the error queue. We recommend that you read self-test results from the error queue. No further commands are allowed while the test is running. After the self-test the instrument is returned to the setting that was active at the time the self-test query was processed. The self-test does not require operator interaction beyond sending the *TST? query.

Parameters:	none
Response:	Selftest failed 1 A value of zero indicates no errors.
Example:	*TST? > 0

Command:	***WAI**
Syntax:	*WAI
Description:	The WAIt command prevents the instrument from executing any further commands until the current command has finished executing. Some module firmware includes commands that set a "StatNOPC" flag during execution to indicate that the module is busy. *WAI blocks the GPIB bus to all commands until every module hosted by the instrument is no longer busy. All pending operations, are completed during the wait period.
Parameters:	none
Response:	none
Example:	*WAI

Interface/Instrument Behaviour Settings – The SYSTem Subsystem

The SYSTem subsystem lets you control the instrument’s serial interface. You can also control some internal data (like date, time, and so on)

Command:	**:SYSTem:DATE**
Syntax:	:SYSTem:DATE<wsp><year>,<month>,<day>
Description:	Sets the instrument’s internal date.
Parameters:	The date in the format year, month, day.
Response:	none
Example:	:syst:date 2019, 10, 12

Command:	**:SYSTem:DATE?**
Syntax:	:SYSTem:DATE?
Description:	Returns the instrument’s internal date.
Parameters:	none
Response:	The date in the format year, month, day.
Example:	:syst:date? –> +2019,+10,+12

Command:	**:SYSTem:HELP:HEADers?**
Syntax:	:SYSTem:HELP:HEADers?
Description:	Returns a list of commands.
Parameters:	none
Response:	Returns a list of commands
Example:	:syst:help:head? -> Returns a list of all commands

Command:	**:SYSTem:HELP:ERRors?**
Syntax:	:SYSTem:HELP:ERRors?
Description:	Return an overview about all Errorcodes and a short description.
Parameters:	none
Response:	String list of error codes
Example:	:syst:help:err? -> +0,"No error",-100,"Command error",- 101,"Invalid character",-102,"Syntax error",-103, "Invalid separator",-104,"Data type error",-105,"GET not allowed",-108,"Parameter not allowed",...

Command:	**:SYSTem:PRESet**
Syntax:	:SYSTem:PRESet
Description:	Sets the instrument to the standard settings. This command has the same function as the Preset hardkey. Pressing the "LAN Reset" Button for a short time has the same effect. Long pressing of the "LAN Reset" Button resets the LAN parameter. The following are not affected by this command: the interface address, the output and error queues, the Service Request Enable register (SRE), the Status Byte (STB), the Standard Event Status Enable Mask (SESEM), and the Standard Event Status Register (SESR).
Parameters:	none
Response:	none
Example:	:SYST:PRES

Command:	**:SYSTem:TIME**
Syntax:	:SYSTem:TIME<wsp><hour>,<minute>,<second>
Description:	Sets the instrument’s internal time.
Parameters:	24-hour time format: hours (0-23), minutes (0-59), seconds (0-59).
Response:	none
Example:	:syst:time 20,15,30

Command:	**:SYSTem:TIME?**
Syntax:	:SYSTem:TIME?
Description:	Returns the instrument’s internal time.
Parameters:	none
Response:	The time in the format hour, minute, second. Hours are counted 0...23.
Example:	:syst:time? –> +20,+15,+30

Command:	**:SYSTem:ERRor[:NEXT]?**
Syntax:	:SYSTem:ERRor[:NEXT]?
Description:	Returns the next error from the error queue.
Parameters:	none
Response:	The number of the latest error, and its meaning.
Example:	:syst:err? -> -113,"Undefined header"

Command:	**:SYSTem:ERRor:COUNt?**
Syntax:	:SYSTem:ERRor:COUNt?
Description:	Returns the total no. of errors.
Parameters:	none
Response:	The total count of errors.
Example:	:syst:err:coun? -> 20

Command:	**:SYSTem:VERSion?**
Syntax:	:SYSTem:VERSion?
Description:	Returns the SCPI revision to which the instrument complies.

Parameters:	none
Response:	The revision year and number.
Example:	:syst:vers? —> 1999

Command:	**:SYSTem:REBoot**
Syntax:	:SYSTem:REBoot
Description:	Reboots the instrument.
Parameters:	none
Response:	None
Example:	:syst:reb

System Communicate - The :SYST:COMMunicate Subsystem

We recommend you change network settings using the local user interface.

NOTE

The instrument does not close open connections when restarting the network interface (:SYSTem:COMMunicate:ETHernet:RESTart). This means the number of possible connections is reduced by the number of previously open connections. However, the instrument does make sure connections are still alive. It should release unused open connections after about two minutes.

Some notes on DHCP/AutoIP/DNS

- If DHCP is enabled but no DHCP server is found, the instrument tries to use AutoIP as a fallback. This may take about 2 minutes.
- Depending on the available network capabilities, the instrument tries to tell the DNS server its host name or read the host and domain named it has been assigned.

MAC address:

The Media Access Control (MAC) number is a unique number associated with each network adapter.

Command:	**:SYSTem:COMMunicate:ETHernet:AUTOip:ENABle?**
Syntax:	:SYSTem:COMMunicate:ETHernet:AUTOip:ENABle?
Description:	Check whether Automatic IP addressing is enabled or disabled.
Parameters:	None
Response:	Boolean (0 | 1)
Example:	:SYST:COMM:ETH:AUTO:ENAB?

Command:	**:SYSTem:COMMunicate:ETHernet:AUTOip:ENABle**
Syntax:	:SYSTem:COMMunicate:ETHernet:AUTOip:ENABle
Description:	Enable or disable whether IP addresses can be created automatically by the instrument. Automatic IP addressing is only used if DHCP is enabled, but the instrument cannot find a DHCP server.
Parameters:	Boolean (0 | 1 | off | on)
Response:	None
Example:	:SYST:COMM:ETH:AUTO:ENAB 1

Command:	**:SYSTem:COMMunicate:ETHernet:CANCel**
Syntax:	:SYSTem:COMMunicate:ETHernet:CANCel
Description:	Undo all changes to the network parameters that have been made since the last save, reboot or ":syst:comm:eth:restart" command.
Parameters:	None
Response:	None
Example:	:SYST:COMM:ETH:CANC

Command:	**:SYSTem:COMMunicate:ETHernet:DGATeway**
Syntax:	:SYSTem:COMMunicate:ETHernet:DGATeway
Description:	Set the default gateway.
Parameters:	string (Up to four groups of up to 3 digits, groups separated by ".". Groups with leading zeros are interpreted as octal numbers.)
Response:	None
Example:	:syst:comm:eth:dgat “192.168.101.11“

Command:	**:SYSTem:COMMunicate:ETHernet:DGATeway?**
Syntax:	:SYSTem:COMMunicate:ETHernet:DGATeway?
Description:	Get the default gateway.

Parameters:	None
Response:	String
Example:	:syst:comm:eth:dgat? -> “192.168.101.11“

Command:	**:SYSTem:COMMunicate:ETHernet:DGATeway:CURRent?**
Syntax:	:SYSTem:COMMunicate:ETHernet:DGATeway:CURRent?
Description:	Get the currently used default gateway.
Parameters:	None
Response:	String
Example:	:syst:comm:eth:dgat:curr? -> “192.168.101.11“

Command:	**:SYSTem:COMMunicate:ETHernet:DHCP:ENABle?**
Syntax:	:SYSTem:COMMunicate:ETHernet:DHCP:ENABle?
Description:	Check whether DHCP is enabled or disabled.
Parameters:	None
Response:	Boolean (0 | 1)
Example:	:syst:comm:eth:dhcp:enab? -> 1

Command:	**:SYSTem:COMMunicate:ETHernet:DHCP:ENABle**
Syntax:	:SYSTem:COMMunicate:ETHernet:DHCP:ENABle
Description:	Enable or disable DHCP
Parameters:	Boolean (0 | 1 | off | on)
Response:	None
Example:	:syst:comm:eth:dhcp:enab on

Command:	**:SYSTem:COMMunicate:ETHernet:DOMainname?**
Syntax:	:SYSTem:COMMunicate:ETHernet:DOMainname?
Description:	Get the domain name.
Parameters:	None
Response:	String
Example:	:syst:comm:eth:dom? -> “.companyname.com”

Command:	**:SYSTem:COMMunicate:ETHernet:DOMainname**
Syntax:	:SYSTem:COMMunicate:ETHernet:DOMainname
Description:	Set the domain name (used if DHCP is disabled).
Parameters:	String
Response:	None
Example:	:syst:comm:eth:dom “.companyname.com”

Command:	**:SYSTem:COMMunicate:ETHernet:DOMainname:CURRent?**
Syntax:	:SYSTem:COMMunicate:ETHernet:DOMainname:CURRent?
Description:	Get the currently used domain name.
Parameters:	None
Response:	String
Example:	:syst:comm:eth:dom:curr? -> “.companyame.com”

Command:	**:SYSTem:COMMunicate:ETHernet:HOSTname**
Syntax:	:SYSTem:COMMunicate:ETHernet:HOSTname
Description:	Set the host name.

