Regenerative DC Electronic Load

IT3800 Series Programming Guide

Model: IT3800 Series
Version: V1.0/01,2022

Notices

Manual Part Number

402225

Trademarks

Pentium is U.S. registered trademarks of Intel Corporation.
Microsoft, Visual Studio, Windows and MS Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries and regions.

Warranty

The materials contained in this document are provided “as is”, and is subject to change, without prior notice, in future editions. Further, to the maximum extent permitted by applicable laws, ITECH disclaims all warrants, 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. ITECH shall not be held liable for errors or for incidental or indirect damages in connection with the furnishing, use or application of this document or of any information contained herein. Should ITECH and the user enter into a separate written agreement with warranty terms covering the materials in this document that conflict with these terms, the warranty terms in the separate agreement shall prevail.

Technology Licenses

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

Restricted Rights Legend

Restricted permissions of the U.S. government. Permissions for software and technical data which are authorized to the U.S. Government only include those for custom provision to end users. ITECH follows FAR 12.211 (technical data), 12.212 (computer software). DFARS 252.227-7015 (technical data–commercial products) and DFARS 227.7202-3 (permissions for commercial computer software or computer software documents) while providing the customized business licenses of software and technical data.

Safety Notices

CAUTION

A CAUTION sign denotes a hazard. It calls attention to an operating procedure or practice 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 sign until the indicated conditions are fully understood and met.

WARNING

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

Note

A NOTE sign denotes important hint. It calls attention to tips or supplementary information that is essential for users to refer to.

Quality Certification and Assurance

We certify that series instrument meets all the published specifications at time of shipment from the factory.

Warranty

ITECH warrants that the product will be free from defects in material and workmanship under normal use for a period of one (1) year from the date of delivery (except those described in the Limitation of Warranty below).

For warranty service or repair, the product must be returned to a service center designated by ITECH.

- The product returned to ITECH for warranty service must be shipped PREPAID. And ITECH will pay for return of the product to customer.
- If the product is returned to ITECH for warranty service from overseas, all the freights, duties and other taxes shall be on the account of customer.

Limitation of Warranty

This Warranty will be rendered invalid in case of the following:

- Damage caused by circuit installed by customer or using customer own products or accessories;
- Modified or repaired by customer without authorization;
- Damage caused by circuit installed by customer or not operating our products under designated environment;
- The product model or serial number is altered, deleted, removed or made illegible by customer;
- Damaged as a result of accidents, including but not limited to lightning, moisture, fire, improper use or negligence.

Safety Symbols

	Direct current		ON (power)
	Alternating current		OFF (power)
	Both direct and alternating current		Power-on state
	Chassis (earth ground) symbol.		Power-off state
	Earth (ground) terminal		Reference terminal
	Caution		Positive terminal
	Warning (refer to this manual for specific Warning or Caution information)		Negative terminal
	A chassis terminal	-	-

Safety Precautions

The following safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or specific warnings elsewhere in this manual will constitute a default under safety standards of design, manufacture and intended use of the instrument. ITECH assumes no liability for the customer's failure to comply with these precautions.

WARNING

- **Do not use the instrument if it is damaged. Before operation, check the casing to see whether it cracks. Do not operate the instrument in the presence of inflammable gasses, vapors or dusts.**
- **The instrument is provided with a power cord during delivery and should be connected to a socket with a protective earth terminal, a junction box or a three-phase distribution box. Before operation, be sure that the instrument is well grounded.**
- **Please always use the provided cable to connect the instrument.**
- **Check all marks on the instrument before connecting the instrument to power supply.**
- **Ensure the voltage fluctuation of mains supply is less than 10% of the working voltage range in order to reduce risks of fire and electric shock.**
- **Do not install alternative parts on the instrument or perform any unauthorized modification.**
- **Do not use the instrument if the detachable cover is removed or loosen.**
- **To prevent the possibility of accidental injuries, be sure to use the power adapter supplied by the manufacturer only.**
- **We do not accept responsibility for any direct or indirect financial damage or loss of profit that might occur when using the instrument.**
- **This instrument is used for industrial purposes, do not apply this product to IT power supply system.**
- **Never use the instrument with a life-support system or any other equipment subject to safety requirements.**

WARNING

- **SHOCK HAZARD Ground the Instrument. This product is provided with a protective earth terminal. To minimize shock hazard, the instrument must be connected to the AC mains through a grounded power cable, with the ground wire firmly connected to an electrical ground (safety ground) at the power outlet or distribution box. Any interruption of the protective (grounding) conductor or disconnection of the protective earth terminal will cause a potential shock hazard that could result in injury or death.**
- **Before applying power, verify that all safety precautions are taken. All connections must be made with the instrument turned off, and must be performed by qualified personnel who are aware of the hazards involved. Improper actions can cause fatal injury as well as equipment damage.**
- **SHOCK HAZARD, LETHAL VOLTAGES This product can input the dangerous voltage that can cause personal injury, and the operator must always be protected from electric shock. Ensure that the input electrodes are either insulated or covered using the safety covers provided, so that no accidental contact with lethal voltages can occur.**
- **Never touch cables or connections immediately after turning off the instrument. Verify that there is no dangerous voltage on the electrodes or sense terminals before touching them.**
- **After using the device, turn off the power switch of the device before unplugging the power cord or disassembling the terminals. Do not touch the cable or the terminal immediately. Depending on the model, the dangerous voltage at the plug or terminal is maintained for 10 seconds after the device is switched off. Make sure that there is no dangerous voltage before touching them.**

CAUTION

- **Failure to use the instrument as directed by the manufacturer may render its protective features void.**
- **Always clean the casing with a dry cloth. Do not clean the internals.**
- **Make sure the vent hole is always unblocked.**

Environmental Conditions

The instrument is designed for indoor use and an area with low condensation. The table below shows the general environmental requirements for the instrument.

Environmental Conditions	Requirements
Operating temperature	0°C ~ 40°C
Operating humidity	20% ~ 80%(non-condensation)
Storage temperature	-10°C ~ 70 °C
Altitude	Operating up to 2,000 meters
Installation category	II
Pollution degree	Pollution degree 2

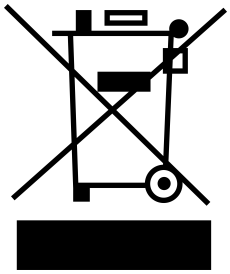
Note

In order to ensure the accuracy of measurement, it is recommended to operate the instrument half an hour after start-up.

Regulation Tag

CE	The CE tag shows that the product complies with the provisions of all relevant European laws (if the year is shown, it indicates that the year when the design is approved).
UK CA	The UKCA tag shows that the product complies with the provisions of all relevant United Kingdom laws (if the year is shown, it indicates that the year when the design is approved).
	This instrument complies with the WEEE directive (2002/96/EC) tag requirements. This attached product tag shows that the electrical/electronic product cannot be discarded in household waste.
10	This symbol indicates that no danger will happen or toxic substances will not leak or cause damage in normal use within the specified period. The service life of the product is 10 years. The product can be used safely within the environmental protection period; otherwise, the product should be put into the recycling system.

Waste Electrical and Electronic Equipment (WEEE) Directive



Waste electrical and electronic equipment (WEEE) directive, 2002/96/EC
The product complies with tag requirements of the WEEE directive (2002/96/EC). This tag indicates that the electronic equipment cannot be disposed of as ordinary household waste. Product Category
According to the equipment classification in Annex I of the WEEE directive, this instrument belongs to the “Monitoring” product.
If you want to return the unnecessary instrument, please contact the nearest sales office of ITECH.

Compliance Information

Complies with the essential requirements of the following applicable European Directives, and carries the CE marking accordingly:

- Electromagnetic Compatibility (EMC) Directive 2014/30/EU
- Low-Voltage Directive (Safety) 2014/35/EU

Conforms with the following product standards:

EMC Standard

IEC 61326-1:2012/ EN 61326-1:2013 [1][2][3]

Reference Standards

CISPR 11:2015+A1:2016 Ed 6.1

IEC 61000-3-2: 2018 RLV

IEC 61000-3-3: 2013+A1:2017

IEC 61000-4-2:2008

IEC 61000-4-3 2006+A1:2007+A2:2010/ EN 61000-4-3 A1:2008+A2:2010

IEC 61000-4-4:2012

IEC 61000-4-5:2014+A1:2017

IEC 61000-4-6:2013+cor1:2015

IEC 61000-4-11:2004+A1:2017

1. The product is intended for use in non-residential/non-domestic environments. Use of the product in residential/domestic environments may cause electromagnetic interference.
2. Connection of the instrument to a test object may produce radiations beyond the specified limit.
3. Use high-performance shielded interface cable to ensure conformity with the EMC standards listed above.

Safety Standard

IEC 61010-1:2010+A1:2016

Content

Quality Certification and Assurance I
Warranty I
Limitation of Warranty I
Safety Symbols II
Safety Precautions II
Environmental Conditions IV
Regulation Tag V
Waste Electrical and Electronic Equipment (WEEE) Directive VI
Compliance Information VII
1 SCPI Introduction 1
1.1 Overview 1
1.2 Command Type of SCPI 1
1.3 Message Type of SCPI 4
1.4 Response Data Type 6
1.5 Command Format 7
1.6 Data Type 10
1.7 Remote Interface Connections 11
2 SYSTem Subsystem 12
SYSTem:POSetup <CPD> 12
SYSTem:POSetup? 13
SYSTem:VERSion? 13
SYSTem:ERRor? 14
SYSTem:CLEar 15
SYSTem:REMote 15
SYSTem:LOCal 16
SYSTem:RWLock 17
SYSTem:BEEPer:IMMediate 18
SYSTem:BEEPer[:STATe] <CPD> 18
SYSTem:BEEPer[:STATe]? 19
SYSTem:DATE <*yyyy*>,<*mm*>,<*dd*> 20
SYSTem:DATE? 21
SYSTem:TIME <*hh*>,<*mm*>,<*ss*> 21
SYSTem:TIME? 22
SYSTem:COMMunicate:SELect <CPD> 23
SYSTem:COMMunicate:SELect? 24
SYSTem:COMMunicate:GPIB:ADDRess <NR1> 25
SYSTem:COMMunicate:GPIB:ADDRess? 25
SYSTem:COMMunicate:SERial:BAUDrate <CPD> 26
SYSTem:COMMunicate:SERial:BAUDrate? 27
SYSTem:COMMunicate:LAN:IP[:CONFiguration] <SPD> 28
SYSTem:COMMunicate:LAN:IP[:CONFiguration]? 28
SYSTem:COMMunicate:LAN:IP[:CONFiguration]:MODE <CPD> 29
SYSTem:COMMunicate:LAN:IP[:CONFiguration]:MODE? 30
SYSTem:COMMunicate:LAN:SMASk <SPD> 31
SYSTem:COMMunicate:LAN:SMASk? 31
SYSTem:COMMunicate:LAN:DGATeway <SPD> 32
SYSTem:COMMunicate:LAN:DGATeway? 33
3 ABORt Subsystem 34
ABORt:ACQuire 34
ABORt:ARB 34
4 INITiate Subsystem 36
INITiate[:IMMediate]:ACQuire 36
INITiate:CONTinuous:ACQuire <Bool> 37
INITiate[:IMMediate]:DLOG 37
INITiate[:IMMediate]:ELOG 38

5 CONFigurable Subsystem 40
IO:SELect <NR1> 40
IO:SELect? 41
IO:DIREction <NRL>, <Bool> 41
IO:DIREction? <NRL> 42
IO:REVErse <NRL>, <Bool> 43
IO:REVErse? <NRL> 44
IO:PWM[:ENABle] <NRL>, <Bool> 45
IO:PWM[:ENABle]? <NRL> 46
IO:PWM:FREQuency <NRL>, <NRf+> 47
IO:PWM:FREQuency? <NRL>[,][MINimum|MAXimum|DEFault] 47
IO:PWM:DUTY <NRL>, <NR1> 48
IO:PWM:DUTY? <NRL> 49
IO:PULSe:WIDTh <NRL>, <NRf+> 50
IO:PULSe:WIDTh? <NRL>[,][MINimum|MAXimum|DEFault] 50
IO:TYPE <NRL>, <CPD> 51
IO:TYPE? <NRL> 52
IO:TRIGger:TYPE <CPD> 54
IO:TRIGger:TYPE? 55
IO:TRIGer:SOURce <CPD> 55
IO:TRIGer:SOURce? 56
IO:OUTPut:LEVel <NRL>, <CPD> 57
IO:OUTPut:LEVel? <NRL> 58
IO:INPut:LEVel? <NRL> 59
6 TRIGger Subsystem 60
TRIGger:ACQuire[:IMMediate] 60
TRIGger:ACQuire:MODE <CPD> 61
TRIGger:ACQuire:STATe? 61
TRIGger:ACQuire:SOURce <CPD> 62
TRIGger:ACQuire:VOLTage:SLOPe <CPD> 63
TRIGger:ACQuire:VOLTage[:LEVel] <NRf+> 64
TRIGger:ACQuire:VOLTage:HYSTeresis:HIGH <NRf+> 65
TRIGger:ACQuire:VOLTage:HYSTeresis:LOW <NRf+> 66
TRIGger:ACQuire:CURRent:SLOPe <CPD> 67
TRIGger:ACQuire:CURRent[:LEVel] <NRf+> 68
TRIGger:ACQuire:CURRent:HYSTeresis:HIGH <NRf+> 69
TRIGger:ACQuire:CURRent:HYSTeresis:LOW <NRf+> 69
TRIGger:ARB[:IMMediate] 70
TRIGger:ARB:SOURce <CPD> 71
7 STATus Subsystem 73
STATus:QUEStionable[:EVENt]? 77
STATus:QUEStionable:ENABle <NR1> 78
STATus:QUEStionable:PTRansition <NR1> 79
STATus:QUEStionable:NTRansition <NR1> 80
STATus:QUEStionable:CONDition? 81
STATus:OPERation[:EVENt]? 82
STATus:OPERation:ENABle <NR1> 83
STATus:OPERation:PTRansition <NR1> 84
STATus:OPERation:NTRansition <NR1> 85
STATus:OPERation:CONDition? 86
STATus:PRESet 87
8 FETCh Subsystem 89
FETCh[:SCALar]:CURRent[:DC]? 89
FETCh[:SCALar]:CURRent[:DC]:HIGH? 90
FETCh[:SCALar]:CURRent[:DC]:LOW? 90
FETCh[:SCALar]:CURRent[:DC]:MAXimum? 91
FETCh[:SCALar]:CURRent[:DC]:MINimum? 92

FETCh[:SCALar]:VOLTage[:DC]? ... 92
FETCh[:SCALar]:VOLTage[:DC]:HIGH? ... 93
FETCh[:SCALar]:VOLTage[:DC]:LOW? ... 94
FETCh[:SCALar]:VOLTage[:DC]:MAXimum? ... 95
FETCh[:SCALar]:VOLTage[:DC]:MINimum? ... 95
FETCh[:SCALar]:POWer[:DC]? ... 96
FETCh[:SCALar]:POWer[:DC]:MAXimum? ... 97
FETCh[:SCALar]:POWer[:DC]:MINimum? ... 98
FETCh:ARRay:CURRent[:DC]? ... 98
FETCh:ARRay:VOLTage[:DC]? ... 99
FETCh:ARRay:POWer[:DC]? ... 100
FETCh[:SCALar]? ... 101
FETCh:ACMeter:EACStage? ... 101
FETCh:ACMeter:EACTotal? ... 102
9 MEASure Subsystem ... 104
MEASure[:SCALar]:CURRent[:DC]? ... 104
MEASure[:SCALar]:CURRent:HIGH? ... 105
MEASure[:SCALar]:CURRent:LOW? ... 105
MEASure[:SCALar]:CURRent:MAXimum? ... 106
MEASure[:SCALar]:CURRent:MINimum? ... 107
MEASure[:SCALar]:VOLTage[:DC]? ... 108
MEASure[:SCALar]:VOLTage:HIGH? ... 108
MEASure[:SCALar]:VOLTage:LOW? ... 109
MEASure[:SCALar]:VOLTage:MAXimum? ... 110
MEASure[:SCALar]:VOLTage:MINimum? ... 111
MEASure[:SCALar]:POWer[:DC]? ... 111
MEASure[:SCALar]:POWer:MAXimum? ... 112
MEASure[:SCALar]:POWer:MINimum? ... 113
MEASure:ARRay:CURRent[:DC]? ... 114
MEASure:ARRay:VOLTage[:DC]? ... 114
MEASure:ARRay:POWer[:DC]? ... 115
MEASure[:SCALar]? ... 116
10 PARallel Subsystem ... 117
PARallel:ROLE <CPD> ... 117
PARallel:NUMBer <NR1> ... 118
11 SENSe Subsystem ... 119
[SOURce:]REMote:SENSe[:STATe] <Bool> ... 119
SENSe:ACQuire:POINts <NRf+> ... 120
SENSe:ACQuire:OFFSet:POINt <NRf+> ... 120
SENSe:ACQuire:TINTerval <NRf+> ... 121
12 SOURce Subsystem ... 123
12.1 [SOURce:]FUNCtion <CPD> ... 123
12.2 [SOURce:]LOOP:SPEed <CPD> ... 124
12.3 [SOURce:]FUNCtion:MODE <CPD> ... 125
12.4 [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <NRf+> ... 126
12.5 [SOURce:]CURRent:SLEW[:BOTH] <NRf+> ... 127
12.6 [SOURce:]CURRent:SLEW:POSitive <NRf+> ... 127
12.7 [SOURce:]CURRent:SLEW:NEGative <NRf+> ... 128
12.8 [SOURce:]CURRent[:OVER]:PROTection:STATe <Bool> ... 129
12.9 [SOURce:]CURRent[:OVER]:PROTection[:LEVel] <NRf+> ... 130
12.10 [SOURce:]CURRent[:OVER]:PROTection:DELay <NRf+> ... 131
12.11 [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf+> ... 132
12.12 [SOURce:]VOLTage:SLEW[:BOTH] <NRf+> ... 132
12.13 [SOURce:]VOLTage:SLEW:POSitive <NRf+> ... 133
12.14 [SOURce:]VOLTage:SLEW:NEGative <NRf+> ... 134
12.15 [SOURce:]VOLTage:ON[:LEVel] <NRf+> ... 135
12.16 [SOURce:]VOLTage:UNDer:PROTection:STATe <Bool> ... 136