Parameters:	string (maximum 19 characters, though not all characters can be used) The default host name is K-P...P-S...S; where P...P is the product Number, and S...S is as many of the last digits of the serial number as it takes to get a 15 character host name. If you set an empty host name (""), the host name will be set to its default value.
Response:	none
Example:	:syst:comm:eth:host “N7745C”

Command:	**:SYSTem:COMMunicate:ETHernet:HOSTname?**
Syntax:	:SYSTem:COMMunicate:ETHernet:HOSTname?
Description:	Get the host name.
Parameters:	None
Response:	String
Example:	:syst:comm:eth:host? -> “K-N7745C-0PP03”<END

Command:	**:SYSTem:COMMunicate:ETHernet:HOSTname:CURRent?**
Syntax:	:SYSTem:COMMunicate:ETHernet:HOSTname:CURRent?
Description:	Get the current host name.
Parameters:	None
Response:	String
Example:	:syst:comm:eth:host:curr? -> "K-N778-C-12345"

Command:	**:SYSTem:COMMunicate:ETHernet:NSERver?**
Syntax:	:SYSTem:COMMunicate:ETHernet:NSERver?
Description:	Get the defined (DNS) nameserver for name resolution.
Parameters:	None
Response:	IP Address String
Example:	:syst:comm:eth:nser? -> "1.1.1.1", "2.2.2.2"

Command:	**:SYSTem:COMMunicate:ETHernet:NSERver**
Syntax:	:SYSTem:COMMunicate:ETHernet:NSERver
Description:	Set one or two nameservers for name resolution. (used if DHCP is disabled).
Parameters:	IP Address String
Response:	None
Example:	:syst:comm:eth:nser "1.1.1.1"

Command:	**:SYSTem:COMMunicate:ETHernet:NSERver:CURRent?**
Syntax:	:SYSTem:COMMunicate:ETHernet:NSERver:CURRent?
Description:	Get the DNS server addresses assigned from your DHCP sever (this is only valide if DHCP is available and enabled.
Parameters:	None
Response:	IP Address String
Example:	:syst:comm:eth:nser:curr? -> "10.127.72.11","10.127.90.11"

Command:	**:SYSTem:COMMunicate:ETHernet:IDN**
Syntax:	:SYSTem:COMMunicate:ETHernet:IDN
Description:	The LAN LED on the front panel of the instrument flashes for identification.
Parameters:	Boolean (0 | 1 | off | on)
Response:	None
Example:	:syst:comm:eth:idn 1

Command:	**:SYSTem:COMMunicate:ETHernet:IPADdress**
Syntax:	:SYSTem:COMMunicate:ETHernet:IPADdress
Description:	Set the IP address of the system manually (used if DHCP is disabled).

Parameters:	String (Up to four groups of up to 3 digits, groups separated by ".". Groups with leading zeroes are interpreted as octal numbers.)
Response:	None
Example:	:syst:comm:eth:ipad “192.132.13.2”

Command:	**:SYSTem:COMMunicate:ETHernet:IPADdress?**
Syntax:	:SYSTem:COMMunicate:ETHernet:IPADdress?
Description:	Get the manually set IP address of the system.
Parameters:	None
Response:	String
Example:	:syst:comm:eth:ipad? –> “192.132.13.2”

Command:	**:SYSTem:COMMunicate:ETHernet:IPADdress:CURRent?**
Syntax:	:SYSTem:COMMunicate:ETHernet:IPADdress:CURRent?
Description:	Get the current IP address of the instrument.
Parameters:	None
Response:	String
Example:	:syst:comm:eth:ipad:curr? -> “192.132.13.2”

Command:	**:SYSTem:COMMunicate:ETHernet:MACaddress?**
Syntax:	:SYSTem:COMMunicate:ETHernet:MACaddress?
Description:	Get the MAC address of the network adapter.
Parameters:	None
Response:	String (hexadecimal value).
Example:	:syst:comm:eth:mac? -> "00-07-E0-14-AE- 08"

Command:	**:SYSTem:COMMunicate:ETHernet:NTP:ENABle?**
Syntax:	:SYSTem:COMMunicate:ETHernet:NTP:ENABle?
Description:	Returns the usage of a NTP Server
Parameters:	None
Response:	Boolean (0 | 1)
Example:	:syst:comm:eth:ntp:enab? -> 1

Command:	**:SYSTem:COMMunicate:ETHernet:NTP:ENABle**
Syntax:	:SYSTem:COMMunicate:ETHernet:NTP:ENABle
Description:	Disables or enables instrument's use of NTP. The acronym NTP stands for Network Time Protocol, a protocol for clock synchronization between computer systems.
Parameters:	Boolean (0 | 1)
Response:	None
Example:	:syst:comm:eth:ntp:enab 1

Command:	**:SYSTem:COMMunicate:ETHernet:NTP:SERVer?**
Syntax:	:SYSTem:COMMunicate:ETHernet:NTP:SERVer?
Description:	Get the defined Network Time Protocol (NTP) server for clock synchronization.
Parameters:	None
Response:	Address String
Example:	:syst:comm:eth:ntp:serv? -> "pool.ntp.org"

Command:	**:SYSTem:COMMunicate:ETHernet:NTP:SERVer**
Syntax:	:SYSTem:COMMunicate:ETHernet:NTP:SERVer
Description:	Get the defined Network Time Protocol (NTP) server for clock synchronization.

Parameters:	Address String
Response:	None
Example:	:syst:comm:eth:ntp:serv "pool.ntp.org"

Command:	**:SYSTem:COMMunicate:ETHernet:DESCription?**
Syntax:	:SYSTem:COMMunicate:ETHernet:DESCription?
Description:	Get the desired mDNS service name.
Parameters:	None
Response:	Quoted string of up to 260 characters
Example:	:syst:comm:eth:desc? -> "Keysight N778-C - 42321"

Command:	**:SYSTem:COMMunicate:ETHernet:DESCription**
Syntax:	:SYSTem:COMMunicate:ETHernet:DESCription
Description:	Set the desired mDNS service name.
Parameters:	Quoted string of up to 260 characters
Response:	None
Example:	:syst:comm:eth:desc "Keysight N778-C - 42321"

Command:	**:SYSTem:COMMunicate:ETHernet:WHITEList:ENABle?**
Syntax:	:SYSTem:COMMunicate:ETHernet:WHITEList:ENABle?
Description:	Get the deive access IP whitelist status
Parameters:	None
Response:	Boolean (0 | 1)
Example:	:syst:comm:eth:whitel:enab? +0

Command:	**:SYSTem:COMMunicate:ETHernet:WHITEList:ENABle**
Syntax:	:SYSTem:COMMunicate:ETHernet:WHITEList:ENABle
Description:	Set or enables the device access IP whitelist
Parameters:	Boolean (0 | 1)
Response:	None
Example:	:syst:comm:eth:whitel:enab 0

Command:	**:SYSTem:COMMunicate:ETHernet:WHITEList#:IPADdress?**
Syntax:	:SYSTem:COMMunicate:ETHernet:WHITEList#:IPADdress?
Description:	Get one entry from the device access IP whitelist (up to 10 IP Addresses) of permitted client which could control this instrument.
Parameters:	None
Response:	IP Address String
Example:	:syst:comm:eth:whitel1:ipad? " -> 192.168.0.2"

Command:	**:SYSTem:COMMunicate:ETHernet:WHITEList#:IPADdress**
Syntax:	:SYSTem:COMMunicate:ETHernet:WHITEList#:IPADdress
Description:	Define an entry in the device access IP whitelist (up to 10 IP Addresses) of permitted client which could control this instrument.
Parameters:	IP Address String
Response:	None
Example:	:syst:comm:eth:whitel1:ipad "192.168.0.2"

Command:	**:SYSTem:COMMunicate:ETHernet:WHITEList#:SMASk?**
Syntax:	:SYSTem:COMMunicate:ETHernet:WHITEList#:SMASk?
Description:	Get an entry in the device access ip whitelist (up to 10 IP Addresses) of permitted client which could control this instrument.

Parameters:	Int (1-10)
Response:	IP Address String
Example:	:syst:comm:eth:whitel1:smas? -> "192.168.0.2"

Command:	**:SYSTem:COMMunicate:ETHernet:WHITEList#:SMASk**
Syntax:	:SYSTem:COMMunicate:ETHernet:WHITEList#:SMASk
Description:	Define an entry in the device access ip whitelist (up to 10 IP Addresses) of permitted client which could control this instrument.
Parameters:	Subnet Mask String
Response:	None
Example:	:syst:comm:eth:whitel1:smas "192.168.0.2"

Command:	**::SYSTem:COMMunicate:ETHernet:RESet**
Syntax:	:SYSTem:COMMunicate:ETHernet:RESet
Description:	Press the "LAN Reset" Button for a long time has the same effect. Pressing the "LAN Reset" Button for a short time is the same as system:preset. DHCP On AutoIP On NTP Off Whitelist Off Hostname is a concatenation of product number and serial number. The password for the web based LAN configuration interface is reset to 'blank' (no password).
Parameters:	None
Response:	None
Example:	:syst:comm:eth:res

Command:	**:SYSTem:COMMunicate:ETHernet:RESTart**
Syntax:	:SYSTem:COMMunicate:ETHernet:RESTart
Description:	Restart the system’s network interface with the new parameters. This command only works if the instrument has a working network connection at the time the command is issued. If not you either have to wait until the instrument decides on an IP address using AutoIP or reboot the instrument.