12.17 [SOURce:]VOLTage:UNDer:PROTection[:LEVel] <NRf+> 137
12.18 [SOURce:]VOLTage:UNDer:PROTection:DELay <NRf+> 137
12.19 [SOURce:]VOLTage:UNDer:PROTection:WARM <NRf+>..... 138
12.20 [SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude] <NRf+> 139
12.21 [SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude] <NRf+> 140
12.22 [SOURce:]POWer:PROTection:STATe <bool> 141
12.23 [SOURce:]POWer:PROTection[:LEVel] <NRf+> 142
12.24 [SOURce:]POWer:PROTection:DELay <NRf+> 142
12.25 [SOURce:]POWer:SLEW[:BOTH] <NRf+>..... 143
12.26 [SOURce:]POWer:SLEW:POSitive <NRf+> 144
12.27 [SOURce:]POWer:SLEW:NEGative <NRf+> 145
12.28 [SOURce:]VOLTage[:ON]:LATCh[:STATe] <Bool> 146
12.29 [SOURce:]VOLTage:ON:HYSTeresis[:LEVel] <NRf+>..... 146
12.30 [SOURce:]EXTern[:STATe] <Bool> 147
12.31 [SOURce:]ACMeter:EACStage:CLEar..... 148
13 INPut Subsystem..... 150
13.1 INPut[:STATe] <CPD>..... 150
13.2 INPut:DELay:FALL <NRf+>..... 151
13.3 INPut:DELay:RISE <NRf+>..... 151
13.4 INPut:SHORt[:STATe] <CPD> 152
13.5 PROTection:WDOG[:STATe] <CPD> 153
13.6 PROTection:WDOG:DELay <NRf+>..... 154
14 BATTery Subsystem 155
14.1 BATTery:SHUT:CAPacity <NRf+> 155
14.2 BATTery:SHUT:TIME <NRf+>..... 156
14.3 BATTery:SHUT:VOLTage <NRf+>..... 156
15 ARB Subsystem 158
15.1 ARB:UDEFined:COUNt <NR1>..... 158
15.2 ARB:FUNCtion <CPD>..... 159
15.3 ARB:UDEFined:LEVel <NR1>,<NRf+>..... 159
15.4 ARB:UDEFined:DWELl <NR1>,<NRf+> 160
15.5 ARB:UDEFined:SLEW <NR1>,<NRf+> 161
15.6 ARB:COUNt <NRf+> 162
15.7 ARB:SAVE <NR1>..... 163
15.8 ARB:RECall <NR1>..... 163
16 IEEE-488 Common Commands..... 165
*CLS..... 165
*ESE <NR1>..... 166
*ESE? 167
*ESR? 167
*IDN? 168
*OPC..... 169
*OPC?..... 170
*RST 171
*SRE <NR1>..... 173
*SRE? 174
*STB?..... 175
*TRG 176
*SAV <NR1>..... 176
*RCL <NR1>..... 177
*TST?..... 178
*WAI..... 179
*PSC <Bool>..... 180
*PSC? 180
17 Error Messages 182

1 SCPI Introduction

This chapter describes in detail the command types, parameter types and other related information about SCPI.

♦ Overview
♦ Command Type of SCPI
♦ Message Type of SCPI
♦ Response Data Type
♦ Command Format
♦ Data Type
♦ Remote Interface Connections

1.1 Overview

SCPI is short for Standard Commands for Programmable Instruments which defines a communication method of bus controller and instrument. It is based on ASCII and supply for testing and measuring instruments. SCPI command is based on hierarchical architecture which also known as tree system. In this system, Relevant Command is returned to a common node or root, so that a subsystem is formed. A part of OUTPut subsystem is listed below:

OUTPut:

- **SYNC {OFF|0|ON|1}**
- **SYNC:**
  - **MODE {NORMal|CARRier}**
  - **POLarity {NORMal|INVerted}**

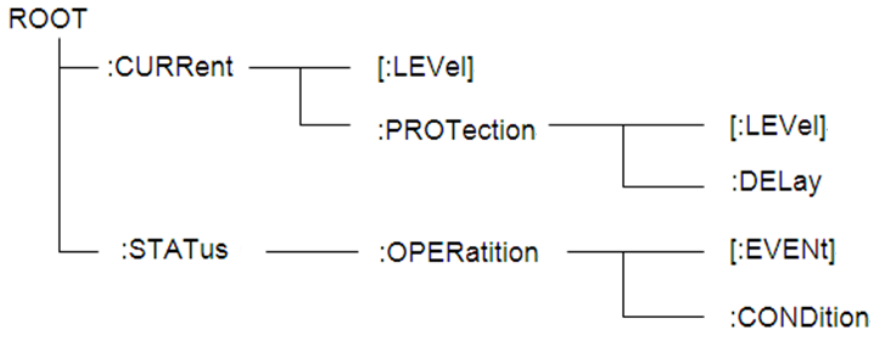
OUTPut is the root class keyword, SYNC is the second keyword, MODE and POLarity are the third keyword. Colon(:) is used for separating the command keyword and the next level keyword.

1.2 Command Type of SCPI

SCPI has two types of commands, common and subsystem.

- Common commands generally are not related to specific operation but to controlling overall instrument functions, such as reset, status, and synchronization. All common commands consist of a three-letter mnemonic preceded by an asterisk: ***RST *IDN? *SRE 8**.

- Subsystem commands perform specific instrument functions. They are organized into an inverted tree structure with the "root" at the top. The following figure shows a portion of a subsystem command tree, from which you access the commands located along the various paths.



Multiple commands in a message

Multiple SCPI commands can be combined and sent as a single message with one message terminator. There are two important considerations when sending several commands within a single message:

- Use a semicolon to separate commands within a message.
- Head paths influence how the instrument interprets commands.

We consider the head path as a string which will be inserted in front of every command of a message. As for the first command of a message, the head path is a null string; for each subsequent command, the head path is a string which is defined to form the current command until and including the head of the last colon separator. A message with two combined commands:

CURR:LEV 3;PROT:STAT OFF

The example indicates the effect of semicolon and explains the concept of head path. Since the head path is defined to be "CURR" after "curr: lev 3", the head of the second command, "curr", is deleted and the instrument explains the second command as:

CURR:PROT:STAT OFF

If "curr" is explicitly included in the second command, it is semantically wrong. Since combining it with the head path will become "CURR:CURR:PROT:STAT OFF", resulting in wrong command.

Movement in the subsystem

In order to combine commands from different subsystems, you need to be able to reset the header path to a null string within a message. You do this by beginning the command with a colon (:), which discards any previous header path.

For example, you could clear the output protection and check the status of the Operation Condition register in one message by using a root specifier as follows:

PROTection:CLEAr;:STATus:OPERation:CONDition?

The following message shows how to combine commands from different subsystems as well as within the same subsystem:

POWer:LEVel 200;PROTection 28; :CURRent:LEVel 3;PROTection: STATeON

Note the use of the optional header LEVel to maintain the correct path within the voltage and current subsystems, and the use of the root specifier to move between subsystems.

Including common commands

You can combine common commands with subsystem commands in the same message. Treat the common command as a message unit by separating it with a semicolon (the message unit separator). Common commands do not affect the header path; you may insert them anywhere in the message.

VOLTage:TRIGgered 17.5;:INITialize;*TRG

OUTPut OFF;*RCL 2;OUTPut ON

Case sensitivity

Common commands and SCPI commands are not case sensitive. You can use upper or lower, for example:

***RST = *rst**

:DATA? = :data?

:SYSTem:PRESet = :system:preset

Long-form and short-form versions

A SCPI command word can be sent in its long-form or short-form version. However, the short-form version is indicated by upper case characters. Examples:

:SYSTem:PRESet long-form

:SYST:PRES short form

:SYSTem:PRES long-form and short-form combination

Note that each command word must be in long-form or short-form, and not something in between.

For example, **:SYSTe:PRESe** is illegal and will generate an error. The command will not be executed.

Query

Observe the following precautions with queries:

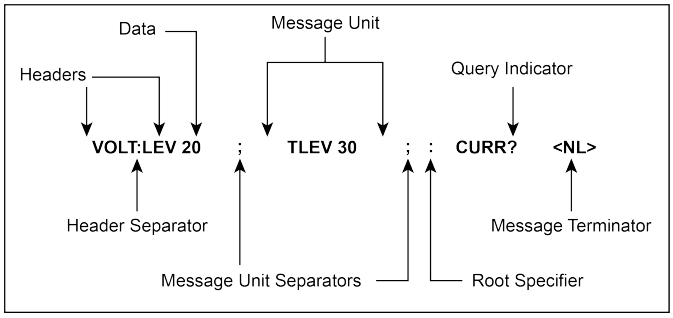
- Set up the proper number of variables for the returned data. For example, if you are reading back a measurement array, you must dimension the array according to the number of measurements that you have placed in the measurement buffer.
- Read back all the results of a query before sending another command to the instrument. Otherwise a Query Interrupted error will occur and the unreturned data will be lost.

1.3 Message Type of SCPI

There are two types of SCPI messages, program and response.

- Program message: A program message consists of one or more properly formatted SCPI commands sent from the controller to the instrument. The message, which may be sent at any time, requests the instrument to perform some action.
- Response message: A response message consists of data in a specific SCPI format sent from the instrument to the controller. The instrument sends the message only when commanded by a program message called a "query."

The next figure illustrates SCPI message structure:



The message unit

The simplest SCPI command is a single message unit consisting of a command header (or keyword) followed by a message terminator. The message unit may include a parameter after the header. The parameter can be numeric or a string.

ABORt<NL>

VOLTage 20<NL>

Headers

Headers, also referred to as keywords, are instructions recognized by the instrument. Headers may be either in the long form or the short form. In the long form, the header is completely spelled out, such as VOLTAGE, STATUS and DELAY. In the short form, the header has only the first three or four letters, such as VOLT, STAT and DEL.

Query indicator

Following a header with a question mark turns it into a query (**VOLTage?** , **VOLTage:PROTection?**). If a query contains a parameter, place the query indicator at the end of the last header(**VOLTage:PROTection?MAX**).

Message unit separator

When two or more message units are combined into a compound message, separate the units with a semicolon (**STATus:OPERation?;QUEStionable?**).

Root specifier

When it precedes the first header of a message unit, the colon becomes the root specifier. It tells the command parser that this is the root or the top node of the command tree.

Message terminator

A terminator informs SCPI that it has reached the end of a message. Three permitted message terminators are:

- newline (<NL>), decimal 10 or hexadecimal 0X0A in ASCII.
- end or identify (<END>)
- both of the above (<NL><END>).

In the examples of this guide, there is an assumed message terminator at the end of each message.

Command execution rules

- Commands execute in the order that they are presented in the program message.
- An invalid command generates an error and, of course, is not executed.

- Valid commands that precede an invalid command in a multiple command program message are executed.
- Valid commands that follow an invalid command in a multiple command program message are ignored.

1.4 Response Data Type

Character strings returned by query statements may take either of the following forms, depending on the length of the returned string:

- **<CRD>**: character response data. Permits the return of character strings.
- **<AARD>**: arbitrary ASCII response data. Permits the return of undelimited 7-bit ASCII. This data type has an implied message terminator.
- **<SRD>**: string response data. Returns string parameters enclosed in double quotes.
- **<Block>**: arbitrary block data.

Response messages

A response message is the message sent by the instrument to the computer in response to a query command.

Sending a response message

After sending a query command, the response message is placed in the Output Queue. When the instrument is then addressed to talk, the response message is sent from the Output Queue to the computer

Multiple response messages

If you send more than one query command in the same program message, the multiple response messages for all the queries is sent to the computer when the instrument is addressed to talk. The responses are sent in the order that the query commands were sent and are separated by semicolons (;). Items within the same query are separated by commas (,). The following example shows the response message for a program message that contains four single item query commands:

0; 1; 1; 0

Response message terminator (RMT)

Each response is terminated with an LF (line feed) and EOI (end or identify). The following example shows how a multiple response message is terminated:

0; 1; 1; 0; <RMT>

Message exchange protocol

Two rules summarize the message exchange protocol:

- **Rule 1**: You must always tell the instrument what to send to the computer.

  The following two steps must always be performed to send information from the instrument other computer:

  1. Send the appropriate query command(s) in a program message.
  2. Address the instrument to talk.

- **Rule 2**: The complete response message must be received by the computer before another program message can be sent to the instrument.

1.5 Command Format

Formats for command display are as follows:

[SOURce[1|2]:]VOLTage:UNIT {VPP|VRMS|DBM}

[SOURce[1|2]:]FREQuency:CENTer {<frequency>|MINimum|MAXimum|DEFault}

Based on the command syntax, most commands (and certain Parameter) are expressed in both upper and lower cases. Upper case refers to abbreviation of commands. Shorter program line may send commands in abbreviated format. Long-format commands may be sent to ensure better program readability.

For example, both formats of VOLT and VOLTAGE are acceptable in the above syntax statements. Upper or lower case may be used. Therefore, formats of VOLTAGE, volt and Volt are all acceptable. Other formats (such as VOL and VOLTAG) are invalid and will cause errors.

- Parameter options with given command strings are included in the brace ({ }). The brace is not sent along with command strings.
- Vertical stripes (|) separate several parameter options with given command strings. For example, {VPP|VRMS|DBM} indicates that you may assign "APP", "VRMS" or "DBM" in the above commands. Vertical stripes are not sent along with command strings.

- Angle brackets (< >) in the second example indicates that a value must be assigned to the parameter in the brace. For example, the parameter in the angle bracket is <frequency> in the above syntax statements. Angle brackets are not sent along with command strings. You must assign a value (such as "FREQ:CENT 1000") to the parameter, unless you select other options displayed in the syntax (such as "FREQ:CENT MIN").
- Some syntax elements (such as nodes and Parameter) are included in square brackets ([ ]). It indicates that these elements can be selected and omitted. Angle brackets are not sent along with command strings. If no value is assigned to the optional Parameter, the instrument will select a default value. In the above examples, "SOURce[1|2]" indicates that you may refer to source channel 1 by "SOURce" or "SOURce1" or "SOUR1" or "SOUR". In addition, since the whole SOURce node is optional (in the square bracket), you can refer to the channel 1 by omitting the whole SOURce node. It is because the channel 1 is the default channel for SOURce language node. On the other hand, if you want to refer to channel 2, "SOURce2" or "SOUR2" must be used in the program line.

Colon (:)

It is used to separate key words of a command with the key words in next level. As shown below:

APPL:SIN 455E3,1.15,0.0

In this example, APPLy command assigns a sine wave with frequency of 455 KHz, amplitude of 1.15 V and DC offset of 0.0 V.

Semicolon (;)

It is used to separate several commands in the same subsystem and can also minimize typing. For example, to send the following command string:

TRIG:SOUR EXT; COUNT 10

has the same effect as sending the following two commands:

TRIG:SOUR EXT

TRIG:COUNT 10

Question mark (?)

You can insert question marks into a command to query current values of most Parameter. For example, the following commands will trigger to set the count as 10:

TRIG:COUN 10

Then, you may query count value by sending the following command:

TRIG:COUN?

You may also query the allowable minimum or maximum count as follows:

TRIG:COUN?MIN

TRIG:COUN?MAX

Comma (,)

If a command requires several Parameter, then a comma must be used to separate adjacent Parameter.

Space

You must use blank characters, [TAB] or [Space] to separate Parameter with key words of commands.

Common commands (*)

The IEEE-488.2 standard defines a set of common commands that perform functions such as reset, self-test, and status operations. Common commands always start with a asterisk (*) and occupy 3 character sizes, including one or more Parameter. Key words of a command and the first parameter are separated by a space. Semicolon (;) can separate several commands as follows:

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

Command terminator

Command strings sent to the instrument must end with a <Newline> (<NL>) character. IEEE-488 EOI (End or Identify) information can be used as <NL> character to replace termination command string of <NL> character. It is acceptable to place one <NL> after a <Enter>. Termination of command string always resets current SCPI command path to root level.

Note

As for every SCPI message with one query sent to the instrument, the instrument will use a <NL> or newline sign (EOI) to terminate response of return. For example, if "DISP:TEXT?" is sent, <NL> will be placed after the returned data string to terminate response. If an SCPI message includes several queries separated by semicolon (such as "DISP?;DISP:TEXT?"), <NL> will terminate response returned after response to the last query. In all cases, the program must read <NL> in response before another command is sent to the instrument, otherwise errors will be caused.

1.6 Data Type

SCPI language defines several data types used for program message and response messages.