Parameters:	None
Response:	String
Example:	:syst:comm:eth:rest

Command:	**:SYSTem:COMMunicate:ETHernet:SAVE**
Syntax:	:SYSTem:COMMunicate:ETHernet:SAVE
Description:	Save the system's network interface parameters.
Parameters:	None
Response:	None
Example:	:syst:comm:eth:save

Command:	**:SYSTem:COMMunicate:ETHernet:SMASk?**
Syntax:	:SYSTem:COMMunicate:ETHernet:SMASk?
Description:	Get the subnet mask.
Parameters:	None
Response:	String
Example:	:syst:comm:eth:smas? -> "255.255.255.0"

Command:	**:SYSTem:COMMunicate:ETHernet:SMASk**
Syntax:	:SYSTem:COMMunicate:ETHernet:SMASk
Description:	Set the subnet mask.
Parameters:	String (Up to four groups of up to 3 digits, groups separated by ".". Groups with leading zeroes are interpreted as octal numbers.)
Response:	None
Example:	:syst:comm:eth:smas "255.255.255.0">

Command:	**:SYSTem:COMMunicate:ETHernet:SMASk:CURRent?**
Syntax:	:SYSTem:COMMunicate:ETHernet:SMASk:CURRent?
Description:	Get the currently used subnet mask.
Parameters:	None
Response:	String
Example:	:syst:comm:eth:smas:curr? -> “255.255.255.0”

Keysight N778-C Series Polarization Instruments

Programming Guide

4 Measurement Operations & Settings

This chapter gives descriptions of commands that you can use when you are setting up or performing measurements. The commands are split up into the following subsystems:

Configure subsystem commands that control all instruments.

TRIGger subsystem commands that control triggering.

Configure Subsystem Commands

This section provides the description of the following commands.

Command:	**:CONFigure:MEASurement:SETTing:ACTual?**
Syntax:	:CONFigure:MEASurement:SETTing:ACTual?
Description:	Get the index of the setting currently being used.
Parameters:	None
Response:	int A value >0 is returned if the setting has been stored in FLASH memory (using :CONFigure:MEASurement:SETTing:SAVE), or has been recalled from FLASH memory (using :CONFigure:MEASurement:SETTing:RECall), and has not been changed since. 0 is returned if the setting has not yet been stored. 0 is returned if the FLASH setting has been deleted (using:CONFigure:MEASurement:SETTing:ERASe) since the last recall or store. -1 is returned if the setting was changed but has not been saved yet.
Example:	:conf:meas:sett:act? → +2
Affects:	All instruments

Command:	**:CONFigure:MEASurement:SETTing:NUMBer?**
Syntax:	:CONFigure:MEASurement:SETTing:NUMBer?
Description:	Get the number of settings. In addition to the settings spaces in FLASH memory, the working memory can hold a setting.
Parameters:	None
Response:	int
Example:	:conf:meas:sett:numb? → +1
Affects:	All instruments

Command:	**:CONFigure:MEASurement:SETTing:PRESet**
Syntax:	:CONFigure:MEASurement:SETTing:PRESet
Description:	Resets the setting values in the working memory. In contrast to the *RST and System:Preset commands, the previous stored settings remain in nonvolatile RAM and can be recalled again.
Parameters:	None

Response:	None
Example:	:conf:meas:sett:pres
Affects:	All instruments

Command:	**:CONFigure:MEASurement:SETTing:CANCel**
Syntax:	:CONFigure:MEASurement:SETTing:CANCel
Description:	Discard all the changes to the setting since the last save or recall
Parameters:	None
Response:	None
Example:	:conf:meas:sett:canc
Affects:	All instruments

Command:	**:CONFigure:MEASurement:SETTing:RECall**
Syntax:	:CONFigure:MEASurement:SETTing:RECall
Description:	Recall a setting from FLASH memory
Parameters:	Integer
Response:	None
Example:	:conf:meas:sett:rec 1
Affects:	All instruments

Command:	**:CONFigure:MEASurement:SETTing:SAVE**
Syntax:	:CONFigure:MEASurement:SETTing:SAVE
Description:	Recall a setting from FLASH memory
Parameters:	Integer
Response:	None
Example:	:conf:meas:sett:save 1
Affects:	All instruments

Command:	**:CONFigure:MEASurement:SETTing:ERASe**
Syntax:	:CONFigure:MEASurement:SETTing:ERASe
Description:	Erase a setting from memory
Parameters:	Integer
Response:	None
Example:	:conf:meas:sett:eras 1
Affects:	All instruments

Triggering – The TRIGger Subsystem

Command:	**:TRIGger**	
Syntax:	:TRIGger<wsp>NODEA|1|NODEB|2	
Description:	Generates a hardware trigger.	
Parameters:	1 or NODEA: 2 or NODEB:	Is identical to a trigger at the Input Trigger Connector. Generates trigger at the Output Trigger Connector.
	A hardware trigger cannot be effective in the DISabled triggering mode but can be effective in DEFault, PASSthrough or LOOPback triggering modes, see :TRIGger:CONFiguration on page 55 for information on triggering modes.	
Response:	None	
Example:	:trig 1	

Command:	**:TRIGger:CONFiguration**	
Syntax:	:TRIGger:CONFiguration	
Description:	Sets the hardware trigger configuration with regard to Output and Input Trigger Connectors.	
Parameters:	0 or DISabled: 1 or DEFault: 2 or PASSthrough: 3 or LOOPback: 4 or SCRambler2polarimeter: 5 or POLarimeter2scrambler:	Trigger connectors are disabled. The Input Trigger Connector is activated, the incoming trigger response for each slot. A trigger at the Input Trigger Connector generates a trigger at the Output Trigger Connector automatically. No triggers from the instrument at the output. The same as PASSthrough. This is included for compatibility reasons. The internal polarization scrambler forward the trigger-out signal to the built-in polarimeter. This signal can be delayed by configure the :PCONtroller:SEQuence:HOLDoff command). The internal polarimeter forward the trigger-out signal to the built-in scrambler.
Response:	none	
Example:	:trig:conf dis	
Affects	All modules (SCR and POL are only available at the N7786C and N7788C which have a build-in polarization scrambler and polarimeter).	

Command:	**:TRIGger:CONFiguration?**	
Syntax:	:TRIGger:CONFiguration?	
Description:	Returns the hardware trigger configuration.	
Parameters:	None	
Parameters:	0 or DISabled:	Trigger connectors are disabled.
	1 or DEFault:	The Input Trigger Connector is activated, the incoming trigger response for each slot.
	2 or PASSthrough:	A trigger at the Input Trigger Connector generates a trigger at the Output Trigger Connector automatically. No triggers from the instrument at the output.
	3 or LOOPback:	The same as PASSthrough. This is included for compatibility reasons.
	4 or SCRambler2polarimeter:	The internal polarization scrambler forward the trigger-out signal to the built-in polarimeter. This signal can be delayed by configure the :PCONtroller:SEQuence:HOLDoff command).
	5 or POLarimeter2scrambler:	The internal polarimeter forward the trigger-out signal to the built-in scrambler.
Example:	:trig:conf? -> DIS	
Affects	All modules (SCR and POL are only available at the N7786C and N7788C which have a build-in polarization scrambler and polarimeter).	

5 Polarization Commands

This chapter provides description of the polarization commands.

Polarization Analyzer Commands

Command:	**:POLarimeter:SOP?**
Syntax:	:POLarimeter:SOP?
Description:	Returns the measured S0, S1, S2 and S3 stokes parameter The normalized Stokes parameter are obtained by dividing the Stokes parameters by the total optical power s1 = S1/S0 s2=S2/S0 s3=S3/S0
Parameters:	None
Response:	Comma-separated values
Example:	:POLarimeter:SOP? -> +1.53819801E-05,+1.15407602E-05,+4.74578428E06,+5.56273017E-06
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SOP:FETCh?**
Syntax:	:POLarimeter:SOP:FETCh?
Description:	Returns the S0, S1, S2 and S3 stokes parameter from last measurement. The normalized Stokes parameter are obtained by dividing the Stokes parameters by the total optical power s1 = S1/S0 s2=S2/S0 s3=S3/S0
Parameters:	None
Response:	Comma-separated values
Example:	:POLarimeter:SOP:FETCH? -> +1.53819801E-05,+1.15407602E-05,+4.74578428E06,+5.56273017E-06
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:POWer?**
Syntax:	:POLarimeter:POWer?
Description:	Returns the measured optical power in configured power unit.
Parameters:	None
Response:	Float
Example:	:POL:POW? -> +1.53759156E-05
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:POWer:FETCh?**
Syntax:	:POLarimeter:POWer:FETCh?
Description:	Returns the power from last measurement
Parameters:	None
Response:	Float
Example:	:POL:POW:FETC? -> +1.53759156E-05
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:POWer:UNIT**
Syntax:	:POLarimeter:POWer:UNIT<wsp>DBM|0|Watt|1
Description:	Sets the power unit
Parameters:	0 or DBM: dBm 1 or Watt: Watts
Example:	:POL:POW:UNIT 1
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:POWer:UNIT?**
Syntax:	:POLarimeter:POWer:UNIT?
Description:	Returns the power unit
Parameters:	None
Response:	0: dBm 1: Watt
Example:	:POL:POW:UNIT? -> 1
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:WAVelength**
Syntax:	:POLarimeter:WAVelength<wsp><wavelength> [PM|NM|UM|MM|M]|MINimum|MAXimum|DEFault
Description:	This command sets the current wavelength / start wavelength.
Parameters:	The wavelength as a float value. If you specify no units in your command, meters are used as the default.
Example:	:POLarimeter:WAVelength 1.550E-06
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:WAVelength?**	
Syntax:	:POLarimeter:WAVelength? [MINimum|MAXimum|DEFault]	
Description:	Returns wavlength	
Parameters:	Optional	MINimum: Returns the minimum wavelength MAXimum: Returns the maximum wavelength DEFault: Returns default wavelength
Response	The wavelength as a float value in meters.	
Example:	:pol:wav? 1.550E-06	
Affects	N7781C / N7786C / N7788C	

Command:	**:POLarimeter:GAIN**
Syntax:	:POLarimeter:GAIN<wsp><gain level>
Description:	0...5 = High Bandwith about 250kHz 6,7 = Medium Bandwith about 100kHz 8,9 = Slow Bandwith about 10kHz NOTE - Use only gain between 0 and 7 for Stabilizer Mode because 8 and 9 have a small Bandwith. For best results check the Leveling of the Polarimeter. Look at the POL:SWE:LPR? command. Maximum Speed is at Gain 0 to 5 Even Gains have Factors of 10 and odd Gains Factors of 8. Choose what fits better.
Parameters:	Gain level as integer
Response	Gain level as integer
Example:	:pol:gain 5
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:GAIN?**
Syntax:	:POLarimeter:GAIN?
Description:	Returns the gain level
Parameters:	None
Response	Gain level as integer
Example:	:pol:gain? -> +8
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:AGFLag**
Syntax:	:POLarimeter:AGFLag
Description:	Enable or disable auto gain NOTE - Auto Gain is disabled when setting manual Gain with the POL:GAIN command. NOTE - Auto gain setting is ignored in Logging / Sweep and Stabilizer Mode.
Parameters:	0: auto gain off 1: auto gain on
Example:	:pol:agfl 1
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:AGFLag?**
Syntax:	:POLarimeter:AGFLag?
Description:	Returns the auto gain flag
Parameters:	A boolean value: 0 –auto gain is switched off 1 – auto gain is switched on
Example:	:pol:agfl? -> 1
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:STOP**
Syntax:	:POLarimeter:STOP
Description:	Stops logging
Parameters:	None
Example:	:pol:swe:stop
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:ZERO**
Syntax:	:POLarimeter:ZERO
Description:	Zeros photodiodes. That means dark current is measured and will be subtracted from future measurements.
Parameters:	None
Example:	:pol:zero
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:ZERO?**
Syntax:	:POLarimeter:ZERO?
Description:	Returns zero result
Parameters:	None
Response	0: last zero was successfull 1: last zero failed (too much light, for example)
Example:	:pol:zero?
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:LOOP**
Syntax:	:POLarimeter:SWEep:LOOP
Description:	Sets the number of loops, which should be logged after starting

Parameters:	Integer 0: endless
Example:	:pol:swe:loop 1
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:LOOP?**
Syntax:	:POLarimeter:SWEep:LOOP?
Description:	Gets the number of loops, which should be logged after starting
Parameters:	None
Response:	Integer
Example:	:pol:swe:loop?
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:STARt**
Syntax:	:POLarimeter:SWEep:STARt<wsp>[SOP|SOPCONTINUOUS]
Description:	This command starts the polarimeter logging.
Parameters:	None: Starts logging as configured SOP: Starts logging and sets loop to 1 SOPCONTINUOUS: Starts logging and sets loop to 0 (endless logging)
Example:	:pol:swe:star sop
Affects	N7781C / N7786C / N7788C

Command:	**POLarimeter:SWEep:STATe?**
Syntax:	POLarimeter:SWEep:STATe?
Description:	This command returns the current logging state Return value status string e.g SAMPLING,NO_DATA
Parameters:	None

Response:	IDLE,NO_DATA: No data available, logging stopped SAMPLING,DATA_AVAILABLE: Data available, logging running SAMPLING,NO_DATA: No data available, logging running READY,DATA_AVAILABLE: Data available, logging not running IDLE,DATA_AVAILABLE: Data available, logging stopped
Example:	:pol:swe:stat? -> SAMPLING,DATA_AVAILABLE
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:GET?**
Syntax:	:POLarimeter:SWEep:GET?<wsp>[SOP|NORMalized]
Description:	This command returns the measurement result as float binblock.
Parameters:	None: returns 4,* dimensional array with S_0, S_1, S_2, S_3 SOP: returns 4,* dimensional array with S_0, S_1, S_2, S_3 NORMalized: returns 3,* dimensional array with S_1, S_2, S_3
Response:	float binblock
Example:	pol:swe:get? sop
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:FUNCtion:RESult?**
Syntax:	:POLarimeter:FUNCtion:RESult?
Description:	Returns a binblock with measured power
Parameters:	None
Response:	binblock float
Example:	:pol:func:res? → binblock
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:GET:INDex?**
Syntax:	:POLarimeter:SWEep:GET:INDex?
Description:	Returns the number of already finished logging loops

Parameters:	None
Response:	Integer
Example:	:pol:swe:get:ind? → +1120055703
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:SAMPles**
Syntax:	:POLarimeter:SWEep:SAMPles
Description:	This command sets the number of samples/logging count (1-1048576)
Parameters:	Number of samples
Example:	:pol:swe:samp 1000
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:SAMPles?**
Syntax:	:POLarimeter:SWEep:SAMPles?
Description:	This command gets the number of samples/logging count
Parameters:	None
Response:	Interger
Example:	:pol:swe:samp? → +1000
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:SAMPles:CURRent?**
Syntax:	:POLarimeter:SWEep:SAMPles:CURRent?
Description:	This command gets the number of already logged values. Will be 0, if logging finished.
Parameters:	None
Response:	Number of already logged samples.
Example:	:pol:swe:samp:curr? → +34
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:RATe**
Syntax:	:POLarimeter:SWEep:RATe<wsp>[nm/s|pm/s]
Description:	This command sets the sweep rate in m/s. When performing a swept measurement, very often, the wavelength is changed over time. This parameter allows you to inform the instrument about the speed at which the wavelength of the laser source is changing. The start wavelength is given by the property Wavelength. This works for disabled trigger input only. If you want to sweep with trigger input, please set sweep step: :POLarimeter:SWEep:STEP.
Parameters:	Float sweep rate in nm/s. If you do not pass unit, assumption is m/s.
Example:	:pol:swe:rat 40nm/s
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:RATe?**
Syntax:	:POLarimeter:SWEep:RATe?
Description:	This command returns the sweep rate in m/s.
Parameters:	None
Response:	Float sweep rate
Example:	:pol:swe:rat? → +4.00000000E-08
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:SRATe**
Syntax:	:POLarimeter:SWEep:SRATe<wsp><value>[Hz|kHz|MHz]|MIN|MAX|DEF[,<value>S|MS|US|MIN|MAX|DEF]
Description:	This command sets the sampling rate. You can pass just the sampling rate (or MIN, MAX, DEF) or sampling rate and averaging time (or MIN, MAX, DEF). Sampling periode (1/sampling rate) have to be larger or equal averaging time.
Parameters:	Sampling rate in Hz or MIN, MAX, DEF Averaging time in s or MIN, MAX, DEF(optional)
Example:	:pol:swe:srat 0.5MHz,1us
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:SRATe?**	
Syntax:	:POLarimeter:SWEep:SRATe?[<wsp>MIN|MAX|DEF]	
Description:	This command gets the sampling rate and averaging time	
Parameters:	None: to get current sampling rate and averaging time MIN: minimum sampling rate MAX: maximum sampling rate DEF: default sampling rate	
Response:	No parameter:	sampling rate in float averaging time in float
	MIN|MAX|DEF parameter:	sampling rate in float
Example:	:pol:swe:srat? → +1e+06,+1e-06	
Affects	N7781C / N7786C / N7788C	

Command:	**:POLarimeter:SWEep:LPRange?**
Syntax:	:POLarimeter:SWEep:LPRange?
Description:	Returns how well the ADC range was used for the last peak power. If this value is below 0.5, the measurement sweep should be repeated with a higher amplifier gain setting. If the Value is > 1.0 then there is an increasing Risk of Overflow. Select a lower amplifier setting in this case. NOTE: use only gain between 0 and 7 for Stabilizer Mode because 8 and 9 have a small Bandwith
Parameters:	None
Response:	Float
Example:	:pol:swe:lpr? → +0.50000000E+00
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:STEP**
Syntax:	:POLarimeter:SWEep:STEP<wsp><value>[m|nm|pm]
Description:	Sets the sweep step in m