- Numerical parameter

  Commands requiring numerical parameter support the notations of all common decimal notations, including optional signs, decimal points, scientific notation, etc. Special values of numerical Parameter are also acceptable, such as MIN, MAX and DEF. In addition, suffixes for engineering units can also be sent together with numerical Parameter (including M, k, m or u). If the command accepts only some specific values, the instrument will automatically round the input Parameter to acceptable values. The following commands require numerical Parameter of frequency value:

  [SOURce[1|2]:]FREQuency:CENTer {<Frequency>|MINimum|MAXimum}

  – **<NR1>**: represents an integer value, such as 273;

  – **<NR2>**: represents a real number in floating-point format, such as .273;

  – **<NR3>**: represents a real number in scientific notation, such as 2.73E+2;

  – **<Nrf>**: The extensible form includes <NR1>, <NR2> and <NR3>;

  – **<Nrf+>**: The extensible decimal form includes <Nrf>, MIN, MAX and DEF. MIN and MAX are the minimum and maximum finite number. Within the range of the parameter definition, DEF is the default of the parameter.

- Discrete parameter

  Discrete Parameter are used for settings with limited number of programming values (such as IMMediate, EXTernal or BUS). They can use short and long format like key words of commands. They may be expressed in both upper and lower case. The query response always returns uppercase parameter in short format. The following commands require discrete parameter in voltage unit:

  [SOURce[1|2]:]VOLTage:UNIT {VPP|VRMS|DBM}

- Boolean parameter

  Boolean parameter refer to true or false binary conditions. In case of false conditions, the instrument will accept "OFF" or "0". In case of true conditions, the instrument will accept "ON" or "1". In query of Boolean settings, the instrument will always return "0" or "1". Boolean parameter are required by the following commands:

DISPlay {OFF|0|ON|1}

- ASCII string parameter

  String parameter may actually include all ASCII character sets. Character strings must start and end with paired quotation marks; and single quotation marks or double quotation marks are both allowed. Quotation mark separators may also act as one part of a string, they can be typed twice without any character added between them. String parameter is used in the following command:

  DISPlay:TEXT <quoted string>

  For example, the following commands display message of "WAITING..." (without quotation marks) on the front panel of the instrument.

  DISP:TEXT "WAITING..."

  Single quotation marks may also be used to display the same message.

  DISP:TEXT 'WAITING...'

  – **<SPD>**: string program data. String parameters enclosed in single or double quotes.

  – **<CPD>**: character program data.

1.7 Remote Interface Connections

Please refer to user manual for detailed introductions of the remote interface connections.

Note

If the user want to change the settings of the instrument, for instance, the input setting value, the command SYST:REM must be sent to the instrument after finishing the connection between the instrument and PC.

2 SYSTem Subsystem

SYSTem:POSetup <CPD>

This command is used to set some parameter settings or working status when the instrument is powered on.

- RST: The parameter setting or status when the instrument is powered on is equivalent to executing the ***RST** command.

  For details, see *RST.

- LAST: The instrument will remain the same parameter settings and output status as last time you powered off the instrument.
- LAST_OFF: The instrument will remain the same settings as last time you powered off the instrument, but the output status is Off.

Group

System

Syntax

SYSTem:POSetup <CPD>

Arguments

<CPD>

RST|LAST|LAST_OFF

Default Value

RST

Returns

None

Example

SYST:POS SAV0

Also see

SYSTem:POSetup?

SYSTem:POSetup?

This command is used to query some parameter settings or working status when the instrument is powered on.

Group

System

Syntax

SYSTem:POSetup?

Arguments

None

Default Value

None

Returns

<CPD>

RST|LAST|LAST_OFF

Example

SYST:POS?

Also see

None

SYSTem:VERSion?

This command is used to query the version number of the used SCPI command. To check the serial number and version of the instrument, please use the ***IDN?** command.

Group

System

Syntax

SYST:VERS?

Arguments

None

Default Value

None

Returns

<SRD>

Example

IT39XXSrc-v1.3.7.xx

Also see

None

SYSTem:ERRor?

This command is used to query the error information of the instrument. When the ERROR indicator on the front panel is lit, it indicates that one or more errors have occurred in the hardware or command syntax of the detected instrument. Up to 20 sets of error messages can be stored in the error queue. This command is sent once to read an error message from the error queue.

- The front-panel ERROR annunciator turns on when one or more errors are currently stored in the error queue. Error retrieval is first-in-first-out (FIFO), and errors are cleared as you read them. When you have read all errors from the error queue, the ERROR annunciator turns off.
- If more than 20 errors have occurred, the last error stored in the queue (the most recent error) is replaced with -350 ("Error queue overflow"). No additional errors are stored until you remove errors from the queue. If no errors have occurred when you read the error queue, the instrument responds with +0 ("No error").

- If the instrument is turned off or the *CLS (clear status) command is sent, the error message in the error queue will be cleared. The *RST command will not clear the error message in the error queue.

SYSTem:CLEar

This command is used to clear the error queue.

Group

System

Syntax

SYST:CLE

Arguments

None

Default Value

None

Returns

None

Example

SYST:CLE

Also see

None

SYSTem:REMote

This command is used to set the instrument to the remote control mode via the communication interface. Except the Local key on the front panel, other keys are locked and cannot be used.

Group

System

Syntax

SYST:REM

Arguments

None

Default Value

None

Returns

None

Example

SYST:REM

Also see

None

SYSTem:LOCal

This command is used to set the instrument to local mode, i.e. panel control mode. All keys on the front panel will be available after executing this command.

Group

System

Syntax

SYST:LOC

Arguments

None

Default Value

None

Returns

None

Example

SYST:LOC

Also see

None

SYSTem:RWLock

This command is used to set the device to the remote state and lock the local keys.

Group

System

Syntax

SYST:RWL

Arguments

None

Default Value

None

Returns

None

Example

SYST:RWL

Also see

None

SYSTem:BEEPer:IMMediate

This command is used to set the buzzer to make a beep.

Group

System

Syntax

SYST:BEEP:IMM

Arguments

None

Default Value

None

Returns

None

Example

SYST:BEEP:IMM

Also see

None

SYSTem:BEEPer[:STATe] <CPD>

This command is used to set the buzzer enable or disable.

Group

System

Syntax

SYSTem:BEEPer[:STATe] <CPD>

Arguments

OFF|ON

Default Value

ON

Returns

None

Example

SYST:BEEP OFF

Also see

SYSTem:BEEPer[:STATe]?

SYSTem:BEEPer[:STATe]?

This command is used to query the status of the buzzer: enable or disable.

Group

System

Syntax

SYSTem:BEEPer[:STATe]?

Arguments

None

Default Value

None

Returns

<CRD>

Example

SYST:BEEP?

Also see

None

SYSTem:DATE <*yyyy*>,<*mm*>,<*dd*>

This command is used to set the date of the system clock. Specify the number of years (2000 to 2099), the number of months (1 to 12), and the number of days (1 to 31).

Group

System

Syntax

SYSTem:DATE <yyyy>,<mm>,<dd>

Arguments

NR1

Default Value

None

Returns

None

Example

SYST:DATE 2017,06,30

Also see

SYSTem:DATE?

SYSTem:DATE?

This command is used to query the date of the system clock.

Group

System

Syntax

SYSTem:DATE?

Arguments

None

Default Value

None

Returns

<SRD>

<yyyy>,<mm>,<dd>

Example

SYST:DATE?

Also see

None

SYSTem:TIME <*hh*>,<*mm*>,<*ss*>

This command is used to set the time of the system clock. Specify the number of hours (0 to 23), minutes (0 to 59), and seconds (0 to 59).

Note

The real-time clock does not adjust itself to accommodate time zone changes or daylight saving time.

Group

System

Syntax

SYSTem:TIME <hh>,<mm>,<ss>

Arguments

SPD

Default Value

12,30,01

Returns

None

Example

Set the clock to 8:30 PM: **SYST:TIME 20,30,0**

Also see

SYSTem:TIME?

SYSTem:TIME?

This command is used to query the time of the system clock.

Group

System

Syntax

SYSTem:TIME?

Arguments

None

Default Value

None

Returns

<SRD>

<hh>,<mm>,<ss>

Example

SYST:TIME?

Also see

None

SYSTem:COMMunicate:SELect <CPD>

This command is used to set the communication method. This series instrument comes standard with four communication interfaces: USB, LAN, VCP and CAN, and supports two optional communication interfaces: GPIB, RS-232. And the RS232 and GPIB options can be selected only after the communication board corresponding to RS232 and GPIB is successfully inserted into the corresponding position on the rear panel of the instrument.

Group

System

Syntax

SYSTem:COMMunicate:SELect <CPD>

Arguments

RS232|USB|GPIB|LAN|CAN|VCP

Default Value

VCP

Returns

None

Example

SYST:COMM:SEL LAN

Also see

SYSTem:COMMunicate:SELect?

SYSTem:COMMunicate:SELect?

This command is used to query the currently selected communication interface.

Group

System

Syntax

SYSTem:COMMunicate:SELect?

Arguments

None

Default Value

None

Returns

<CRD>

Example

SYST:COMM:SEL?

Also see

None

SYSTem:COMMunicate:GPIB:ADDRess <NR1>

This command is used to set the GPIB communication address.

Group

System

Syntax

SYSTem:COMMunicate:GPIB:ADDRess <NR1>

Arguments

<NR1>

Setting range: 1 to 30

Default Value

1

Returns

None

Example

SYST:COMM:GPIB:ADDR 2

Also see

SYSTem:COMMunicate:GPIB:ADDRess?

SYSTem:COMMunicate:GPIB:ADDRess?

This command is used to query the GPIB communication address.

Group

System

Syntax

SYSTem:COMMunicate:GPIB:ADDRess?

Arguments

None

Default Value

None

Returns

<NR1>

Example

SYST:COMM:GPIB:ADDR?

Also see

None

SYSTem:COMMunicate:SERial:BAUDrate <CPD>

This command is used to set the baud rate of RS232 interface.

Group

System

Syntax

SYSTem:COMMunicate:SERial:BAUDrate <CPD>

Arguments

<CPD>

115200|57600|38400|19200|9600|4800

Default Value

9600

Returns

None

Example

SYST:COMM:SER:BAUD 4800

Also see

SYSTem:COMMunicate:SERial:BAUDrate?

SYSTem:COMMunicate:SERial:BAUDrate?

This command is used to query the currently RS232 baud rate.

Group

System

Syntax

SYSTem:COMMunicate:SERial:BAUDrate?

Arguments

None

Default Value

None

Returns

<CRD>

115200|57600|38400|19200|9600|4800

Example

SYST:COMM:SER:BAUD?

Also see

None

SYSTem:COMMunicate:LAN:IP[:CONFiguration] <SPD>

This command is used to set the IP address of the instrument.

Group

System

Syntax

SYSTem:COMMunicate:LAN:IP[:CONFiguration] <SPD>

Arguments

<SPD>

Default Value

"192.168.0.10"

Returns

None

Example

SYST:COMM:LAN:IP "192.168.0.11"

Also see

SYSTem:COMMunicate:LAN:IP[:CONFiguration]?

SYSTem:COMMunicate:LAN:IP[:CONFiguration]?

This command is used to query the IP address of the instrument.

Group

System

Syntax

SYSTem:COMMunicate:LAN:IP[:CONFiguration]?

Arguments

None

Default Value

None

Returns

<SRD>

Example

SYST:COMM:LAN:IP?

Also see

None

SYSTem:COMMunicate:LAN:IP[:CONFiguration]:MODE <CPD>

This command is used to set the IP mode of the LAN port.

- Manual: The user manually sets the IP related parameters.
- AUTO: The system automatically configures IP related parameters.

Group

System

Syntax

SYSTem:COMMunicate:LAN:IP[:CONFiguration]:MODE <CPD>

Arguments

<CPD>

AUTO|MANual

Default Value

MANual

Returns

None

Example

SYST:COMM:LAN:IP:MODE AUTO

Also see

SYSTem:COMMunicate:LAN:IP[:CONFiguration]:MODE?

SYSTem:COMMunicate:LAN:IP[:CONFiguration]:MODE?

This command is used to query the IP mode of the LAN port.

Group

System

Syntax

SYSTem:COMMunicate:LAN:IP[:CONFiguration]:MODE?

Arguments

None

Default Value

None

Returns

<CRD>

AUTO|MANual

Example

SYST:COMM:LAN:IP:MODE?

Also see

None

SYSTem:COMMunicate:LAN:SMASk <SPD>

This command is used to set the subnet mask.

Group

System

Syntax

SYSTem:COMMunicate:LAN:SMASk <SPD>

Arguments

<SPD>

Default Value

"255.255.255.0"

Returns

None

Example

SYST:COMM:LAN:SMAS "255.255.255.1"

Also see

SYSTem:COMMunicate:LAN:SMASk?

SYSTem:COMMunicate:LAN:SMASk?

This command is used to query the subnet mask of the LAN communication.

Group

System

Syntax

SYSTem:COMMunicate:LAN:SMASk?

Arguments

None

Default Value

None

Returns

<SRD>

Example

SYST:COMM:LAN:SMAS?

Also see

None

SYSTem:COMMunicate:LAN:DGATeway <SPD>

This command is used to set the gateway address of the LAN communication.

Group

System

Syntax

SYSTem:COMMunicate:LAN:DGATeway <SPD>

Arguments

<SPD>

Default Value

"192.168.200.1"

Returns

None

Example

SYST:COMM:LAN:DGAT "192.168.0.1"

Also see

SYST:COMM:LAN:DGAT?

SYSTem:COMMunicate:LAN:DGATeway?

This command is used to query the gateway address of the LAN communication.

Group

System

Syntax

SYSTem:COMMunicate:LAN:DGATeway?

Arguments

None

Default Value

None

Returns

<SRD>

Example

SYST:COMM:LAN:DGAT?

Also see

None

3 ABORt Subsystem

Abort commands cancel any triggered actions and returns the trigger system back to the Idle state.

ABORt:ACQuire

Cancel any triggered measurement, i.e. discard the present measurement.

Group

ABORt

Syntax

ABORt:ACQuire

Arguments

None

Default Value

None

Returns

None

Example

ABOR:ACQ

Also see

None

ABORt:ARB

Cancel the execution of the ARB subsystem instructions.

Group

ABORt

Syntax

ABORt:ARB

Arguments

None

Default Value

None

Returns

None

Example

ABOR:ARB

Also see

None

4 INITiate Subsystem

Initiate commands initialize the trigger system. This moves the trigger system from the "idle" state to the "wait-for- trigger" state; which enables the instrument to receive triggers.

- It takes a few milliseconds for the instrument to be ready to receive a trigger signal after receiving the INITiate command.
- If a trigger occurs before the trigger system is ready for it, the trigger will be ignored.
- Use ABORt commands to return the instrument to Idle.

INITiate[:IMMediate]:ACQuire

Initiates the measurement trigger system.

Group

INITiate

Syntax

INITiate[:IMMediate]:ACQuire

Arguments

None

Default Value

None

Returns

None

Example

INIT:ACQ

Also see

None

INITiate:CONTinuous:ACQuire <Bool>

This command is used to enable or disable the Meter continuous trigger function.

Group

INITiate

Syntax

INITiate:CONTinuous:ACQuire <Bool>

Arguments

<Bool>

0|OFF|1|ON

Default Value

1|ON

Returns

None

Example

INIT:CONT:ACQ 0

Also see

INITiate:CONTinuous:ACQuire?

INITiate[:IMMediate]:DLOG

This command is used to enable the data logging function on the front panel of the instrument. Before starting DLOG, you need to plug in a USB flash drive.

After the last dlog ends, you need to wait 5 seconds before starting the next recording.

Group

INITiate

Syntax

INITiate[:IMMediate]:DLOG

Arguments

None

Default Value

None

Returns

None

Example

INIT:DLOG

Also see

None

INITiate[:IMMediate]:ELOG

This command is used to enable the ELOG function switch. The ELOG function refers to reading test data stored in the instrument's buffer by instructions.

Group

INITiate

Syntax

INITiate[:IMMediate]:ELOG

Arguments

None

Default Value

None

Returns

None

Example

INIT:ELOG

Also see

None

5 CONFigurable Subsystem

Applies to digital I/O function.

IO:SELect <NR1>

This command is used to set the pin number of the digital I/O interface. Pins 1 to 7 correspond to number 0 to 6.

Group

CONFigurable

Syntax

IO:SELect <NR1>

Arguments

<NR1>

The setting range is from 0 to 6.

Default Value

0

Returns

None

Example

Select pin 3: **IO:SEL 2**

Also see

IO:SELect?

IO:SELect?

This command is used to query the pin number of the digital I/O interface. Pins 1 to 7 correspond to number 0 to 6.

Group

CONFigurable

Syntax

IO:SELect?

Arguments

None

Default Value

None

Returns

<NR1>

The range is from 0 to 6.

Example

IO:SEL?

Also see

None

IO:DIREction <NRL>, <Bool>

This command is used to set the direction of the digital signal of the specified pin:

- <NRL>: Used to specify the pin number. Pins 1 to 7 correspond to number 0 to 6.
- <Bool>: Used to specify the direction of the digital I/O signal.
  - 0|OUT: The digital I/O signal is sent out from the pin of this instrument.

– 1|IN: The digital I/O signal is sent from the external device to the pin of the instrument.

Group

CONFigurable

Syntax

IO:DIREction <NRL>, <Bool>

Arguments

<NRL>, <Bool>

0 to 6, 0|OUT|1|IN

Default Value

0, 0|OUT

Returns

None

Example

Set the pin 1 as the signal input: **IO:DIRE 0, 1**

Also see

IO:DIREction? <NRL>

IO:DIREction? <NRL>

This command is used to query the direction of the digital signal of the specified pin.

Group

CONFigurable

Syntax

IO:DIREction? <NRL>

Arguments

<NRL>

0 to 6

Default Value

None

Returns

<Bool>

0|OUT|1|IN

Example

IO:DIRE? 0

Also see

None

IO:REVErse <NRL>, <Bool>

This command is used to control whether the digital signal of the specified pin is inverted:

- <NRL>: Used to specify the pin number. Pins 1 to 7 correspond to number 0 to 6.
- <Bool>: Used to control whether the digital signal of the specified pin is inverted.
  - 0|OFF: No
  - 1|ON: Yes

Group

CONFigurable

Syntax

IO:REVErse <NRL>, <Bool>

Arguments

<NRL>, <Bool>

0 to 6, 0|OFF|1|ON

Default Value

0, 0|OFF

Returns

None

Example

Reverse the signal of the pin 1: **IO:REVE 0, 1**

Also see

IO:REVErse? <NRL>

IO:REVErse? <NRL>

This command is used to query whether the digital signal of the specified pin is inverted.

Group

CONFigurable

Syntax

IO:REVErse? <NRL>

Arguments

<NRL>

0 to 6

Default Value

None

Returns

<Bool>

0|OFF|1|ON

Example

IO:REVE? 1

Also see

None

IO:PWM[:ENABle] <NRL>, <Bool>

This command is used to control whether the PWM function of the specified pin is turned on:

- <NRL>: Used to specify the pin number. Pins 1 to 7 correspond to number 0 to 6.
- <Bool>: Used to control whether the PWM function of the specified pin is turned on.
  - 0|OFF: No
  - 1|ON: Yes

Group

CONFigurable

Syntax

IO:PWM[:ENABle] <NRL>, <Bool>

Arguments

<NRL>, <Bool>

0 to 6, 0|OFF|1|ON

Default Value

0, 0|OFF

Returns

None

Example

Turn on the PWM function of the pin 1: **IO:PWM 0, 1**

Also see

IO:PWM[:ENABle]? <NRL>

IO:PWM[:ENABle]? <NRL>

This command is used to query the PWM function switch status of the specified pin.

Group

CONFigurable

Syntax

IO:PWM[:ENABle]? <NRL>

Arguments

<NRL>

0 to 6

Default Value

None

Returns

<Bool>

0|OFF|1|ON

Example

IO:PWM? 1

Also see

None

IO:PWM:FREQuency <NRL>, <NRf+>

This command is used to set the PWM frequency of the specified pin.

Group

CONFigurable

Syntax

IO:PWM:FREQuency <NRL>, <NRf+>

Arguments

<NRL>, <NRf+>

0 to 6, MINimum|MAXimum|DEFault|<value>

Value range: 2 to 21KHz

Default Value

0, 0

Returns

None

Example

Set the PWM frequency of the pin 1 to 100Hz: **IO:PWM:FREQ 0, 100**

Also see

IO:PWM:FREQuency? <NRL>

IO:PWM:FREQuency? <NRL>[,][MINimum|MAXimum|DEFault]

This command is used to query the PWM frequency of the specified pin.

Group

CONFigurable

Syntax

IO:PWM:FREQuency? <NRL>[,][MINimum|MAXimum|DEFault]

Arguments

<NRL>[,][MINimum|MAXimum|DEFault]

Default Value

None

Returns

<NRf+>

Example

IO:PWM:FREQ? 1

Also see

None

IO:PWM:DUTY <NRL>, <NR1>

This command is used to set the PWM duty cycle of the specified pin.

Group

CONFigurable

Syntax

IO:PWM:DUTY <NRL>, <NR1>

Arguments

<NRL>, <NR1>

0 to 6, 1 to 100

Default Value

0, 0

Returns

None

Example

Set the PWM duty cycle of the pin 1 to 10%: **IO:PWM:DUTY 0, 10**

Also see

IO:PWM:DUTY? <NRL>

IO:PWM:DUTY? <NRL>

This command is used to query the PWM duty cycle of the specified pin.

Group

CONFigurable

Syntax

IO:PWM:DUTY? <NRL>

Arguments

<NRL>

0 to 6

Default Value

None

Returns

<NR1>

Example

IO:PWM:DUTY? 1

Also see

None

IO:PULSe:WIDTh <NRL>, <NRf+>

This command is used to set the pulse width of the specified pin.

Group

CONFigurable

Syntax

IO:PULSe:WIDTh <NRL>, <NRf+>

Arguments

<NRL>, <NRf+>

0 to 6, MINimum|MAXimum|DEFault|<value>

Default Value

0, 0

Returns

None

Example

Set the pulse width of the pin 1 to 1S: **IO:PULS:WIDT 0, 1**

Also see

IO:PULSe:WIDTh? <NRL>

IO:PULSe:WIDTh? <NRL>[,][MINimum|MAXimum|DEFault]

This command is used to query the pulse width of the specified pin.

Group

CONFigurable

Syntax

IO:PULSe:WIDTh? <NRL>[,][MINimum|MAXimum|DEFault]

Arguments

<NRL>[,][MINimum|MAXimum|DEFault]

Default Value

None

Returns

<NRf+>

Example

IO:PULS:WIDT? 1

Also see

None

IO:TYPE <NRL>, <CPD>

This command is used to set the specified pin function. Can be set to the first function and the second function: the default first functions of the seven pins are: PSCLear, PASSta, OFFSta, EXTTrig (TOUT), INHLiv (INHLat), SYON, SY-OFF; the second function (i.e., the common function) is: IORD, IOWR and PWM.

- <NRL>: Used to specify the pin number. Pins 1 to 7 correspond to number 0 to 6.
- <CPD>: Used to specify the pin function.

  First function (correspond to pin 1 to 7):

  – PSCLear

  – PSSTa

  – OFFSta

- EXTTrig|TOUT
- INHLiv|INHLat
- SYON
- SYOFF

Second function

- IORD
- IOWR
- PWM

Group

CONFigurable

Syntax

IO:TYPE <NRL>, <CPD>

Arguments

<NRL>, <CPD>

Default Value

None

Returns

None

Example

IO:TYPE 0, PWM

Also see

IO:TYPE? <NRL>

IO:TYPE? <NRL>

This command is used to query the pin function of the specified pin.

- PSCLear: Corresponds to the default function of pin 1. Indicates that the protection status is cleared by this pin when the instrument generates protection.
- PSSTa: Corresponds to the default function of pin 2. Indicates that the output level of pin 2 indicates whether the instrument is protected.
- OFFSta: Corresponds to the default function of pin 3. Used to indicate the **[On/Off]** status of the instrument.
- EXTTrig|TOUT: Corresponds to the default function of pin 4. Indicates that when the instrument generates a trigger signal (trigger Meter, data logging or List function running), a pulse signal will be output from pin 4.
- INHLiv|INHLat: Corresponds to the default function of pin 5. Indicates that the instrument is operated in the LIVING mode by the pin 5.
- SYON: Corresponds to the default function of pin 6. Indicates that pin 6 performs bi-directional and synchronous control over the turn-on of **[On/Off]** under parallel connection.
- SYOFF: Corresponds to the default function of pin 7. Indicates that pin 7 performs bi-directional and synchronous control over the turn-off of **[On/Off]** under parallel connection.

Group

CONFigurable

Syntax

IO:TYPE? <NRL>

Arguments

<NRL>

0 to 6

Default Value

None

Returns

<CRD>

OFFSta|PSSTa|INHLiv|EXTTrig|SYON|SYOFF|INHLat|IORD|IOWR|PSCLear|PWM|TOUT

Example

IO:TYPE? 1

Also see

None

IO:TRIGger:TYPE <CPD>

This command is used to set the Ext-Trig function type of pin 4:

- TOUT: When the instrument generates a trigger signal (trigger Meter, data logging or List function running), a pulse signal will be output from pin 4;
- TIN: Indicates that the instrument will trigger the running of the Meter function, data logging function or List function after receiving a pulse signal from the outside.

Group

CONFigurable

Syntax

IO:TRIGger:TYPE <CPD>

Arguments

<CPD>

TOUT|TIN

Default Value

TOUT

Returns

None

Example

IO:TRIG:TYPE TIN

Also see

IO:TRIGger:TYPE?

IO:TRIGger:TYPE?

This command is used to query the Ext-Trig function type of pin 4.

Group

CONFigurable

Syntax

IO:TRIGger:TYPE?

Arguments

None

Default Value

None

Returns

<CRD>

TOUT|TIN

Example

IO:TRIG:TYPE?

Also see

None

IO:TRIGer:SOURce <CPD>

This command is used to set the trigger source for pin 4:

- METEr: Triggers the running of the meter function;
- DLOG: Triggers the running of the data logging function;

- LIST: Triggers the running of the list function.

Group

CONFigurable

Syntax

IO:TRIGer:SOURce <CPD>

Arguments

<CPD>

METEr|DLOG|LIST

Default Value

METEr

Returns

None

Example

Set the trigger source of pin 4 to LIST: **IO:TRIG:SOUR LIST**

Also see

IO:TRIGer:SOURce?

IO:TRIGer:SOURce?

This command is used to query the trigger source of pin 4.

Group

CONFigurable

Syntax

IO:TRIGer:SOURce?

Arguments

None

Default Value

None

Returns

<CRD>

METEr|DLOG|LIST

Example

IO:TRIG:SOUR?

Also see

None

IO:OUTPut:LEVel <NRL>, <CPD>

This command is used to set the output of the specified pin to a high level or a low level.

Group

CONFigurable

Syntax

IO:OUTPut:LEVel <NRL>, <CPD>

Arguments

<NRL>, <CPD>

0 to 6, 0|LOW|1|HIGH

Default Value

0, 1

Returns

None

Example

IO:OUTP:LEV 1, 0

Also see

IO:OUTPut:LEVel? <NRL>

IO:OUTPut:LEVel? <NRL>

This command is used to query the output of the specified pin is a high level or a low level.

Group

CONFigurable

Syntax

IO:OUTPut:LEVel? <NRL>

Arguments

<NRL>

0 to 6

Default Value

None

Returns

<CRD>

0|LOW|1|HIGH

Example

IO:OUTP:LEV? 1

Also see

None

IO:INPut:LEVel? <NRL>

This command is used to query the input of the specified pin is a high level or a low level.

Group

CONFigurable

Syntax

IO:INPut:LEVel? <NRL>

Arguments

<NRL>

0 to 6

Default Value

None

Returns

<CRD>

0|LOW|1|HIGH

Example

IO:INP:LEV? 1

Also see

None

6 TRIGger Subsystem

The commands in the TRIGger subsystem are used to trigger the use or running of related functions.

TRIGger:ACQuire[:IMMediate]

This command is used to generate a trigger for the Meter function. When the Meter function is triggered, the instrument will display the voltage/current measurement in real time.

Group

TRIGger

Syntax

TRIGger:ACQuire[:IMMediate]

Arguments

None

Default Value

None

Example

TRIG:ACQ

Query Syntax

None

Returns

None

TRIGger:ACQuire:MODE <CPD>

This command is used to set the trigger mode of the Meter function.

- AUTO: Automatic continuous triggering.
- NORMal: Non-automatic triggering, triggered once when the instrument receives a trigger signal from the trigger source.

Group

TRIGger

Syntax

TRIGger:ACQuire:MODE <CPD>

Arguments

<CPD>

AUTO|NORMal

Default Value

NORMal

Example

TRIGger:ACQuire:MODE AUTO

Query Syntax

TRIGger:ACQuire:MODE?

Returns

<CRD>

AUTO|NORMal

TRIGger:ACQuire:STATe?

This command is used to query the status of the Meter function.

- 0: idle

- 1: pretrig
- 2: WTG
- 3: Action
- 4: AcqEnd
- 5:CarryEnd

Group

TRIGger

Syntax

TRIGger:ACQuire:STATe?

Returns

0|1|2|3|4|5

TRIGger:ACQuire:SOURce <CPD>

This command is used to set the trigger source for the Meter function.

- VOLTage: Voltage triggering, which triggers the Meter function when the voltage reaches the set trigger threshold.
- CURRent: Current triggering, which triggers the Meter function when the current reaches the set trigger threshold.
- EXTernal: External triggering, that is, when pin 4 of the digital I/O interface receives a fixed pulse signal, the Meter function is triggered.

  Note

  Before using the external trigger, you need to configure the relevant parameters of pin 4. For details, please refer to the instruction of the **CONFigurable** subsystem.

- BUS: Command (***TRG**) triggering
- MANual: Manual triggering, that is, the Meter function is triggered by the combination key **[Shift]+[On/Off]** (Trigger).
- IMMediate: Trigger immediately, that is, when the instrument receives the **TRIGger:ACQuire[:IMMediate]** command, it triggers the Meter function.

Group

TRIGger

Syntax

TRIGger:ACQuire:SOURce <CPD>

Arguments

<CPD>

VOLTage|CURRent|EXTernal|BUS|MANual|IMMediate

Default Value

IMMediate

Example

TRIGger:ACQuire:SOURce EXTernal

Query Syntax

TRIGger:ACQuire:SOURce?

Returns

<CRD>

VOLTage|CURRent|EXTernal|BUS|MANual|IMMediate

TRIGger:ACQuire:VOLTage:SLOPe <CPD>

This command is used to set the voltage trigger edge of the Meter.

- POSitive: Rising edge
- NEGative: Falling edge
- EITHer: Both rising and falling edges

Note

Before executing this command, you need to set the trigger source of the Meter function to voltage.

Group

TRIGger

Syntax

TRIGger:ACQuire:VOLTage:SLOPe <CPD>

Arguments

<CPD>

POSitive|NEGative|EITHer

Default Value

POSitive

Example

TRIGger:ACQuire:VOLTage:SLOPe NEGative

Query Syntax

TRIGger:ACQuire:VOLTage:SLOPe?

Returns

<CRD>

POSitive|NEGative|EITHer

TRIGger:ACQuire:VOLTage[:LEVel] <NRf+>

This command is used to set the voltage trigger threshold of the Meter.

Group

TRIGger

Syntax

TRIGger:ACQuire:VOLTage[:LEVel] <NRf+>

Arguments

<NRf+>

MIN|MAX|DEF|<value>

Setting range: MIN to MAX

Default Value

MIN

Example

TRIGger:ACQuire:VOLTage[:LEVel] 100

Query Syntax

TRIGger:ACQuire:VOLTage[:LEVel]? [MIN|MAX|DEF]

Returns

<NRf+>

MIN|MAX|DEF|<value>

TRIGger:ACQuire:VOLTage:HYSTeresis:HIGH <NRf+>

This command is used to set the voltage trigger high threshold.

Group

TRIGger

Syntax

TRIGger:ACQuire:VOLTage:HYSTeresis:HIGH <NRf+>

Arguments

<NRf+>

MIN|MAX|DEF|<value>

Setting range: MIN to MAX

Default Value

MIN

Example

TRIGger:ACQuire:VOLTage:HYSTeresis:HIGH 100

Query Syntax

TRIGger:ACQuire:VOLTage:HYSTeresis:HIGH? [MIN|MAX|DEF]

Returns

<NRf+>

MIN|MAX|DEF|<value>

TRIGger:ACQuire:VOLTage:HYSTeresis:LOW <NRf+>

This command is used to set the voltage trigger low threshold.

Group

TRIGger

Syntax

TRIGger:ACQuire:VOLTage:HYSTeresis:LOW <NRf+>

Arguments

<NRf+>

MIN|MAX|DEF|<value>

Setting range: MIN to MAX

Default Value

MIN

Example

TRIGger:ACQuire:VOLTage:HYSTeresis:LOW 10

Query Syntax

TRIGger:ACQuire:VOLTage:HYSTeresis:LOW? [MIN|MAX|DEF]

Returns

<NRf+>

MIN|MAX|DEF|<value>

TRIGger:ACQuire:CURRent:SLOPe <CPD>

This command is used to set the current trigger edge of the Meter.

- POSitive: Rising edge
- NEGative: Falling edge
- EITHer: Both rising and falling edges

Note

Before executing this command, you need to set the trigger source of the Meter function to current.

Group

TRIGger

Syntax

TRIGger:ACQuire:CURRent:SLOPe <CPD>

Arguments

<CPD>

POSitive|NEGative|EITHer

Default Value

POSitive

Example

TRIGger:ACQuire:CURRent:SLOPe NEGative

Query Syntax

TRIGger:ACQuire:CURRent:SLOPe?

Returns

<CRD>

POSitive|NEGative|EITHer

TRIGger:ACQuire:CURRent[:LEVel] <NRf+>

This command is used to set the current trigger threshold of the Meter.

Group

TRIGger

Syntax

TRIGger:ACQuire:CURRent[:LEVel] <NRf+>

Arguments

<NRf+>

MIN|MAX|DEF|<value>

Setting range: MIN to MAX

Default Value

MIN

Example

TRIGger:ACQuire:CURRent[:LEVel] 10

Query Syntax

TRIGger:ACQuire:CURRent[:LEVel]? [MIN|MAX|DEF]

Returns

<NRf+>

MIN|MAX|DEF|<value>

TRIGger:ACQuire:CURRent:HYSTeresis:HIGH <NRf+>

This command is used to set the current trigger high threshold.