Parameters:	Sweep step in m
Example:	:pol:swe:step 1pm
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:STEP?**
Syntax:	:POLarimeter:SWEep:STEP?
Description:	In SME mode, this is the step (wavelength increased by this value) which is done for every trigger while logging.
Parameters:	None
Response:	Sweep step in float
Example:	:pol:swe:step? → +1.00000000E-12
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:TRIGger:PRE:SAMPles**
Syntax:	:POLarimeter:SWEep:TRIGger:PRE:SAMPles
Description:	Sets the number of pre samples. That means samples before trigger event. Numbers of pre samples and post samples have to be smaller than 1048576.
Parameters:	Number of pre samples
Example:	:pol:swe:trig:pre:samp 1000
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:TRIGger:PRE:SAMPles?**
Syntax:	:POLarimeter:SWEep:TRIGger:PRE:SAMPles?
Description:	Returns the number of pre samples.
Parameters:	None
Response:	Number of pre samples
Example:	:pol:swe:trig:pre:samp? → +0
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:TRIGger:POST:SAMPles**
Syntax:	:POLarimeter:SWEep:TRIGger:POST:SAMPles
Description:	Sets the number of post samples. That means samples after trigger event. Numbers of pre samples and post samples have to be smaller than 1048576.
Parameters:	Number of post samples
Example:	:pol:swe:trig:post:samp 2000
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:TRIGger:POST:SAMPles?**
Syntax:	:POLarimeter:SWEep:TRIGger:POST:SAMPles?
Description:	Returns the number of post samples
Parameters:	None
Response:	Number of post samples
Example:	:pol:swe:trig:post:samp? → +2000
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:CONTinuous:SACCuracy**
Syntax:	:POLarimeter:SWEep:CONTinuous:SACCuracy
Description:	Sets the SOP accuracy in grad used in the continuous sampling mode. A value of 0 causes all samples to be stored. Otherwise SOP changes smaller than SOP accuracy will not cause a sample to be stored.
Parameters:	SOP accuracy in grad
Example:	:pol:swe:cont:sacc 0.1
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:SWEep:CONTinuous:SACCuracy?**
Syntax:	:POLarimeter:SWEep:CONTinuous:SACCuracy?
Description:	Returns the SOP accuracy in grad used in the continuous sampling mode.
Parameters:	None
Response:	SOP accuracy in grad
Example:	:pol:swe:cont:sacc? → +0.00000000E+00
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:TRIGger:INPut**
Syntax:	:POLarimeter:TRIGger:INPut<wsp><value>
Description:	Defines the trigger input
Parameters:	NONE: Don't wait for trigger TTLHIGH: Wait for rising edge on BNC input TTLLOW: Wait for falling edge on BNC input SOPCHANGE: Wait for specified change of SOP PRETRIGGER_TTLHIGH: Specified numbers of samples are logged before rising edge on BNC input PRETRIGGER_TTLLOW: Specified numbers of samples are logged before falling edge on BNC input SMEasure: One sample for every trigger will be logged CMEasure: One trigger excutes one loop with specified number of samples MMEasure: One trigger executes specified number of loops with specified number of samples THReshold: Define minimum and maximum treshold power limits for triggering. You have to pass two paramters additionally. If you don't want minimum or maximum limit, send NAN instead of number.
Example:	:pol:trig:inp NONE pol:trig:inp THR,1mW,5mW
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:TRIGger:DELay**
Syntax:	:POLarimeter:TRIGger:DELay<wsp><value>
Description:	Defines factor for delay. delay = factor/32 MHz That means, trigger event is excuted after specified time after input trigger.

Parameters:	Integer number between 0-997 (including limits)
Example:	:pol:trig:del 10
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:TRIGger:DELay?**
Syntax:	:POLarimeter:TRIGger:DELay?
Description:	Returns factor for delay. delay = factor/32 MHz
Parameters:	None
Response:	Factor
Example:	:pol:trig:del? → +0
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:TRIGger:OFFSet**
Syntax:	:POLarimeter:TRIGger:OFFSet
Description:	Defines number of triggers, which will be ignored before first trigger event.
Parameters:	Trigger offset (integer)
Example:	:pol:trig:offs 10
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:TRIGger:OFFSet?**
Syntax:	:POLarimeter:TRIGger:OFFSet?
Description:	Returns number of triggers, which will be ignored before first trigger event.
Parameters:	None
Response:	Trigger offset
Example:	:pol:trig:offs? → +0
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:TRIGger:INPut?**
Syntax:	:POLarimeter:TRIGger:INPut?
Description:	Returns input triger configuration.
Parameters:	None
Response:	Explanation of the parameter see: POLarimeter:TRIGger:INPut
Example:	:pol:trig:inp? → NONE
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:TRIGger:OUTPut**
Syntax:	:POLarimeter:TRIGger:OUTPut
Description:	Defines trigger output
Parameters:	DISabled: no trigger at output BNC AVGover: trigger when averaging starts at output BNC MEASure: trigger when measuring starts at output BNC
Example:	:pol:trig:outp dis
Affects	N7781C / N7786C / N7788C

Command:	**:POLarimeter:TRIGger:OUTPut?**
Syntax:	:POLarimeter:TRIGger:OUTPut?
Description:	Returns trigger output configuration
Parameters:	None
Response:	see POLarimeter:TRIGger:OUTPut
Example:	:pol:trig:outp? → DISabled
Affects	N7781C / N7786C / N7788C

Polarization Synchronous Scrambler Commands

Command:	**:PCONtroller:STOP**
Syntax:	:PCONtroller:STOP
Description:	This command stops the Polcontroller.
Parameters:	None
Response:	None
Example:	:PCONtroller:STOP
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:WPLAtes**
Syntax:	:PCONtroller:WPLAtes<wsp><orientation waveplate 1>,<retardation waveplate1>,...
Description:	Sets the orientation and retardation of the polarization controller waveplates. The data consists of 12 double values. The order is given by: Orientation Waveplate 1, Retardation Waveplate 1, Orientation Waveplate 2, Retardation Waveplate 2, Orientation Waveplate 3, Retardation Waveplate 3, Orientation Waveplate 4, Retardation Waveplate 4, Orientation Waveplate 5, Retardation Waveplate 5, Orientation Waveplate 6, Retardation Waveplate 6 The orientation is given in grad, the retardation is given in fractions of Lambda. A retardation of 0.25 corresponds to a quarter waveplate. The maximum retardation value is 0.25.
Parameters:	Double array with 12 values
Response:	None
Example:	:PCONtroller:WPLAtes 10,0.25,20,0.25,30,0.25,40,0.25,50,0.25,60,0.25
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:WPLAtes?**
Syntax:	:PCONtroller:WPLAtes?
Description:	Returns the orientation and retardation of the polarization controller waveplates
Parameters:	None

Response:	Double array (comma separated values)
Example:	:PCONtroller:WPLAtes? -> +0.00000000E+00,+2.50000000E-01,+0.00000000E+00,+2.50000000E-01,+0.00000000E+00,+2.50000000E-01,+0.00000000E+00,+2.50000000E-01,+0.00000000E+00,+2.50000000E-01,+0.00000000E+00,+2.50000000E-01
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence:LENGth**
Syntax:	:PCONtroller:SEQuence:LENGth<wsp><length>
Description:	Set the polarization controller sequence length.
Parameters:	Integer value (0-100000)
Response:	None
Example:	:PCONtroller:SEQuence:LENGth 5
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence:LENGth?**
Syntax:	:PCONtroller:SEQuence:LENGth?
Description:	Get the polarization controller sequence length.
Parameters:	None
Response:	Int value
Example:	:PCONtroller:SEQuence:LENGth? -> +1
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence:RRATe**
Syntax:	:PCONtroller:SEQuence:RRATe
Description:	Set the polarization controller repetition rate for each data point in kHz.
Parameters:	Double value

Response:	None
Example:	:PCONtroller:SEQuence:RRATe 50.2
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence:RRATe?**
Syntax:	:PCONtroller:SEQuence:RRATe?
Description:	Get the polarization controller repetition rate for each datapoint in kHz
Parameters:	None
Response:	Double value
Example:	:PCONtroller:SEQuence:RRATe? +5.02000000E+01
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence:SMODe**
Syntax:	:PCONtroller:SEQuence:SMODe<wsp><mode>
Description:	Sets the sequence mode. A sequence consist of the number of datapoints multiplied by the repetition (repetition=0 means endless) 0 = Sequence will automatically start and restart when reaching the end. 1 = Sequence will automatically start and run only once (like repetition=1). 2 = Sequence will start when a low/high-transition is detected on the BNC input. When reaching the end of the sequence, it will be restarted. 3 = Same as above, except that the sequence will not be repeated. 4 to 6 = sequence stepped mode - single extern trigger config waveform, define repetitions / repetition rate , set SyncMode, start/arm scrambler (:PCON:STARt)
Parameters:	Integer value
Response:	None
Example:	:PCONtroller:SEQuence:SMODe 0
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence:SMODe?**
Syntax:	:PCONtroller:SEQuence:SMODe?
Description:	Returns the current configured Sequence Mode 0 = Sequence will automatically start and restart when reaching the end. 1 = Sequence will automatically start and run only once. 2 = Sequence will start when a low/high-transition is detected on the BNC input. When reachig the end of the sequence, it will be restarted. 3 = Same as above, except that the sequence will not be repeated. 4 to 6 = sequence stepped mode - single extern trigger
Parameters:	None
Response:	Integer value
Example:	:PCONtroller:SEQuence:SMODe? +0
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence:DCOMpensation**
Syntax:	:PCONtroller:SEQuence:DCOMpensation<wsp><value>
Description:	Set the polarization controller drift-compensation. 0 = drift-compensation is OFF (dynamic correction is inactive) 1 = drift-compensation is ON (dynamic correction is active) (DEFAULT)
Parameters:	Boolean integer value
Response:	None
Example:	:PCONtroller:SEQuence:DCOMpensation 1
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence:DCOMpensation?**
Syntax:	:PCONtroller:SEQuence:DCOMpensation?
Description:	Get the polarization controller drift-compensation. 0 = drift-compensation is OFF (dynamic correction is inactive) 1 = drift-compensation is ON (dynamic correction is active)
Parameters:	None