Group

TRIGger

Syntax

TRIGger:ACQuire:CURRent:HYSTeresis:HIGH <NRf+>

Arguments

<NRf+>

MIN|MAX|DEF|<value>

Setting range: MIN to MAX

Default Value

MIN

Example

TRIGger:ACQuire:CURRent:HYSTeresis:HIGH 10

Query Syntax

TRIGger:ACQuire:CURRent:HYSTeresis:HIGH? [MIN|MAX|DEF]

Returns

<NRf+>

MIN|MAX|DEF|<value>

TRIGger:ACQuire:CURRent:HYSTeresis:LOW <NRf+>

This command is used to set the current trigger low threshold.

Group

TRIGger

Syntax

TRIGger:ACQuire:CURRent:HYSTeresis:LOW <NRf+>

Arguments

<NRf+>

MIN|MAX|DEF|<value>

Setting range: MIN to MAX

Default Value

MIN

Example

TRIGger:ACQuire:CURRent:HYSTeresis:LOW 1

Query Syntax

TRIGger:ACQuire:CURRent:HYSTeresis:LOW? [MIN|MAX|DEF]

Returns

<NRf+>

MIN|MAX|DEF|<value>

TRIGger:ARB[:IMMediate]

This command is used to generate a trigger for the ARB subsystem function. For example, trigger the output of LIST, car waveform, etc.

Group

TRIGger

Syntax

TRIGger:ARB[:IMMediate]

Arguments

None

Default Value

None

Example

TRIG:ARB

Query Syntax

None

Returns

None

TRIGger:ARB:SOURce <CPD>

This command is used to set the trigger source for the ARB subsystem function.

- EXTernal: External triggering, that is, when pin 4 of the digital I/O interface receives a fixed pulse signal, it triggers the ARB subsystem function.

  Note

  Before using the external trigger, you need to configure the relevant parameters of pin 4. For details, please refer to the instruction of the **CONFigurable** subsystem.

- BUS: Triggered by the command (***TRG**).
- MANual: Manual triggering, that is, the ARB subsystem function is triggered by the combination keys **[Shift]+[On/Off]** (Trigger).

Group

TRIGger

Syntax

TRIGger:ARB:SOURce <CPD>

Arguments

<CPD>

EXTernal|BUS|MANual

Default Value

MANual

Example

TRIGger:ARB:SOURce EXTernal

Query Syntax

TRIGger:ARB:SOURce?

Returns

<CRD>

EXTernal|BUS|MANual

7 STATus Subsystem

Status register programming lets you determine the operating condition of the instrument at any time. The instrument has three groups of status registers; Operation, Questionable, and Standard Event. The Operation and Questionable status groups each consist of the Condition, Enable, and Event registers as well as NTR and PTR filters.

Status Register

The Operation and Questionable status groups use four different types of registers to track qualify, flag, and enable instrument events. The Standard Event group only uses Event and Enable registers.

- A Condition register continuously monitors the state of the instrument. The bits in the condition register are updated in real time and the bits are not latched.
- An PTR/NTR register qualifies the signal that passes to the event register. When a PTR bit is set, signals with positive edge transition pass to the event register. When an NTR bit is set, signals with a negative edge transition pass to the event register. When both bits are set, all signal pass. When neither bits are set, no signals pass.
- An Event register latches transitions that pass through the PTR and NTR registers. When an event bit is set, it remains set until the Event register is read. Reading the Event register clears it.
- An Enable register defines which bits in the event register will be reported to the Status Byte register. You can write to or read from an enable register.

Operation Status Group

These registers record signals that occur during normal operation. The groups consist of a Condition, PTR/NTR, Event, and Enable register.

Questionable Status Group

These registers record signals that indicate abnormal operation. The groups consist of a Condition, PTR/NTR, Event, and Enable register.

Standard Event Status Group

These registers are programmed by Common commands. The group consists of an Event and Enable register. The Standard Event event register latches events relating to communication status. It is a read-only register that is cleared when read. The Standard Event enable register functions similarly to the enable registers of the Operation and Questionable status groups.

Status Byte Register

This register summarizes the information from all other status groups as defined in the IEEE 488.2 Standard Digital Interface for Programmable Instrumentation.

Master Status Summary and Request for Service Bits

MSS is a real-time (unlatched) summary of all Status Byte register bits that are enabled by the Service Request Enable register. MSS is set whenever the instrument has one or more reasons for requesting service. ***STB?** reads the MSS in bit position 6 of the response but does not clear any of the bits in the Status Byte register.

The RQS bit is a latched version of the MSS bit. Whenever the instrument requests service, it sets the SRQ interrupt line true and latches RQS into bit 6 of the Status Byte register. When the controller does a serial poll, RQS is cleared inside the register and returned in bit position 6 of the response. The remaining bits of the Status Byte register are not disturbed.

Error and Output Queues

The Error Queue is a first-in, first-out (FIFO) data register that stores numerical and textual description of an error or event. Error messages are stored until they are read with **SYSTem:ERRor?**. If the queue overflows, the last error/event in the queue is replaced with error -350, "Queue overflow".

The Output Queue is a first-in, first-out (FIFO) data register that stores messages until the controller reads them.

Bit Assignments

Questionable Status Register			
Mnemonic	Bit	Value Bit Weight	Meaning
VF	0	1	Voltage Fault. Either an overvoltage or a reverse voltage has occurred.

OC	1	2	Overcurrent Protection
RS	2	4	Remote Sense: When the real pannel sense is connected, this bit is true or else false.
OP	3	8	Overpower Protection
OT	4	16	Over Temperature Protection
NU1	5	32	Not used
NU2	6	64	Not used
RUN	7	128	List run or stop status, when list is runnig, this bit is true else false.
EPU	8	256	Extended Power Unavailable.
RRV	9	512	Remote Reverse Voltage.
UNR	10	1024	Unknown internal fault of the instrument
LRV	11	2048	Local Reverse Voltage.
OV	12	4096	Overvoltage Protection
PS	13	8192	Fault protection bit (protect shutdown)
VON	14	16384	Voltage of sink current on.
TBF	15	32768	Trace Buffer Full.
UV	16	65536	Undervoltage Protection
UC	17	131072	Undercurrent Protection
Errsense	18	262144	Sense Fault
Share	19	524288	Current sharing fault
INH	20	1048576	Externally inhibited output
OSC	21	2097152	Loop oscillation failure
NU3	22	4194304	Not used
NU4	23	8388608	Not used
Operation Status Bit			
CAL	0	1	Calibrating.
CW	1	2	Constant Power

CR	2	4	Constant Resistance
CC	3	8	Constant Current
CV	4	16	Constant Voltage
WTG	5	32	Waiting.
OFF	6	64	The [on/off] of the instrument is off.
NU1	7	128	Not used
ACQ-WTG	8	256	Waiting for a trigger (indicating the trigger status of Meter, i.e. ACQ)
ARB-WTG	9	512	Waiting for a trigger (indicating the trigger status of ARB)
DLOG-WTG	11	2048	Waiting for a trigger (indicating the trigger status of Dlog)
ACQ-Active	12	4096	ACQ has been triggered and is being executed.
ARB-Active	13	8192	ARB has been triggered and is being executed.
DLOG-Active	14	16384	DLOG has been triggered and is being executed.
Mode	15	32768	Load mode: CC/CR/CW/CV
NU2	16	65536	Not used
Bit description of the standard status register			
OPC	0	1	Operation completed
NU	1	0	Not Used
QYE	2	4	Query Error
DDE	3	8	Device-specific Error
EXE	4	16	Execution Error
CME	5	32	A command syntax error occurred.
NU	6	0	Not Used
PON	7	128	Power On
Bit description of the status byte register			

NU	0~1	0	Not Used
EAV	2	4	Error message cache available
QUES	3	8	One or more bits are set in the Questionable Data Register. Bits must be enabled, see STATus: QUEStionable:ENABle.
MAV	4	16	Data is available in the instrument's output buffer.
ESB	5	32	One or more bits are set in the Standard Event Register. Bits must be enabled, see *ESE.
RQS/MSS	6	64	Master Status Summary and Request for Service Bits
OPER	7	128	One or more bits are set in the Operation Status Register. Bits must be enabled, see STATus:OPERation:ENABle.

STATus:QUEStionable[:EVENt]?

Queries the event register for the Questionable Status group. This is a read-only register, which stores (latches) all events that are passed by the Operation NTR and/or PTR filter. Reading the Questionable Status Event register clears it.

- The value returned is the binary-weighted sum of all enabled bits in the register. For example, with bit 2 (value 4) and bit 4 (value 16) set, the query returns +20.
- ***RST** has no effect on this register.

Group

STATus

Syntax

STATus:QUEStionable[:EVENt]?

Arguments

None

Default Value

None

Returns

<bit value>

Example

STATus:QUEStionable[:EVENt]?

Also see

None

STATus:QUEStionable:ENABle <NR1>

Sets the value of the enable register for the Questionable Status group. The enable register is a mask for enabling specific bits from the Operation Event register to set the QUES (questionable summary) bit of the Status Byte register. **STATus:PRESet** clears all bits in the enable register. ***CLS** does not clear the enable register, but does clear the event register.

Group

STATus

Syntax

STATus:QUEStionable:ENABle <NR1>

Arguments

<NR1>

A decimal value corresponding to the binary weighted sum of the register's bits. Setting range: 0 to 65535.

Default Value

0

Example

Enable bit 2 and 4 in the questionable enable register: **STATus:QUEStionable:ENABle 24**

Query Syntax

STATus:QUEStionable:ENABle?

Returns

<NR1>

STATus:QUEStionable:PTRansition <NR1>

Sets the value of the PTR (Positive-Transition) registers. These registers serve as a polarity filter between the Questionable Condition and Questionable Event registers. When a bit in the PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Questionable Condition register causes that bit in the Questionable Event register to be set. **STATus:PRESet** sets all bits in the PTR registers and clears all bits in the NTR registers.

Group

STATus

Syntax

STATus:QUEStionable:PTRansition <NR1>

Arguments

<NR1>

A decimal value corresponding to the binary weighted sum of the register's bits. Setting range: 0 to 65535.

Default Value

0

Example

Enable bit 3 and 4 in the questionable PTR register: **STATus:QUEStionable:PTRansition 24**

Query Syntax

STATus:QUEStionable:PTRansition?

Returns

<NR1>

STATus:QUEStionable:NTRansition <NR1>

Sets the value of the NTR (Negative-Transition) registers. These registers serve as a polarity filter between the Questionable Condition and Questionable Event registers. When a bit in the NTR register is set to 1, then a 1-to-0 transition of the corresponding bit in the Questionable Condition register causes that bit in the Questionable Event register to be set. **STATus:PRESet** sets all bits in the PTR registers and clears all bits in the NTR registers.

- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Questionable Condition register sets the corresponding bit in the Questionable Event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Questionable Condition register can set the corresponding bit in the Questionable Event register.
- The value returned is the binary-weighted sum of all enabled bits in the register.

Group

STATus

Syntax

STATus:QUEStionable:NTRansition <NR1>

Arguments

<NR1>

A decimal value corresponding to the binary weighted sum of the register's bits. Setting range: 0 to 65535.

Default Value

0

Example

Enable bit 3 and 4 in the questionable NTR register: **STATus:QUEStionable:NTRansition 24**

Query Syntax

STATus:QUEStionable:NTRansition?

Returns

<NR1>

STATus:QUEStionable:CONDition?

Queries the condition register for the Questionable Status group. This is a read-only register, which holds the live (unlatched) operational status of the instrument. Reading the Questionable Status Condition register does not clear it.

- The value returned is the binary-weighted sum of all enabled bits in the register. For example, with bit 2 (value 4) and bit 4 (value 16) set, the query returns +20.
- The condition register bits reflect the current condition. If a condition goes away, the corresponding bit is cleared.
- ***RST** clears this register, other than those bits where the condition still exists after ***RST**.

Group

STATus

Syntax

STATus:QUEStionable:CONDition?

Arguments

None

Default Value

None

Returns

<bit value>

Example

STATus:QUEStionable:CONDition?

Also see

None

STATus:OPERation[:EVENt]?

Queries the event register for the Operation Status group. This is a read-only register, which stores (latches) all events that are passed by the Operation NTR and/or PTR filter. Reading the Operation Status Event register clears it.

- The value returned is the binary-weighted sum of all enabled bits in the register. For example, with bit 3 (value 8) and bit 5 (value 32) set and enabled, the query returns +40.
- ***RST** has no effect on this register.

Group

STATus

Syntax

STATus:OPERation[:EVENt]?

Arguments

None

Default Value

None

Returns

<bit value>

Example

STATus:OPERation[:EVENt]?

Also see

None

STATus:OPERation:ENABle <NR1>

Sets the value of the enable register for the Operation Status group. The enable register is a mask for enabling specific bits from the Operation Event register to set the OPER (operation summary) bit of the Status Byte register. **STATus:PRESet** clears all bits in the enable register. ***CLS** does not clear the enable register, but does clear the event register.

Group

STATus

Syntax

STATus:OPERation:ENABle <NR1>

Arguments

<NR1>

A decimal value corresponding to the binary weighted sum of the register's bits. Setting range: 0 to 65535.

Default Value

0

Example

Enable bit 3 and 4 in the enable register: **STATus:OPERation:ENABle 24**

Query Syntax

STATus:OPERation:ENABle?

Returns

<NR1>

STATus:OPERation:PTRansition <NR1>

Sets the value of the PTR (Positive-Transition) registers. These registers serve as a polarity filter between the Operation Condition and Operation Event registers. When a bit in the PTR register is set to 1, then a 0-to-1 transition of the corresponding bit in the Operation Condition register causes that bit in the Operation Event register to be set. **STATus:PRESet** sets all bits in the PTR registers and clears all bits in the NTR registers.

Group

STATus

Syntax

STATus:OPERation:PTRansition <NR1>

Arguments

<NR1>

A decimal value corresponding to the binary weighted sum of the register's bits. Setting range: 0 to 65535.

Default Value

0

Example

Enable bit 3 and 4 in the PTR register: **STATus:OPERation:PTRansition 24**

Query Syntax

STATus:OPERation:PTRansition?

Returns

<NR1>

STATus:OPERation:NTRansition <NR1>

Sets the value of the NTR (Negative-Transition) registers. These registers serve as a polarity filter between the Operation Condition and Operation Event registers. When a bit in the NTR register is set to 1, then a 1-to-0 transition of the corresponding bit in the Operation Condition register causes that bit in the Operation Event register to be set. **STATus:PRESet** sets all bits in the PTR registers and clears all bits in the NTR registers.

- If the same bits in both NTR and PTR registers are set to 1, then any transition of that bit at the Operation Condition register sets the corresponding bit in the Operation Event register.
- If the same bits in both NTR and PTR registers are set to 0, then no transition of that bit at the Operation Condition register can set the corresponding bit in the Operation Event register.
- The value returned is the binary-weighted sum of all enabled bits in the register.

Group

STATus

Syntax

STATus:OPERation:NTRansition <NR1>

Arguments

<NR1>

A decimal value corresponding to the binary weighted sum of the register's bits. Setting range: 0 to 65535.

Default Value

0

Example

Enable bit 3 and 4 in the NTR register: **STATus:OPERation:NTRansition 24**

Query Syntax

STATus:OPERation:NTRansition?

Returns

<NR1>

STATus:OPERation:CONDition?

Queries the condition register for the Operation Status group. This is a read-only register, which holds the live (unlatched) operational status of the instrument. Reading the Operation Status Condition register does not clear it.

- The value returned is the binary-weighted sum of all enabled bits in the register. For example, with bit 3 (value 8) and bit 5 (value 32) set and enabled, the query returns +40.
- The condition register bits reflect the current condition. If a condition goes away, the corresponding bit is cleared.
- ***RST** clears this register, other than those bits where the condition still exists after ***RST**.

Group

STATus

Syntax

STATus:OPERation:CONDition?

Arguments

None

Default Value

None

Returns

<bit value>

Example

STATus:OPERation:CONDition?

Also see

None

STATus:PRESet

Presets all Enable, PTR, and NTR registers.

Operation register	Questionable register	Preset setting
STAT:OPER:ENAB	STAT:QUES:ENAB	All defined bits are disabled
STAT:OPER:NTR	STAT:QUES:NTR	All defined bits are disabled
STAT:OPER:PTR	STAT:QUES:PTR	All defined bits are disabled

Group

STATus

Syntax

STATus:PRESet

Arguments

None

Default Value

None

Returns

None

Example

Preset the Operation and Questionable registers: **STATus:PRESet**

Also see

None

8 FETCh Subsystem

Fetch commands return measurement data that has been previously acquired. FETCh queries do not generate new measurements, but allow additional measurement calculations from the same acquired data. The data is valid until the next MEASure or INITiate command occurs.

FETCh[:SCALar]:CURRent[:DC]?

This command is used to get the average value of the Meter current.

Group

FETCh

Syntax

FETCh[:SCALar]:CURRent[:DC]?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:CURRent[:DC]?

Also see

None

FETCh[:SCALar]:CURRent[:DC]:HIGH?

Returns the High level of a pulse waveform. Values returned in amperes.

Group

FETCh

Syntax

FETCh[:SCALar]:CURRent[:DC]:HIGH?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:CURRent[:DC]:HIGH?

Also see

None

FETCh[:SCALar]:CURRent[:DC]:LOW?

Returns the Low level of a pulse waveform. Values returned in amperes.

Group

FETCh

Syntax

FETCh[:SCALar]:CURRent[:DC]:LOW?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:CURRent[:DC]:LOW?

Also see

None

FETCh[:SCALar]:CURRent[:DC]:MAXimum?