Response:	Boolean integer value
Example:	:PCONtroller:SEQuence:DCOMpensation? 1
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence:HOLDoff**
Syntax:	:PCONtroller:SEQuence:HOLDoff<wsp><number>
Description:	Define an external trigger delay after changing the SOP value
Parameters:	Integer value (0-32766) 1 means 1/32 us
Response:	None
Example:	:PCONtroller:SEQuence:HOLDoff 10
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence:HOLDoff?**
Syntax:	:PCONtroller:SEQuence:HOLDoff?
Description:	Get the external trigger delay after changing the SOP value
Parameters:	None
Response:	Integer value (0-32766) 1 means 1/32 us
Example:	:PCONtroller:SEQuence:HOLDoff? +1
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence**
Syntax:	:PCONtroller:SEQuence<wsp><488.2 Binary Block>
Description:	Sets the polarization controller sequence using waveplate positions and retardations. The parameter list is a series of floating point values. Each waveplate is controlled by two float values: Orientation in grad and Retardation in fractions of Lambda (0.25 corresponds to Lambda 4). Thus, each sequence row contains 12 float values. Hence, the series length has to be a multiple of 12. The order is given by: Orientation Waveplate 1, Retardation Waveplate 1, Orientation Waveplate 2, Retardation Waveplate 2, Orientation Waveplate 3, Retardation Waveplate 3, Orientation Waveplate 4, Retardation Waveplate 4, Orientation Waveplate 5, Retardation Waveplate 5, Orientation Waveplate 6, Retardation Waveplate 6

Parameters:	488.2 Binary Block <real32 / double>
Response:	None
Example:	:PCONtroller:SEQuence <488.2 Binary Block>
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence?**
Syntax:	:PCONtroller:SEQuence?
Description:	Read the last transferred sequence the polarization controller sequence using waveplate positions and retardations Return value: binary block (float values / 32 bit). Each point consist of Orientation in grad and Retardation in fractions of Lambda (0.25 corresponds to Lambda 4). Each sequence data point contains 12 float values. The order is given by: Orientation Waveplate 1, Retardation Waveplate 1, Orientation Waveplate 2, Retardation Waveplate 2, Orientation Waveplate 3, Retardation Waveplate 3, Orientation Waveplate 4, Retardation Waveplate 4, Orientation Waveplate 5, Retardation Waveplate 5, Orientation Waveplate 6, Retardation Waveplate 6
Parameters:	None
Response:	488.2 Binary Block <real32 / double>
Example:	:PCONtroller:SEQuence? <488.2 Binary Block>
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence:SEQVoltage**
Syntax:	:PCONtroller:SEQuence:SEQVoltage<wsp><488.2 Binary Block>
Description:	Sets the polarization controller sequence using the DAC connected to the polarization controller
Parameters:	488.2 Binary Block <word / short>
Response:	None
Example:	:PCONtroller:SEQuence:SEQVoltage <488.2 Binary Block>
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SEQuence:SEQVoltage?**
Syntax:	:PCONtroller:SEQuence:SEQVoltage?
Description:	Get the polarization controller sequence DAC voltage sequence
Parameters:	None
Response:	488.2 Binary Block <word / short>
Example:	:PCONtroller:SEQuence:SEQVoltage?
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SCRambler:ENABle**
Syntax:	:PCONtroller:SCRambler:ENABle
Description:	Enables (1) or disables (0) the Scrambler (same function like :PCONtroller:STARt and :PCONtroller:STOP).
Parameters:	Integer value 1 = Polcontroller is running 0 = Polcontroller is disables
Response:	None
Example:	:PCONtroller:SCRambler:ENABle 1
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SCRambler:ENABle?**
Syntax:	:PCONtroller:SCRambler:ENABle?
Description:	Returns the current polcontroller state
Parameters:	None
Response:	Integer value 1 = Polcontroller is running 0 = Polcontroller is disables
Example:	:PCONtroller:SCRambler:ENABle? Return +0
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:STAGe<stage>:DEGree**
Syntax:	:PCONtroller:STAGe<stage>:DEGree<wsp><orientation>
Description:	Set the orientation for each waveplate (lambda/4 plate)
Parameters:	<stage> integer value between 1 to 6 <orientation> float value from 0.0 to 359.999
Response:	None
Example:	:pcon:stag1:deg 75.4
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:STAGe<stage>:DEGree?**
Syntax:	:PCONtroller:STAGe<stage>:DEGree?
Description:	Get the orientation for each waveplate (lambda/4 plate)
Parameters:	<stage> integer value from 1 to 6
Response:	Float value
Example:	:PCON:STAG5:DEG? -> +2.30000000E+01
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:STARt**
Syntax:	:PCONtroller:STARt
Description:	Start the polarization sequence with the configured repetitions, repetition rate, and Sequence Mode
Parameters:	None
Response:	None
Example:	:pcon:star
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:REPetition**
Syntax:	:PCONtroller:REPetition<wsp><repetitions>
Description:	Sets the polarization controller repetition
Parameters:	<repetitions> NOTE: 0 means endless
Response:	None
Example:	:pcon:rep 10
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:REPetition?**
Syntax:	:PCONtroller:REPetition?
Description:	Get the polarization controller repetition
Parameters:	None
Response:	Integer value between 0 to 2147483647
Example:	:pcon:rep -> +10
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:LOAD:DATA?**
Syntax:	:PCONtroller:LOAD:DATA?<wsp><filename>
Description:	Loads a specified arb segment from the internal memory
Parameters:	Quoted string filename
Response:	status message: "load x datapoints" error message: "Error on dataload"
Example:	:PCON:LOAD:DATA? "random.bin" -> "load 40 datapoints"
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SAVE:ARB?**
Syntax:	:PCONtroller:SAVE:ARB?
Description:	Save the current arb sequence in nonvolatile memory
Parameters:	None
Response:	"save 4 datapoints"
Example:	:pcon:save:arb? -> "save 4 datapoints"
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:LOAD:ARB?**
Syntax:	:PCONtroller:LOAD:ARB?
Description:	Load an arbitrary waveform
Parameters:	None
Response:	Status message: "load x datapoints" Error message: "Error on dataload"
Example:	:pcon:load:arb? -> "load 4 datapoints"
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:GEN:RANDom?**
Syntax:	:PCONtroller:GEN:RANDom? <datapoints>,<stepwidth>
Description:	Generate a random walk sequence with a maximum stepwidth
Parameters:	<datapoints> integer value between 0 to 1000000 <stepwidth> integer value between 0 to 65535
Response:	String "GEN DONE"
Example:	:PCON:GEN:RAND? 4,2000' -> "GEN DONE"
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:GEN:SCRAmble?**
Syntax:	:PCONtroller:GEN:SCRAmble?<wsp><number>
Description:	Generate a random sequence. Example: :PCON:GEN:SCRA? #Datapoints Example: :PCON:GEN:SCRA 120000 Return Value = status string "GEN DONE"
Parameters:	<number> integer value from 1-1000000
Response:	Status message: "GEN DONE" when the generation is finished
Example:	:PCONtroller:GEN:SCRAmble? 300' -> "GEN DONE"
Affects	N7785C / N7786C / N7788C

Command:	**:PCONtroller:SWITch**
Syntax:	:PCONtroller:SWITch
Description:	Sets the optical switch in the N7788C
Parameters:	Integer value 0 = optical Scrambler Input signal will be routed to the internal polarimerer 1 = optical Scrambler Input signal will be routed to the Scrambler Output
Response:	None
Example:	:pcon:swit 1
Affects	N7788C

Command:	**:PCONtroller:SWITch?**
Syntax:	:PCONtroller:SWITch?
Description:	Gets the optical switch build in the N7788C NOTE: this value is the last set value to set the latching MEMS-switch!
Parameters:	None

Response:	Integer value 0 = optical Scrambler Input signal will be routed to the internal polarimerer 1 = optical Scrambler Input signal will be routed to the Scrambler Output
Example:	:pcon:swit? -> +0
Affects	N7788C

Polarization Synthesizer / Polarization Component Analyzer Commands

Command:	**:STABilizer:STABilize**
Syntax:	:STABilizer:STABilize<wsp> <value>
Description:	Sets the Stabilizer Operating Mode. NOTE - The stabilizer function configures and controls both, the Scrambler and the Polarimeter including intern and extern Trigger Configuration. After Stopping the Stabilizer Mode use System:Preset or *RST to activate the normal Scrambler and Polarimeter Mode NOTE - Use only gain between 0 and 7 for Stabilizer Mode because 8 and 9 have a small Bandwith e.g. POL:GAIN 6 Auto Gain is disabled when using Logging / Sweep and Stabilizer Mode For best results check the Leveling of the Polarimeter. Look at the POL:SWE:LPR? command. Maximum Speed is at Gain 0 to 5 Even Gains have Factors of 10 and odd Gains Factors of 8. Choose what fits better. NOTE - If the stabilizer function is active then most other Commands to Scrambler and Polarimeter including Zeroing are blocked. NOTE - An "On Startup Stabilize Mode" where stabilizing is automatically on after reboot can be enabled by saving the Configuration in one of the 10 Setting Positions e.g. with following commands: :stab:sop 1,0,0 :stab:stab 1 :conf:meas:sett:save 1
Parameters:	0 = Stabilizer Off (Default) 1 = Stabilizer On
Response:	None
Example:	:STAB:STAB 1
Affects	N7785C, N7786C, N7788C

Command:	**:STABilizer:STABilize?**
Syntax:	:STABilizer:STABilize?
Description:	Gets the Stabilizer Operating Mode.
Parameters:	None
Response:	0 = Stabilizer Off (Default) 1 = Stabilizer On
Example:	:STAB:STAB? → 1
Affects	N7785C, N7786C, N7788C

Command:	**:STABilizer:SOP**
Syntax:	:STABilizer:SOP<wsp> <s1>,<s2>,<s3>[,<Factor>]
Description:	Sets the target SOP for the Stabilizer. ▪ For the N7786C and N7788C which have a build in Polarimeter the target SOP must be provided as a 3-element normalized Stokes vector (s1,s2,s3 with DOP=1). If the DOP is not 1, then it will be normalized before processing. ▪ For stabilizing with the N7785C, the fourth parameter must be used. The first 3 parameters (s1, s2,s3) are ignored. The feedback parameter (<Factor>) is a positive real number and is determined from a measurement at the DUT that depends on the incoming polarization, such as a power measurement of the output signal from the DUT. By repeatedly sending updates of this value to the N7785C, the instrument adjusts its output polarization to minimize the parameter value after sending a 'TRIG 1' command. Further details can be found in Applications Note 3120-1564 (Polarization Alignment Methods). ▪ The procedure for stabilizing with the N7785C is: ▪ STABilizer:STABilize 0 to deactivate stabilization ▪ STABilizer:STABilize 1 to activate stabilization ▪ STABILIZER:SOP 0,0,0,<Factor> to enter the feedback parameter ▪ TRIG 1 to trigger the next stabilization dithering step ▪ The last 2 steps should be repeated with updated values
Parameters:	None
Response:	None
Example:	N7786C/88C: → stab:sop 0,1,0 N7785C: → stab:sop 0,0,0,1.234
Affects	N7785C, N7786C, N7788C

Command:	**:STABilizer:SOP?**
Syntax:	:STABilizer:SOP?
Description:	Gets the target SOP. NOTE - For the N7786C/88C, when the Stabilizer invert is On, then the STABilizer:SOP? command returns the inverted Target SOP which is the SOP the Stabilizer uses.
Parameters:	None
Response:	A 4-element Stokes vector.
Example:	N7786C/88C: stab:sop? → +0.00000000E+00,+1.00000000E+00,+0.00000000E+00 N7785C: stab:sop? → +0.00000000E+00,+0.00000000E+00,+0.00000000E+00,+1.23400000E+00
Affects	N7785C, N7786C, N7788C

6 Error Codes

This chapter gives information about error codes used with the N778-C series tunable laser source instruments.

Error Strings

Error strings in the range -100 to -183 are defined by the SCPI standard, downloadable from: http://www.ivifoundation.org/docs/scpi-99.pdf

String descriptions taken from this standard (VERSION 1999.0 May, 1999), whether in whole or in part, are enclosed by [ ].

Table 1 Overview for Supported Strings

Error Number	String
0	"No error"
-100	"Command Error" [This is the generic syntax error used when a more specific error cannot be detected. This code indicates only that a Command Error as defined in *IEEE 488*.2,11.5.1.1.4 has occurred.]
-101	"Invalid character" [A syntactic element contains a character which is invalid for that type; for example, a header containing an ampersand, SETUP&. This error might be used in place of error -114 and perhaps some others.]
-102	"Syntax error" [An unrecognized command or data type was encountered; for example, a string was received when the **device** does not accept strings.]
-103	"Invalid separator" [The parser was expecting a separator and encountered an illegal character; for example, the semicolon was omitted after a program message unit]
-104	"Data type error" [The parser recognized a data element different than one allowed; for example,numeric or string data was expected but block data was encountered.]
-105	"GET not allowed" [A Group Execute Trigger was received within a program message (see *IEEE488*.2, 7.7).]
-108	"Parameter not allowed" [More parameters were received than expected for the header]
-109	"Missing parameter" [Fewer parameters were received than required for the header]
-110	"Command header error"
-111	"Header separator error"

Error	
Number	String
-112	"Program mnemonic too long" [The header contains more than twelve characters (see *IEEE 488*.2, 7.6.1.4.1).]
-113	"Undefined header" [The header is syntactically correct, but it is undefined for this specific **devic**e; for example, *XYZ is not defined for any device.]
-114	"Header suffix out of range"
-115	"Unexpected number of parameters"
-120	"Numeric data error" [This error, as well as errors -121 through -129, are generated when parsing a data element which appears to be numeric, including the nondecimal numeric types. This error message is used if the **device** cannot detect a more specific error.]
-121	"Invalid character in number" [An invalid character for the data type being parsed was encountered; for example, an alpha in a decimal numeric]
-123	"Exponent too large" [The magnitude of the exponent was larger than 32000 (see *IEEE 488*.2,7.7.2.4.1).]
-124	"Too many digits" [The mantissa of a decimal numeric data element contained more than 255 digits excluding leading zeros (see *IEEE 488*.2, 7.7.2.4.1).]
-128	"Numeric data not allowed" [A legal numeric data element was received, but the **device** does not accept one in this position for the header.]
-130	"Suffix error"
-131	"Invalid suffix" [The suffix does not follow the syntax described in *IEEE 488*.2, 7.7.3.2, or thesuffix is inappropriate for this **devic**e.]
-134	“Suffix too long” [The suffix contained more than 12 characters (see *IEEE 488*.2, 7.7.3.4).]
-138	“Suffix not allowed” [A suffix was encountered after a numeric element which does not allow suffixes.]
-140	"Character data error"
-141	“Invalid character data” [Either the character data element contains an invalid character or the particular element received is not valid for the header.]
-144	"Character data too long"

Error Number	String
-148	“Character data not allowed” [A legal character data element was encountered where prohibited by the **devic**e.]
-150	“String data error” [This error, as well as errors -151 through -159, are generated when parsing a string data element. This error message is used when the **device** cannot detect a more specific error.]
-151	“Invalid string data” [A string data element was expected, but was invalid for some reason (see *IEEE 488.*2, 7.7.5.2); for example, an END message was received before the terminal quote character.]
-158	“String data not allowed” [A string data element was encountered but was not allowed by the **device** at this point in parsing.]
-160	"Block data error"
-161	“Invalid block data” [A block data element was expected, but was invalid for some reason (see *IEEE 488.*2, 7.7.6.2); for example, an END message was received before the length was satisfied.]
-168	“Block data not allowed” [A legal block data element was encountered but was not allowed by the **device** at this point in parsing.]
-170	“Expression error” [This error, as well as errors -171 through -179, are generated when parsing an expression data element. This particular error message is used when the **device** cannot detect a more specific error.]
-171	“Invalid expression” [The expression data element was invalid (see *IEEE 488.*2, 7.7.7.2); for example, unmatched parentheses or an illegal character.]
-178	“Expression data not allowed” [A legal expression data was encountered but was not allowed by the **device** at this point in parsing.]
-180	"Macro error"
-181	“Invalid outside macro definition” [Indicates that a macro parameter placeholder ($<number) was encountered outside of a macro definition.]
-183	“Invalid inside macro definition” [Indicates that the program message unit sequence, sent with a *DDT or *DMC command, is syntactically invalid (see *IEEE 488.*2, 10.7.6.3).]
-184	"Macro parameter error"

Error Number	String
-185	“Subop out of range” *Description:* Suboperations are parameters that are passed to refine the destination of a command. They are used to address slots, channels, laser selections and GPIB/SCPI register levels. This error is generated if the parameter is not valid in the current context or system configuration. *Example:* This error occurs if the user queries the status of a summary register and passes an invalid status level (also see "Status for 816x" on page 28 programmer's guide). *Note:* Incorrect slots and channels addresses are handled by error code -301
-200	“Execution error (StatExecError)" *Description:* This error occurs when the current function, instrument or module state (or status) prevents the execution of a command. This is a generic error which can occur for a number of reasons. *Example:* When a powermeter has finished a logging application and data is available, the user is not able to reconfigure the logging application parameters. First, the user must stop the logging application.
-201	"Invalid while in local" "Please be patient - GPIB currently locked out" *Description:* Some operations block the complete system. Since no sensible measurements are possible while this is true, the GPIB is locked out. *Example:* When ARA, Lambda zeroing or zeroing is executing on a TLS module, the GPIB is not accessible.
-202	"Settings lost due to rtl"
-203	"Command protected"
-210	"Trigger error"
-211	“Trigger ignored” *Description:* A trigger has been detected but ignored because of timing contraints. (For Example: average time to large).
-212	“Arm ignored” *Description:* The user can set the automatic re-arming option for input and output trigger events (see Error Strings on page 88). When this error occurs, the device ignores the setting because the current module status does not allow the change of trigger settings.
-213	“Init ignored” *Description:* The INIT:IMM command initiates a trigger and completes a full measurement cycle. The continuous measurement must be DISABLED. This error code is generated if the module is still in cont. measurement mode.