Returns the maximum value of Meter current. Values returned in amperes.

Group

FETCh

Syntax

FETCh[:SCALar]:CURRent[:DC]:MAXimum?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:CURRent[:DC]:MAXimum?

Also see

None

FETCh[:SCALar]:CURRent[:DC]:MINimum?

Returns the minimum value of Meter current. Values returned in amperes.

Group

FETCh

Syntax

FETCh[:SCALar]:CURRent[:DC]:MINimum?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:CURRent[:DC]:MINimum?

Also see

None

FETCh[:SCALar]:VOLTage[:DC]?

This command is used to get the average value of the Meter voltage.

Group

FETCh

Syntax

FETCh[:SCALar]:VOLTage[:DC]?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:VOLTage[:DC]?

Also see

None

FETCh[:SCALar]:VOLTage[:DC]:HIGH?

Returns the High level of a pulse waveform. Values returned in volts.

Group

FETCh

Syntax

FETCh[:SCALar]:VOLTage[:DC]:HIGH?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:VOLTage[:DC]:HIGH?

Also see

None

FETCh[:SCALar]:VOLTage[:DC]:LOW?

Returns the Low level of a pulse waveform. Values returned in volts.

Group

FETCh

Syntax

FETCh[:SCALar]:VOLTage[:DC]:LOW?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:VOLTage[:DC]:LOW?

Also see

None

FETCh[:SCALar]:VOLTage[:DC]:MAXimum?

Returns the maximum value of Meter voltage. Values returned in volts.

Group

FETCh

Syntax

FETCh[:SCALar]:VOLTage[:DC]:MAXimum?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:VOLTage[:DC]:MAXimum?

Also see

None

FETCh[:SCALar]:VOLTage[:DC]:MINimum?

Returns the minimum value of Meter voltage. Values returned in volts.

Group

FETCh

Syntax

FETCh[:SCALar]:VOLTage[:DC]:MINimum?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:VOLTage[:DC]:MINimum?

Also see

None

FETCh[:SCALar]:POWer[:DC]?

This command is used to get the average value of the Meter power.

Group

FETCh

Syntax

FETCh[:SCALar]:POWer[:DC]?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:POWer[:DC]?

Also see

None

FETCh[:SCALar]:POWer[:DC]:MAXimum?

Returns the maximum value of Meter power. Values returned in watts.

Group

FETCh

Syntax

FETCh[:SCALar]:POWer[:DC]:MAXimum?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:POWer[:DC]:MAXimum?

Also see

None

FETCh[:SCALar]:POWer[:DC]:MINimum?

Returns the minimum value of Meter power. Values returned in watts.

Group

FETCh

Syntax

FETCh[:SCALar]:POWer[:DC]:MINimum?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]:POWer[:DC]:MINimum?

Also see

None

FETCh:ARRay:CURRent[:DC]?

Returns the instantaneous measurement. Values returned in amperes. Data is returned as single precision floating point values in a finite-length arbitrary block response format.

Group

FETCh

Syntax

FETCh:ARRay:CURRent[:DC]?

Arguments

None

Default Value

None

Returns

<Block>

Example

Returns the measured current array: **FETCh:ARRay:CURRent[:DC]?**

Also see

None

FETCh:ARRay:VOLTage[:DC]?

Returns the instantaneous measurement. Values returned in volts. Data is returned as single precision floating point values in a finite-length arbitrary block response format.

Group

FETCh

Syntax

FETCh:ARRay:VOLTage[:DC]?

Arguments

None

Default Value

None

Returns

<Block>

Example

Returns the measured voltage array: **FETCh:ARRay:VOLTage[:DC]?**

Also see

None

FETCh:ARRay:POWer[:DC]?

Returns the instantaneous measurement. Values returned in watts. Data is returned as single precision floating point values in a finite-length arbitrary block response format.

Group

FETCh

Syntax

FETCh:ARRay:POWer[:DC]?

Arguments

None

Default Value

None

Returns

<Block>

Example

Returns the measured power array: **FETCh:ARRay:POWer[:DC]?**

Also see

None

FETCh[:SCALar]?

This command is used to obtain a variety of data: voltage, current, power.

Group

FETCh

Syntax

FETCh[:SCALar]?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh[:SCALar]?

Also see

None

FETCh:ACMeter:EACStage?

This command is used to read the total regenerative power.

Group

FETCh

Syntax

FETCh:ACMeter:EACStage?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh:ACMeter:EACStage?

Also see

None

FETCh:ACMeter:EACTotal?

This command is used to read the total historical regenerative power.

Group

FETCh

Syntax

FETCh:ACMeter:EACTotal?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh:ACMeter:EACTotal?

Also see

None

9 MEASure Subsystem

Measure commands measure the input voltage or current. They trigger the acquisition of new data before returning the reading. Measurements are performed by digitizing the instantaneous input voltage or current for a specified measurement time, storing the results in a buffer, and calculating the value for the specified measurement type.

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

Initiates, triggers, and returns the averaged current measurement. Values returned in amperes.

Group

MEASure

Syntax

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

Arguments

None

Default Value

None

Returns

<NRf+>

Example

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

Also see

None

MEASure[:SCALar]:CURRent:HIGH?

Initiates, triggers, and returns the High level of a pulse waveform. Values returned in amperes.

Group

MEASure

Syntax

MEASure[:SCALar]:CURRent:HIGH?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEASure[:SCALar]:CURRent:HIGH?

Also see

None

MEASure[:SCALar]:CURRent:LOW?

Initiates, triggers, and returns the Low level of a pulse waveform. Values returned in amperes.

Group

MEASure

Syntax

MEASure[:SCALar]:CURRent:LOW?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEASure[:SCALar]:CURRent:LOW?

Also see

None

MEASure[:SCALar]:CURRent:MAXimum?

Initiates, triggers, and returns the maximum values of a current measurement. Values returned in amperes.

Group

MEASure

Syntax

MEASure[:SCALar]:CURRent:MAXimum?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEASure[:SCALar]:CURRent:MAXimum?

Also see

None

MEASure[:SCALar]:CURRent:MINimum?

Initiates, triggers, and returns the minimum values of a current measurement. Values returned in amperes.

Group

MEASure

Syntax

MEASure[:SCALar]:CURRent:MINimum?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEASure[:SCALar]:CURRent:MINimum?

Also see

None

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

Initiates, triggers, and returns the averaged voltage measurement. Values returned in volts.

Group

MEASure

Syntax

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

Arguments

None

Default Value

None

Returns

<NRf+>

Example

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

Also see

None

MEASure[:SCALar]:VOLTage:HIGH?

Initiates, triggers, and returns the High level of a pulse waveform. Values returned in volts.

Group

MEASure

Syntax

MEASure[:SCALar]:VOLTage:HIGH?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEASure[:SCALar]:VOLTage:HIGH?

Also see

None

MEASure[:SCALar]:VOLTage:LOW?

Initiates, triggers, and returns the Low level of a pulse waveform. Values returned in volts.

Group

MEASure

Syntax

MEASure[:SCALar]:VOLTage:LOW?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEASure[:SCALar]:VOLTage:LOW?

Also see

None

MEASure[:SCALar]:VOLTage:MAXimum?

Initiates, triggers, and returns the maximum values of a voltage measurement. Values returned in volts.

Group

MEASure

Syntax

MEASure[:SCALar]:VOLTage:MAXimum?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEASure[:SCALar]:VOLTage:MAXimum?

Also see

None

MEASure[:SCALar]:VOLTage:MINimum?

Initiates, triggers, and returns the minimum values of a voltage measurement. Values returned in volts.

Group

MEASure

Syntax

MEASure[:SCALar]:VOLTage:MINimum?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEASure[:SCALar]:VOLTage:MINimum?

Also see

None

MEASure[:SCALar]:POWer[:DC]?

Initiates, triggers, and returns the averaged power measurement. Values returned in watts.

Group

MEASure

Syntax

MEASure[:SCALar]:POWer[:DC]?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEASure[:SCALar]:POWer[:DC]?

Also see

None

MEASure[:SCALar]:POWer:MAXimum?

Initiates, triggers, and returns the maximum values of a power measurement. Values returned in watts.

Group

MEASure

Syntax

MEASure[:SCALar]:POWer:MAXimum?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEASure[:SCALar]:POWer:MAXimum?

Also see

None

MEASure[:SCALar]:POWer:MINimum?

Initiates, triggers, and returns the minimum values of a power measurement. Values returned in watts.

Group

MEASure

Syntax

MEASure[:SCALar]:POWer:MINimum?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEASure[:SCALar]:POWer:MINimum?

Also see

None

MEASure:ARRay:CURRent[:DC]?

Initiates and triggers a current measurement; returns a list of the digitized current measurement samples. Values returned in amperes. Data is returned as single precision floating point values in a finite-length arbitrary block response format.

Group

MEASure

Syntax

MEASure:ARRay:CURRent[:DC]?

Arguments

None

Default Value

None

Returns

<Block>

Example

Returns the measured current array: **MEASure:ARRay:CURRent[:DC]?**

Also see

None

MEASure:ARRay:VOLTage[:DC]?

Initiates and triggers a voltage measurement; returns a list of the digitized voltage measurement samples. Values returned in volts. Data is returned as single precision floating point values in a finite-length arbitrary block response format.

Group

MEASure

Syntax

MEASure:ARRay:VOLTage[:DC]?

Arguments

None

Default Value

None

Returns

<Block>

Example

Returns the measured voltage array: **MEASure:ARRay:VOLTage[:DC]?**

Also see

None

MEASure:ARRay:POWer[:DC]?

Initiates and triggers a power measurement; returns a list of the digitized power measurement samples. Values returned in watts. Data is returned as single precision floating point values in a finite-length arbitrary block response format.

Group

MEASure

Syntax

MEASure:ARRay:POWer[:DC]?

Arguments

None

Default Value

None

Returns

<Block>

Example

Returns the measured power array: **MEASure:ARRay:POWer[:DC]?**

Also see

None

MEASure[:SCALar]?

This command is used to measure a variety of data: voltage, current, power.

Group

MEASure

Syntax

MEASure[:SCALar]?

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEASure[:SCALar]?

Also see

None

10 PARallel Subsystem

The PARallel subsystem contains instructions for instrument parallel operation.

PARallel:ROLE <CPD>

This command is used to select the instrument role in the case of parallel.

- SINGle: Single mode
- SLAVe: Slave mode
- MASTer: Master mode

Group

PARallel

Syntax

PARallel:ROLE <CPD>

Arguments

<CPD>

SINGle|SLAVe|MASTer

Default Value

SINGle

Example

PARallel:ROLE MASTer

Query Syntax

PARallel:ROLE?

Returns

<CRD>

SINGle|SLAVe|MASTer

PARallel:NUMBer <NR1>

This command is used to set the total number of instruments in the case of parallel (i.e., the sum of the masters and slaves).

Group

PARallel

Syntax

PARallel:NUMBer <NR1>

Arguments

<NR1>

Setting range: 2 to 16

Default Value

2

Example

PARallel:NUMBer 3

Query Syntax

PARallel:NUMBer?

Returns

<NR1>

11 SENSe Subsystem

Sense commands control the measurement ranges and window as well as the data acquisition sequence.

[SOURce:]REMote:SENSe[:STATe] <Bool>

This command is used to set the on/off state of the Sense function.

- 0|OFF: Turn off
- 1|ON: Turn on

Group

SENSe

Syntax

[SOURce:]REMote:SENSe[:STATe] <Bool>

Arguments

<Bool>

0|OFF|1|ON

Default Value

0|OFF

Example

REM:SENS 1

Query Syntax

[SOURce:]REMote:SENSe[:STATe]?

Returns

<Bool>

0|OFF|1|ON

SENSe:ACQuire:POINts <NRf+>

This command is used to set the number of sampling points that the Meter completes a measurement.

Group

SENSe

Syntax

SENSe:ACQuire:POINts <NRf+>

Arguments

<NRf+>

MIN|MAX|DEF|<value>

Setting range: MIN to MAX

Default Value

10

Example

SENSe:ACQuire:POINts 50

Query Syntax

SENSe:ACQuire:POINts? [MIN|MAX|DEF]

Returns

<NRf+>

SENSe:ACQuire:OFFSet:POINt <NRf+>

This command is used to set the number of Meter trigger offset points.

Group

SENSe

Syntax

SENSe:ACQuire:OFFSet:POINt <NRf+>

Arguments

<NRf+>

MIN|MAX|DEF|<value>

Setting range: MIN to MAX

Default Value

MIN

Example

SENSe:ACQuire:OFFSet:POINt 10

Query Syntax

SENSe:ACQuire:OFFSet:POINt? [MIN|MAX|DEF]

Returns

<NRf+>

SENSe:ACQuire:TINTerval <NRf+>

This command is used to set the sampling time interval of the Meter module.

Group

SENSe

Syntax

SENSe:ACQuire:TINTerval <NRf+>

Arguments

<NRf+>

MIN|MAX|DEF|<value>

Setting range: MIN to MAX

Default Value

2e-05

Example

SENSe:ACQuire:TINTerval 0.1

Query Syntax

SENSe:ACQuire:TINTerval? [MIN|MAX|DEF]

Returns

<NRf+>

12 SOURce Subsystem

- [SOURce:]FUNCtion <CPD>
- [SOURce:]LOOP:SPEed <CPD>
- [SOURce:]FUNCtion:MODE <CPD>
- [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <NRf+>
- [SOURce:]CURRent:SLEW[:BOTH] <NRf+>
- [SOURce:]CURRent:SLEW:POSitive <NRf+>
- [SOURce:]CURRent:SLEW:NEGative <NRf+>
- [SOURce:]CURRent[:OVER]:PROTection:STATe <Bool>
- [SOURce:]CURRent[:OVER]:PROTection[:LEVel] <NRf+>
- [SOURce:]CURRent[:OVER]:PROTection:DELay <NRf+>
- [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf+>
- [SOURce:]VOLTage:SLEW[:BOTH] <NRf+>
- [SOURce:]VOLTage:SLEW:POSitive <NRf+>
- [SOURce:]VOLTage:SLEW:NEGative <NRf+>
- [SOURce:]VOLTage:ON[:LEVel] <NRf+>
- [SOURce:]VOLTage:UNDer:PROTection:STATe <Bool>
- [SOURce:]VOLTage:UNDer:PROTection[:LEVel] <NRf+>
- [SOURce:]VOLTage:UNDer:PROTection:DELay <NRf+>
- [SOURce:]VOLTage:UNDer:PROTection:WARM <NRf+>
- [SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude] <NRf+>
- [SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude] <NRf+>
- [SOURce:]POWer:PROTection:STATe <bool>
- [SOURce:]POWer:PROTection[:LEVel] <NRf+>
- [SOURce:]POWer:PROTection:DELay <NRf+>
- [SOURce:]POWer:SLEW[:BOTH] <NRf+>
- [SOURce:]POWer:SLEW:POSitive <NRf+>
- [SOURce:]POWer:SLEW:NEGative <NRf+>
- [SOURce:]VOLTage[:ON]:LATCh[:STATe] <Bool>
- [SOURce:]VOLTage:ON:HYSTeresis[:LEVel] <NRf+>
- [SOURce:]EXTern[:STATe] <Bool>
- [SOURce:]ACMeter:EACStage:CLEar

12.1 [SOURce:]FUNCtion <CPD>

This command is used to set the operation mode of the load.

Group

SOURce

Syntax

[SOURce:]FUNCtion <CPD>

Arguments

<CPD>

VOLTage|CURRent|POWer|RESistance|CVCC|CRCC|CVCR|AUTO

Default Value

CURRent

Example

[SOUR:]FUNC VOLT

Query Syntax

[SOURce:]FUNCtion?

Returns

VOLTage|CURRent|POWer|RESistance|CVCC|CRCC|CVCR|AUTO

12.2 [SOURce:]LOOP:SPEed <CPD>

This command is used to set the loop speed of the load.

Group

SOURce

Syntax

[SOURce:]LOOP:SPEed <CPD>

Arguments

<CPD>

HIGH|LOW

Default Value

HIGH

Example

LOOP:SPEed HIGH

Query Syntax

[SOURce:]LOOP:SPEed?

Returns

HIGH|LOW

12.3 [SOURce:]FUNCtion:MODE <CPD>

This command is used to set the functional mode of the load, i.e. transient mode. This determines what happens to the input current when the transient system is initiated and triggered.

- FIXed: The default value indicates that the instrument is operating in fixed mode. Keeps the input current at its immediate value.
- LIST: Indicates that the instrument is operating in LIST mode. When a trigger occurs, LIST will cause the input to follow the list value.
- BATTery: Indicates that the instrument is operating in battery test mode. When a trigger occurs, it is input based on the edited battery test file.

Group

SOURce

Syntax

[SOURce:]FUNCtion:MODE <CPD>

Arguments

<CPD>

FIXed|LIST|BATTery

Default Value

FIXed

Example

FUNC:MODE FIX

Query Syntax

[SOURce:]FUNCtion:MODE?

Returns

<CPD>

12.4 [SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <NRf+>

This command is used to set the input current value in CC operation mode.

Group

SOURce

Syntax

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

CURR 5

Query Syntax

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]?

Returns

<NRf+>

12.5 [SOURce:]CURRent:SLEW[:BOTH] <NRf+>

This command is used to set the current rise and fall time, that is, the two times are set at the same time and take effect at the same time.