Error	
Number	String
-214	"Trigger deadlock"
-215	"Arm deadlock"
-220	"Parameter error (StatParmError)” *Description:* The user has passed a parameter that cannot be changed in this way. The device cannot detect one of the following more specific errors:
-220	-220, "Parameter error (StatParmOutOfRange)" *Description:* The user has passed a parameter that exceeds the valid range for this parameter.
-220	"Parameter error (StatParmIllegalVal)" *Description:* The user has passed a parameter that does not match a value in a list of possible values.
-221	"Settings conflict (StatParmInconsistent)" *Description:* The user has passed a parameter that conflicts with other already configured parameters. *Example:* There are constrains for TLS sweep parameters: this error is generated when lambda step size exceeds the difference between start and stop wavelength. If error -221 is returned after you try to start a wavelength sweep, one of the following cases of sweep parameter inconsistency has occurred: Continuous Sweep mode AND l Start is less than l Stop. Continuous Sweep mode AND Sweep Time is too short. Adjust Sweep Speed, l Start, or l Stop. Continuous Sweep mode AND Sweep Time is too long. Adjust Sweep Speed, l Start, or l Stop. Continuous Sweep mode AND Trigger Frequency is too high. Adjust Step Size. Trigger Frequency is the Sweep Speed divided by the Step Size. Stepped Sweep mode AND Lambda Logging Enabled. Continuous Sweep mode AND Lambda Logging Enabled AND Output trigger mode not set to STFinished (Step finished). Continuous Sweep mode AND Lambda Logging is Enabled AND Modulation Source is not set to OFF. Continuous Sweep mode AND Lambda Logging is Enabled AND Sweep Cycles is not set to 1.
-222	"Data out of range (StatParmTooLarge)" *Description:* The user has passed a continuous parameter that is too large. *Example:* Wavelength 1800nm when maximum wavelength is 1700nm.
-222	"Data out of range (StatParmTooSmall)" *Description:* The user has passed a continuous parameter that is too small. *Example:* Wavelength 700nm when minimum wavelength is 800nm.

Error	
Number	String
-223	“Too much data” *Description:* A function returns more data or the user requests more data than the application is able to handle. *Example:* A tunable laser source produces more data when lambda values of a sweep are stored than the 816x instrument is able to handle. Use the new SENSE:FUNC:RES:BLOCK? command to split the data aquisition into multiple parts.
-224	“Illegal parameter value” [Used where exact value, from a list of possibles, was expected.]
-225	"Out of memory" *Description:* The request application or function cannot be executed because the instrument runs out of memory.
-226	"Lists not same length"
-230	"Data corrupt or stale"
-231	"Data questionable (StatValNYetAcc)" *Description:* The data that is retured is not accurate or reliable. The user should repeat the operation. The reason for this error is unspecific. *Example:* A powermeter configured a long average time has not completed its current measurement cycle when the user queries the current power.
-231	"Data questionable (StatRangeTooLow)" *Description:* As -231 (StatValNYetAcc) but for a more specific reason: The powermeter readout data is not reliable because the currently set (manual) range does not correspond with the input power.
-240	"Hardware error"
-241	"Hardware missing"
-250	"Mass storage error"
-251	"Missing mass storage"
-252	"Missing media"
-253	"Corrupt media"
-254	"Media full"
-255	"Directory full"
-256	"File name not found"

Error	
Number	String
-257	"File name error"
-258	"Media protected"
-260	"Expression error"
-261	"Math error in expression (StatUnitCalculationError)" *Description:* This may occur when the user attempts to transform data in a way that is currently not possible. *Example:* When a powermeter is measuring very small power values in dBm (such as noise power), negative power values in Watt may also be present (such as when the powermeter calibration wavelength does not correspond to the wavelength of input signal). The instrument cannot transform negative Watt values to dBm because the logarithm of a negative value is not defined.
-270	"Macro error"
-271	"Macro syntax error"
-272	“Macro execution error” [Indicates that a syntactically legal macro program data sequence could not beexecuted due to some error in the macro definition (see IEEE 488.2, 10.7.6.3.)]
-273	“Illegal macro label” [Indicates that the macro label defined in the *DMC command was a legal string syntax, but could not be accepted by the **device** (see *IEEE 488.*2, 10.7.3 and 10.7.6.2); for example, the label was too long, the same as a common command header, or contained invalid header syntax.]
-274	"Macro parameter error"
-275	"Macro definition too long"
-276	“Macro recursion error” [Indicates that a syntactically legal macro program data sequence could not be executed because the device found it to be recursive (see *IEEE 488.*2, 10.7.6.6).]
-277	“Macro redefinition not allowed” [Indicates that a syntactically legal macro label in the *DMC command could not be executed because the macro label was already defined (see *IEEE 488.*2,10.7.6.4).]
-278	“Macro header not found” [Indicates that a syntactically legal macro label in the *GMC? query could not be executed because the header was not previously defined.]
-280	"Program error"
-281	"Cannot create program"
-282	"Illegal program name"

Error	
Number	String
-283	"Illegal variable name"
-284	"Function currently running (StatModuleBusy)" *Description:* This error is generated when a function is currently running on a module so that it cannot process another command. *Example:* When a powermeter is running a logging application, you are not able to configure the logging application parameters (also see -200).
-285	"Program syntax error"
-286	"Program runtime error"
-290	"Memory use error"
-291	"Out of memory"
-292	"Referenced name does not exist"
-293	"Referenced name already exists"
-294	"Incompatible type"
-300	"Device-specific error"
-303	"Module slot empty or slot / channel invalid" *Description:* The user has send a command to an empty slot.
-310	“System error” [Indicates that some error, termed “system error” by the device, has occurred. This code is device-dependent.]
-311	"Memory error"
-312	"PUD memory lost"
-313	"Calibration memory lost"
-314	"Save/recall memory lost"
-315	"Configuration memory lost"
-320	"Storage fault"
-321	“Out of memory” [An internal operation needed more memory than was available.]

Error Number	String
-330	“Self-test failed” *Description:* You have started the self test, but the module has detected an error while executing it
-340	"Calibration failed"
-350	“Queue overflow” [A specific code entered into the queue in lieu of the code that caused the error. This code indicates that there is no room in the queue and an error occurred but was not recorded.]
-360	"Communication error"
-361	"Parity error in program message"
-362	"Framing error in program message"
-363	"Input buffer overrun"
-365	"Time out error"
-368	"LambdaStop<=LambdaStart"
-369	"sweepTime < min"
-370	"sweepTime > max"
-371	"triggerFreq > max"
-372	"step < min"
-373	"triggerNum > max"
-374	"LambdaLogging = On AND Modulation = On AND ModulationSource! = CoherenceControl"
-375	"LambdaLogging = On AND TriggerOut! = StepFinished"
-376	"Lambda logging in stepped mode"
-377	"step not multiple of 0.1pm"
-378	"triggerFreq < min"
-400	“Query error” [This is the generic query error for **devices** that cannot detect more specific errors. This code indicates only that a Query Error as defined in *IEEE 488*.2, 11.5.1.1.7 and 6.3 has occurred.]
-410	“Query INTERRUPTED” [Indicates that a condition causing an INTERRUPTED Query error occurred (see *IEEE 488.2*, 6.3.2.3); for example, a query followed by DAB or GET before a response was completely sent.]

Error	
Number	**String**
-420	“Query UNTERMINATED” [Indicates that a condition causing an UNTERMINATED Query error occurred (see *IEEE 488.*2, 6.3.2.2); for example, the **device** was addressed to talk and an incomplete program message was received.]
-430	“Query DEADLOCKED” [Indicates that a condition causing an DEADLOCKED Query error occurred (see *IEEE 488.*2, 6.3.1.7); for example, both input buffer and output buffer are full and the device cannot continue.]
-440	“Query UNTERMINATED after indef resp” [Indicates that a query was received in the same program message after an query requesting an indefinite response was executed (see *IEEE 488.*2, 6.5.7.5).]

KEYSIGHT
TECHNOLOGIES

This information is subject to
change without notice.

Edition 2.0, October 2021

N7780-90C02

www.keysight.com