Group

SOURce

Syntax

[SOURce:]CURRent:SLEW[:BOTH] <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN~MAX

Default Value

None

Example

CURR:SLEW[:BOTH] 1

Query Syntax

[SOURce:]CURRent:SLEW[:BOTH]?

Returns

<NRf+>

12.6 [SOURce:]CURRent:SLEW:POSitive <NRf+>

This command is used to set the current rise time.

Group

SOURce

Syntax

[SOURce:]CURRent:SLEW:POSitive <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

CURR:SLEW:POS 1

Query Syntax

[SOURce:]CURRent:SLEW:POSitive?

Returns

<NRf+>

12.7 [SOURce:]CURRent:SLEW:NEGative <NRf+>

This command is used to set the current fall time.

Group

SOURce

Syntax

[SOURce:]CURRent:SLEW:NEGative <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN~MAX

Default Value

None

Example

CURR:SLEW:NEG 1

Query Syntax

[SOURce:]CURRent:SLEW:NEGative?

Returns

<NRf+>

12.8 [SOURce:]CURRent[:OVER]:PROTection: STATe <Bool>

Enable or disable overcurrent protection. If the overcurrent protection function is enabled and the input enters a current limit, the input is disabled.

Group

SOURce

Syntax

[SOURce:]CURRent[:OVER]:PROTection:STATe <Bool>

Arguments

<Bool>

0|OFF|1|ON

Default Value

0|OFF

Example

CURR:PROT:STAT 1

Query Syntax

[SOURce:]CURRent[:OVER]:PROTection:STATe?

Returns

0|1

12.9 [SOURce:]CURRent[:OVER]:PROTection[:LEVel] <NRf+>

This command is used to set the limit value of overcurrent protection.

Group

SOURce

Syntax

[SOURce:]CURRent[:OVER]:PROTection[:LEVel] <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

CURR:PROT 1

Query Syntax

[SOURce:]CURRent[:OVER]:PROTection[:LEVel]?

Returns

<NRf+>

12.10 [SOURce:]CURRent[:OVER]:PROTection:DELay <NRf+>

This command is used to set the delay time of overcurrent protection.

Group

SOURce

Syntax

[SOURce:]CURRent[:OVER]:PROTection:DELay <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

CURR:PROT:DEL 10

Query Syntax

[SOURce:]CURRent[:OVER]:PROTection:DELay?

Returns

<NRf+>

12.11 [SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf+>

This command is used to set the input voltage value in CV operation mode.

Group

SOURce

Syntax

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

VOLT 5

Query Syntax

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]?

Returns

<NRf+>

12.12 [SOURce:]VOLTage:SLEW[:BOTH] <NRf+>

This command is used to set the voltage rise and fall time, that is, the two times are set at the same time and take effect at the same time.

Group

SOURce

Syntax

[SOURce:]VOLTage:SLEW[:BOTH] <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN~MAX

Default Value

None

Example

VOLT:SLEW[:BOTH] 1

Query Syntax

[SOURce:]VOLTage:SLEW[:BOTH]?

Returns

<NRf+>

12.13 [SOURce:]VOLTage:SLEW:POSitive <NRf+>

This command is used to set the voltage rise time.

Group

SOURce

Syntax

[SOURce:]VOLTage:SLEW:POSitive <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

VOLT:SLEW:POS 1

Query Syntax

[SOURce:]VOLTage:SLEW:POSitive?

Returns

<NRf+>

12.14 [SOURce:]VOLTage:SLEW:NEGative <NRf+>

This command is used to set the voltage fall time.

Group

SOURce

Syntax

[SOURce:]VOLTage:SLEW:NEGative <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN~MAX

Default Value

None

Example

VOLT:SLEW:NEG 1

Query Syntax

[SOURce:]VOLTage:SLEW:NEGative?

Returns

<NRf+>

12.15 [SOURce:]VOLTage:ON[:LEVel] <NRf+>

This command is used to set the start load voltage of the load.

Group

SOURce

Syntax

[SOURce:]VOLTage:ON[:LEVel] <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

MAXimum

Example

VOLT:ON 50

Query Syntax

[SOURce:]VOLTage:ON[:LEVel]?

Returns

<NRf+>

12.16 [SOURce:]VOLTage:UNDer:PROTection:STATe <Bool>

Enable or disable undervoltage protection. If undervoltage protection is enabled and the input reaches the voltage limit, the input is disabled.

Group

SOURce

Syntax

[SOURce:]VOLTage:UNDer:PROTection:STATe <Bool>

Arguments

<Bool>

0|OFF|1|ON

Default Value

0|OFF

Example

VOLT:UND:PROT:STAT 1

Query Syntax

[SOURce:]VOLTage:UNDer:PROTection:STATe?

Returns

0|1

12.17 [SOURce:]VOLTage:UNDer:PROTection[:LEVel] <NRf+>

This command is used to set the limit value of undervoltage protection.

Group

SOURce

Syntax

[SOURce:]VOLTage:UNDer:PROTection[:LEVel] <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

VOLT:UND:PROT 5

Query Syntax

[SOURce:]VOLTage:UNDer:PROTection[:LEVel]?

Returns

<NRf+>

12.18 [SOURce:]VOLTage:UNDer:PROTection:DELay <NRf+>

This command is used to set the delay time of undervoltage protection.

Group

SOURce

Syntax

[SOURce:]VOLTage:UNDer:PROTection:DELay <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

VOLT:UND:PROT:DEL 10

Query Syntax

[SOURce:]VOLTage:UNDer:PROTection:DELay?

Returns

<NRf+>

12.19 [SOURce:]VOLTage:UNDer:PROTection:WARM <NRf+>

This command is used to set the warm time of undervoltage protection.

Group

SOURce

Syntax

[SOURce:]VOLTage:UNDer:PROTection:WARM <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN~MAX

Default Value

None

Example

VOLT:UND:PROT:WARM 5

Query Syntax

[SOURce:]VOLTage:UNDer:PROTection:WARM?

Returns

<NRf+>

12.20 [SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude] <NRf+>

This command is used to set the input resistance value in CR operation mode.

Group

SOURce

Syntax

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude] <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

RES 5

Query Syntax

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude]?

Returns

<NRf+>

12.21 [SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude] <NRf+>

This command is used to set the power value in CW operation mode.

Group

SOURce

Syntax

[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude] <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

POW 10

Query Syntax

[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]?

Returns

<NRf+>

12.22 [SOURce:]POWer:PROTection:STATe <bool>

Enable or disable over power protection. If power protection is enabled and the input reaches the power limit, the input is disabled.

Group

SOURce

Syntax

[SOURce:]POWer:PROTection:STATe <bool>

Arguments

<Bool>

0|OFF|1|ON

Default Value

0|OFF

Example

POW:PROT:STAT 1

Query Syntax

[SOURce:]POWer:PROTection:STATe?

Returns

0|1

12.23 [SOURce:]POWer:PROTection[:LEVel] <NRf+>

This command is used to set the limit value of over power protection.

Group

SOURce

Syntax

[SOURce:]POWer:PROTection[:LEVel] <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

POW:PROT 10

Query Syntax

[SOURce:]POWer:PROTection[:LEVel]?

Returns

<NRf+>

12.24 [SOURce:]POWer:PROTection:DELay <NRf+>

This command is used to set the delay time of over power protection.

Group

SOURce

Syntax

[SOURce:]POWer:PROTection:DELay <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

POW:PROT:DEL 5

Query Syntax

[SOURce:]POWer:PROTection:DELay?

Returns

<NRf+>

12.25 [SOURce:]POWer:SLEW[:BOTH] <NRf+>

This command is used to set the power rise and fall time, that is, the two times are set at the same time and take effect at the same time.

Group

SOURce

Syntax

[SOURce:]POWer:SLEW[:BOTH] <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN~MAX

Default Value

None

Example

POW:SLEW[:BOTH] 1

Query Syntax

[SOURce:]POWer:SLEW[:BOTH]?

Returns

<NRf+>

12.26 [SOURce:]POWer:SLEW:POSitive <NRf+>

This command is used to set the power rise time.

Group

SOURce

Syntax

[SOURce:]POWer:SLEW:POSitive <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

POW:SLEW:POS 1

Query Syntax

[SOURce:]POWer:SLEW:POSitive?

Returns

<NRf+>

12.27 [SOURce:]POWer:SLEW:NEGative <NRf+>

This command is used to set the power fall time.

Group

SOURce

Syntax

[SOURce:]POWer:SLEW:NEGative <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN~MAX

Default Value

None

Example

POW:SLEW:NEG 1

Query Syntax

[SOURce:]POWer:SLEW:NEGative?

Returns

<NRf+>

12.28 [SOURce:]VOLTage[:ON]:LATCh[:STATe] <Bool>

This command is used to turn the Von latch mode on or off.

Group

SOURce

Syntax

[SOURce:]VOLTage[:ON]:LATCh[:STATe] <Bool>

Arguments

<Bool>

0|OFF|1|ON

Default Value

None

Example

VOLT:LATCh 1

Query Syntax

[SOURce:]VOLTage[:ON]:LATCh[:STATe]?

Returns

0|1

12.29 [SOURce:]VOLTage:ON:HYSTeresis[:LEVel] <NRf+>

This command sets the hysteresis value in the Living mode of the load.

Group

SOURce

Syntax

[SOURce:]VOLTage:ON:HYSTeresis[:LEVel] <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

None

Example

VOLT:ON:HYST 5

Query Syntax

[SOURce:]VOLTage:ON:HYSTeresis[:LEVel]?

Returns

<NRf+>

12.30 [SOURce:]EXTern[:STATe] <Bool>

This command is used to set the external control status.

Group

SOURce

Syntax

[SOURce:]EXTern[:STATe] <Bool>

Arguments

0|OFF|1|ON

Default Value

0

Example

EXT 1

Query Syntax

[SOURce:]EXTern[:STATe]?

Returns

0|1

12.31 [SOURce:]ACMeter:EACStage:CLEar

Resets the total regenerative power to zero.

Group

SOURce

Syntax

[SOURce:]ACMeter:EACStage:CLEar

Arguments

None

Default Value

None

Example

[SOURce:]ACMeter:EACStage:CLEar

Query Syntax

None

Returns

None

13 INPut Subsystem

- INPut[:STATe] <CPD>
- INPut:DELay:FALL <NRf+>
- INPut:DELay:RISE <NRf+>
- INPut:SHORt[:STATe] <CPD>
- PROTection:WDOG[:STATe] <CPD>
- PROTection:WDOG:DELay <NRf+>

13.1 INPut[:STATe] <CPD>

Enable or disable the input.

Group

INPut

Syntax

INPut[:STATe] <CPD>

Arguments

<CPD>

0|OFF|1|ON

Default Value

0|OFF

Example

INP 1

Query Syntax

INPut[:STATe]?

Returns

0|1

13.2 INPut:DELay:FALL <NRf+>

This command is used to set the delay time for the input to be turned off.

Group

INPut

Syntax

INPut:DELay:FALL <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

DEFault: 0

Example

INP:DEL:FALL 6

Query Syntax

INPut:DELay:FALL?

Returns

<NRf+>

13.3 INPut:DELay:RISE <NRf+>

This command is used to set the delay time for the input to be turned on.

Group

INPut

Syntax

INPut:DELay:RISE <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

DEFault: 0

Example

INP:DEL:RISE 5

Query Syntax

INPut:DELay:RISE?

Returns

<NRf+>

13.4 INPut:SHORt[:STATe] <CPD>

Enable or disable the short mode.

Group

INPut

Syntax

INPut:SHORt[:STATe] <CPD>

Arguments

0|OFF|1|ON

Default Value

0|OFF

Example

INP:SHOR 1

Query Syntax

INPut:SHORt[:STATe]?

Returns

0|1

13.5 PROTection:WDOG[:STATe] <CPD>

Enables or disables the I/O watchdog timer. When enabled, the input will be disabled if there is no I/O activity on any remote interface within the time period specified by the **PROTection:WDOG:DELay** command. The input is latched off but the programmed input state is not changed.

Syntax

PROTection:WDOG[:STATe] <CPD>

Arguments

<CPD>

0|OFF|1|ON

Default Value

0|OFF

Returns

None

Example

PROT:WDOG 1

Query syntax

PROTection:WDOG[:STATe]?

Returns

0|1

13.6 PROTection:WDOG:DELay <NRf+>

Sets the watchdog delay time. When the watchdog timer is enabled, the input is disabled if there is no SCPI I/O activity on any remote interface within the delay time. The watchdog timer function is NOT reset by activity on the front panel - the input will still shut down after the time period has elapsed. Programmed values can range from 1 to 3600 seconds in 1 second increments.

Syntax

PROTection:WDOG:DELay <NRf+>

Arguments

<NRf+>

<*value*>|MIN|MAX|DEF

Default Value

DEF: 60 S

Returns

None

Example

PROT:WDOG:DEL 600

Query syntax

PROTection:WDOG:DELay?

Returns

<NRf+>

14 BATTery Subsystem

- BATTery:SHUT:CAPacity <NRf+>
- BATTery:SHUT:TIME <NRf+>
- BATTery:SHUT:VOLTage <NRf+>

14.1 BATTery:SHUT:CAPacity <NRf+>

This command is used to set the capacitance value of the battery test cutoff.

Group

BATTery

Syntax

BATTery:SHUT:CAPacity <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

DEFault: 1 AH

Example

BATT:SHUT:CAP 50

Query Syntax

BATTery:SHUT:CAPacity?

Returns

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

14.2 BATTery:SHUT:TIME <NRf+>

This command is used to set the battery test cutoff time.

Group

BATTery

Syntax

BATTery:SHUT:TIME <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

DEFault: 20 s

Example

BATTery:SHUT:TIME 3

Query Syntax

BATTery:SHUT:TIME?

Returns

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

14.3 BATTery:SHUT:VOLTage <NRf+>

This command is used to set the voltage value for the battery test cutoff.

Group

BATTery

Syntax

BATTery:SHUT:VOLTage <NRf+>

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

Setting range: MIN to MAX

Default Value

DEFault: 0

Example

BATT:SHUT:VOLT 200

Query Syntax

BATTery:SHUT:VOLTage?

Returns

<NRf+>

MINimum|MAXimum|DEFault|<*value*>

15 ARB Subsystem

- ARB:UDEFined:COUNt <NR1>
- ARB:FUNCtion <CPD>
- ARB:UDEFined:LEVel <NR1>,<NRf+>
- ARB:UDEFined:DWELl <NR1>,<NRf+>
- ARB:UDEFined:SLEW <NR1>,<NRf+>
- ARB:COUNt <NRf+>
- ARB:SAVE <NR1>
- ARB:RECall <NR1>

15.1 ARB:UDEFined:COUNt <NR1>

This command is used to set the total number of steps in the LIST. Before setting the amplitude, time width and slope, you must execute this command firstly to set the total number of steps in the LIST.

Group

ARB

Syntax

ARB:UDEFined:COUNt <NR1>

Arguments

<NR1>

MIN|MAX|<value>

Setting range: 1 to 200

Example

ARB:UDEFined:COUNt 6

Query Syntax

ARB:UDEFined:COUNt?

Returns

<NR1>

15.2 ARB:FUNCtion <CPD>

This command is used to set the operation mode of the custom waveform.

Group

ARB

Syntax

ARB:FUNCtion <CPD>

Arguments

<CPD>

VOLTage|CURRent|POWer|RESistance

Default Value

CURRent

Example

ARB:FUNCtion CURRent

Query Syntax

ARB:FUNCtion?

Returns

<CRD>

VOLTage|CURRent|POWer|RESistance

15.3 ARB:UDEFined:LEVel <NR1>,<NRf+>

This command is used to set the amplitude corresponding to the *X* step of the user-defined waveform.

Group

ARB

Syntax

ARB:UDEFined:LEVel <NR1>,<NRf+>

Arguments

- <NR1>

  Used to specify which step of the custom waveform, setting range: 1 to 200.

- <NRf+>

  MIN|MAX|<value>

  Used to specify the voltage/current/power/resistance value of the present step. Setting range: MIN to MAX.

Example

ARB:UDEFined:LEVel 2,10

Query Syntax

ARB:UDEFined:LEVel? <NR1>

Query by specifying a step number to return the amplitude corresponding to the step in the LIST waveform.

Returns

<NRf+>

15.4 ARB:UDEFined:DWELI <NR1>,<NRf+>

This command is used to set the time width corresponding to the *X* step of the user-defined waveform.

Group

ARB

Syntax

ARB:UDEFined:DWELI <NR1>,<NRf+>

Arguments

- <NR1>

Used to specify which step of the custom waveform, setting range: 1 to 200.

- <NRf+>

  MIN|MAX|<value>

  Used to specify the time width of the present step, ranging from MIN to MAX.

Example

ARB:UDEFined:DWELI 2,3

Query Syntax

ARB:UDEFined:DWELI? <NR1>

Query by specifying a step number to return the time width corresponding to the step in the LIST waveform.

Returns

<NRf+>

15.5 ARB:UDEFined:SLEW <NR1>,<NRf+>

This command is used to set the slope corresponding to the *X* step of the user-defined waveform.

Group

ARB

Syntax

ARB:UDEFined:SLEW <NR1>,<NRf+>

Arguments

- <NR1>

  Used to specify which step of the custom waveform, setting range: 1 to 200.

- <NRf+>

  MIN|MAX|DEF|<value>

  Used to specify the slope of the present step, in the range: MIN to MAX.

Default Value

1,MIN

Example

ARB:UDEFined:SLEW 2,0.5

Query Syntax

ARB:UDEFined:SLEW? <NR1>

Query by specifying a step number to return the slope corresponding to the step in the LIST waveform.

Returns

<NRf+>

15.6 ARB:COUNt <NRf+>

Specifies the number of times the Arb repeats. Use the INFinity parameter (or set as 0) to repeat the Arb continuously.

Group

ARB

Syntax

ARB:COUNt <NRf+>

Arguments

<NRf+>

MIN|MAX|<value>

Range: 0~65535

Example

Programs a repeat count of 10: **ARB:COUNt 10**

Query Syntax

ARB:COUNt? [MIN|MAX]

Returns

<NRf+>

15.7 ARB:SAVE <NR1>

This command is used to set the save address of the ARB waveform file.

Group

ARB

Syntax

ARB:SAVE <NR1>

Arguments

<NR1>

The setting range is from 1 to 10.

Example

ARB:SAVE 2

Query Syntax

None

Returns

None

15.8 ARB:RECall <NR1>

This command is used to recall the ARB waveform file that has been saved at an address and wait for the trigger to run.

Group

ARB

Syntax

ARB:RECall <NR1>

Arguments

<NR1>

The setting range is from 1 to 10.

Example

ARB:RECall 2

Query Syntax

None

Returns

None

16 IEEE-488 Common Commands

IEEE-488 Common commands generally control overall instrument functions, such as reset, status, and synchronization. All common commands consist of a three-letter mnemonic preceded by an asterisk: ***RST *IDN? *SRE 8**.

*CLS

Clear Status Command. Clears the event registers in all register groups. Also clears the status byte and error queue. If *CLS immediately follows a program message terminator (<NL>), then the output queue and the MAV bit are also cleared.

Group

None

Syntax

***CLS**

Arguments

None

Default Value

None

Returns

None

Example

***CLS**

Also see

None

*ESE <NR1>

Event status enable command. Sets the value in the enable register for the Standard Event Status group. Each set bit of the register enables a corresponding event. All enabled events are logically ORed into the ESB bit of the status byte.

Note

- Any or all conditions can be reported to the ESB bit through the enable register. To set the enable register mask, write a decimal value to the register using *ESE.
- *CLS does not clear the enable register, but does clear the event register.

Group

None

Syntax

***ESE <NR1>**

Arguments

<NR1>

A decimal value corresponding to the binary-weighted sum of the register's bits. The setting range is from 0 to 255.

Default Value

0

Returns

None

Example

Enable bits 3 and 4 in the enable register: ***ESE 24**

Also see

***ESE?**

*ESE?

Query the value of the enable register set by the standard event status group. The value returned is the binary-weighted sum of all enabled bits in the register. For example, with bit 2 (value 4) and bit 4 (value 16) set, the query returns +20.

Group

None

Syntax

***ESE?**

Arguments

None

Default Value

None

Returns

<NR1>

A decimal value corresponding to the binary-weighted sum of the register's bits. The range is from 0 to 255.

Example

***ESE?**

Also see

None

*ESR?

Event status event query. Reads and clears the event register for the Standard Event Status group. The event register is a read-only register, which latches all standard events.

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

- Any or all conditions can be reported to the ESB bit through the enable register. To set the enable register mask, write a decimal value to the register using *ESE.
- Once a bit is set, it remains set until cleared by this query or *CLS.

Group

None

Syntax

***ESR?**

Arguments

None

Default Value

None

Returns

<NR1>

Example

***ESR?**

Also see

None

*IDN?

Identification Query. Returns instrument’s identification string, which contains four comma-separated fields. The first field is the manufacturer's name, the second field is the instrument model number, the third field is the serial number, and the fourth field is the firmware revision.

Group

None

Syntax

***IDN?**

Arguments

None

Default Value

None

Returns

<AARD>

Example

***IDN?**

Also see

None

*OPC

Sets the OPC (operation complete) bit in the standard event register. This occurs at the completion of the pending operation.

- The purpose of this command is to synchronize your application with the instrument.
- Used in conjunction with initiated acquisitions, transients, output state changes, and output settling time to provide a way to poll or interrupt the computer when these pending operations complete.
- Other commands may be executed before the operation complete bit is set.

Group

None

Syntax

***OPC**

Arguments

None

Default Value

None

Returns

None

Example

***OPC**

Also see

***OPC?**

*OPC?

Returns a 1 to the output buffer when all pending operations complete. The response is delayed until all pending operations complete.

- The purpose of this command is to synchronize your application with the instrument.
- Other commands cannot be executed until this command completes.

Group

None

Syntax

***OPC?**

Arguments

None

Default Value

None

Returns

<NR1>

Return a 1 when commands complete.

Example

***OPC?**

Also see

None

*RST

Resets the instrument to pre-defined values that are either typical or safe. The following table shows the reset state. These parameters are reset to the indicated values at power-on or after ***RST**.

SCPI Commands	*RST Initial Settings
ARB:COUNt	1
ARB:CURRent:CDWell:DWELl	0.001
ARB:FUNCtion:SHAPe	CDW
ARB:FUNCtion:TYPE	VOLTage
ARB:TERMinate:LAST	OFF
ARB:VOLTage:CDWell:DWELl	0.001
CALibrate:STATe	OFF
CURRent	0
CURRent:LIMit	1% of rating
CURRent:LIMit:NEGative	-1% of rating
CURRent:MODE	FIXed
CURRent:PROTection:DELay	20ms
CURRent:PROTection:STATe	OFF
CURRent:SHARing	OFF
CURRent:SLEW	MAX
CURRent:SLEW:MAXimum	ON

SCPI Commands	*RST Initial Settings
CURRent:TRIGgered	0
FUNCtion	VOLTage
INITialize:CONTinuous:TRANsient	OFF
OUTPut	OFF
OUTPut:DELay:FALL	0
OUTPut:DELay:RISE	0
OUTPut:PROTection:WDOG	OFF
OUTPut:PROTection:WDOG:DELay	60
RESistance	0
RESistance:STATe	0
TRIGger:ACQuire:CURRent	0
TRIGger:ACQuire:CURRent:SLOPe	POSitive
TRIGger:ACQuire:SOURce	BUS
TRIGger:ACQuire:TOUTput	OFF
TRIGger:ACQuire:VOLTage	0
TRIGger:ACQuire:VOLTage:SLOPe	POSitive
TRIGger:ARB:SOURce	BUS
TRIGger:TRANsient:SOURce	BUS
VOLTage	1% of rating
VOLTage:LIMit	1% of rating
VOLTage:MODE	FIXed
VOLTage:PROTection	120% of rating
VOLTage:RESistance	0
VOLTage:RESistance:STATe	OFF
VOLTage:SLEW	MAX
VOLTage:SLEW:MAXimum	ON

Group

None

Syntax

***RST**

Arguments

None

Default Value

None

Returns

None

Example

***RST**

Also see

None

*SRE <NR1>

Service request enable command. This sets the value of the Service Request Enable register. This determines which bits from the Status Byte Register are summed to set the Master Status Summary (MSS) bit and the Request for Service (RQS) summary bit. A 1 in any Service Request Enable register bit position enables the corresponding Status Byte register bit. All such enabled bits are then logically OR-ed to cause the MSS bit of the Status Byte register to be set.

Group

None

Syntax

***SRE <NR1>**

Arguments

<NR1>

A decimal value corresponding to the binary-weighted sum of the register's bits. The setting range is from 0 to 255.

Default Value

0

Returns

None

Example

Enable bit 3 and bit 4 in the enable register: ***SRE 24**

Also see

***SRE?**

*SRE?

Query the value of the service request enable register.

Group

None

Syntax

***SRE?**

Arguments

None

Default Value

None

Returns

<NR1>

A decimal value corresponding to the binary-weighted sum of the register's bits. The range is from 0 to 255.

Example

***SRE?**

Also see

None

*STB?

Status byte query. Reads the Status Byte Register, which contains the status summary bits and the Output Queue MAV bit. The Status Byte is a read-only register and the bits are not cleared when it is read.

Group

None

Syntax

***STB?**

Arguments

None

Default Value

None

Returns

<NR1>

Example

***STB?**

Also see

None

*TRG

Trigger command. Generates a trigger when the trigger subsystem has BUS selected as its source.

Group

None

Syntax

***TRG**

Arguments

None

Default Value

None

Returns

None

Example

***TRG**

Also see

None

*SAV <NR1>

Save the instrument **[On/Off]** status or several parameter settings to 10 non-volatile memories, and the position can be set from 1 to 10. When shipped, locations 1 through 10 are empty.

Group

None

Syntax

***SAV <NR1>**

Arguments

<NR1>

The setting range is from 1 to 10.

Default Value

None

Returns

None

Example

***SAV 2**

Also see

***RCL <NR1>**

*RCL <NR1>

Recalls a saved instrument state. This restores the instrument to a state that was previously stored in locations 0 through 9 with the ***SAV** command.

Group

None

Syntax

***RCL <NR1>**

Arguments

<NR1>

The setting range is from 1 to 10.

Default Value

None

Returns

None

Example

***RCL 1**

Also see

***SAV <NR1>**

*TST?

Self-test query. Performs a instrument self-test. If self-test fails, one or more error messages will provide additional information. Use **SYSTem:ERRor?** to read error queue. For details, see 17 Error Messages.

Group

None

Syntax

***TST?**

Arguments

None

Default Value

None

Returns

<NR1>

0 (pass) or +1 (failed)

Example

***TST?**

Also see

None

*WAI

Pauses additional command processing until all pending operations are complete.

Group

None

Syntax

***WAI**

Arguments

None

Default Value

None

Returns

None

Example

***WAI**

Also see

None

*PSC <Bool>

This instruction is used to control whether the status register is cleared when the instrument is powered-on. This instruction affects the value of the status register at the next powered-on.

Group

None

Syntax

***PSC <Bool>**

Arguments

<Bool>

0|OFF|1|ON

Default Value

0|OFF

Returns

None

Example

***PSC 1**

Also see

***PSC?**

*PSC?

This instruction is used to query whether the status register is cleared when the instrument is powered-on.

Group

None

Syntax

***PSC?**

Arguments

None

Default Value

None

Returns

<Bool>

0|OFF|1|ON

Example

***PSC?**

Also see

None

17 Error Messages

If the instrument generates a fault during communication with the host computer or executes a programming command that is not supported by the instrument, the word **Error** will be displayed on the front panel VFD. At the same time, the user can send the **SYSTem:ERRor?** command through the host computer, and the instrument will return the error code and the corresponding error message.

The detailed error code and description information are as follows.

Error Code	Error Message	Description
0	No error	This is the response to the ERR? query when there are no errors.
Parameter setting related errors		
101	DESIGN ERROR: Too many numeric suffices in Command Spec	The number of numeric parameters of the command exceeds the limit.
110	No Input Command to parse	No command input (for example, the error code is reported when an empty command is sent).
114	Numeric suffix is invalid value	The number sent to the command does not match the number specified in the command specification.
116	Invalid value in numeric or channel list, e.g. out of range	The value or list parameter is invalid, such as out of range.
117	Invalid number of dimensions in a channel list	Invalid value in parameter list
120	Parameter of type Numeric Value overflowed its storage	Digital parameter overflow, for example, the parameter setting value is not within the settable range.
130	Wrong units for parameter	The unit of the parameter is incorrect.

Error Code	Error Message	Description
140	Wrong type of parameter(s)	The type of the parameter is incorrect.
150	Wrong number of parameters	The number of parameters is incorrect.
160	Unmatched quotation mark (single/double) in parameters	The quotation marks (single/ double) in the argument do not match.
165	Unmatched bracket	The parentheses do not match.
170	Command keywords were not recognized	The command keyword is not recognized, which is an invalid command.
180	No entry in list to retrieve (number list or channel list)	The function entry is incorrect. For example, three parameters are sent to the command, and the system program only processes two.
190	Too many dimensions in entry to be returned in parameters	There are too many values returned in the argument.
191	Too many char	The number of characters exceeds the limit. For example, when using serial port or USB communication, the transmitted data length is greater than 256.
-150	String data error	The string data is incorrect.
-151	Invalid string data [e.g., END received before close quote]	Invalid string data. For example, the END flag was received before the command reference was closed.
-158	String data not allowed	The data type is not allowed as a string.
-160	Block data error	Block data is incorrect.
-161	Invalid block data [e.g., END received before length satisfied]	Invalid block data. For example, the END flag was received before the data length met the requirements.

Error Code	Error Message	Description
-168	Block data not allowed	The data type is not allowed to be block data.
-170	Expression error	The expression is incorrect.
-171	Invalid expression	Invalid expression.
-178	Expression data not allowed	Expression data is not allowed.
command execution related errors		
-200	Execution error [generic]	If the status or setting of the command is incorrect, the error will be reported. For example, ***TRG** must be executed when the LIST trigger source is set to BUS, so executing ***TRG** will prompt the error when the LIST trigger source is set to Manual.
-221	Settings conflict [check current device state]	A data element could not be executed because of the present instrument state.
-222	Data out of range [e.g., too large for this device]	A data element could not be executed because the value was outside the valid range.
-223	Too much data [out of memory; block, string, or expression too long]	A data element was received that contains more data than the instrument can handle.
-224	Illegal parameter value [device-specific]	An exact value was expected but not received.
-225	Out of memory	The device has insufficient memory to perform the requested operation.
-230	Data Corrupt or Stale	Possible invalid data. A new reading was started but not completed.
-270	Macro error	Macro definition error.
-272	Macro execution error	Macro execution error
-273	Illegal macro label	Illegal macro tag

Error Code	Error Message	Description
-276	Macro recursion error	Macro recursion error
-277	Macro redefinition not allowed	Redefinition macros are not allowed
System errors		
-310	System error [generic]	System error (general)
-350	Too many errors [errors beyond 9 lost due to queue overflow]	Too many errors (more than 9 error messages were lost due to queue overflow).
Query errors		
-499	sets Standard Event Status Register bit #2	When a query error occurs, the second bit of the standard status register is set to 1.
-400	Query error [generic]	Generic error query
-410	Query INTERRUPTED [query followed by DAB or GET before response complete]	A condition causing an interrupted query error occurred.
-430	Query DEADLOCKED [too many queries in command string]	A condition causing a deadlocked query error occurred.
-440	Query UNTERMINATED [after indefinite response]	A query was received in the same program message after a query indicating an indefinite response was executed.
Self-test errors		
1	Module Initialization Lost	Module initialization lost
2	Mainframe Initialization Lost	Host initialization lost
3	Module Calibration Lost	Module calibration data is lost
4	Non-volatile RAM STATE section checksum failed	Non-volatile RAM area verification failed
5	Non-volatile RAM RST section checksum failed	Non-volatile RAM area reset failed
10	RAM selftest	RAM self-test error

Error Code	Error Message	Description
11	CVDAC selftest 1	In CV mode, DAC channel 1 self-test failed.
12	CVDAC selftest 2	In CV mode, DAC channel 2 self-test failed.
13	CCDAC selftest 1	In CC mode, DAC channel 1 self-test failed.
14	CCDAC selftest 2	In CC mode, DAC channel 2 self-test failed.
15	CRDAC selftest 1	In CR mode, DAC channel 1 self-test failed.
16	CRDAC selftest 2	In CR mode, DAC channel 2 self-test failed.
20	Input Down	Input drop
40	Flash write failed	Writing to flash failed.
41	Flash erase failed	Failed to erase the flash.
80	Digital I/O selftest error	Digital I/O self-test error
Device related errors		
213	RS232 buffer overrun error	RS232 buffer overflow
216	RS232 receiver framing error	RS232 receiver frame error
217	RS232 receiver parity error	RS232 receiver parity error
218	RS232 receiver overrun error	RS232 receiver overflow
220	Front panel uart overrun	The serial port communication between the front panel and the control board overflows.
221	Front panel uart framing	The serial port frame between the front panel and the control board is incorrect.
222	Front panel uart parity	The serial port parity between the front panel and the control board is incorrect.
223	Front panel buffer overrun	The serial port buffer between the front panel and the control board overflows.

Error Code	Error Message	Description
224	Front panel timeout	The serial port connection between the front panel and the control board has timed out.
225	Front Crc Check error	The CRC check of the serial port between the front panel and the control board is incorrect.
226	Front Cmd Error	The serial port command between the front panel and the control board is incorrectly used.
401	CAL switch prevents calibration	Calibration is disabled due to the status setting of the calibration switch.
402	CAL password is incorrect	The calibration password is incorrect.
403	CAL not enabled	Calibration is not enabled.
404	Computed readback cal constants are incorrect	An error occurred in reading the calibration data.
405	Computed programming cal constants are incorrect	An error occurred in calculating the calibration data.
406	Incorrect sequence of calibration commands	The execution order of the calibration commands is incorrect.
407	CV or CC status is incorrect for this command	The CC or CV status of this command is incorrect.
408	Output mode switch must be in NORMAL position	The switch for the output mode must be NORMAL.
600	Lists inconsistent [lists have different list lengths]	The lists are inconsistent, such as a different list length.
601	Too many sweep points	There are too many sweep points.
602	Command only applies to RS232 interface	This command only applies to the RS232 interface.
603	FETCH of data that was not acquired	The data related to the FETCH command is not acquired.

Error Code	Error Message	Description
604	Measurement overrange	Out of measurement range
605	Command not allowed while list initiated	It is forbidden to execute this command while the List is initiated.
610	Corrupt update data	The system update data is corrupt.
611	Not Updating	System not updated

Connect with us

Thank you for purchasing ITECH products. Any questions, pls. feel free to let us know.

chat

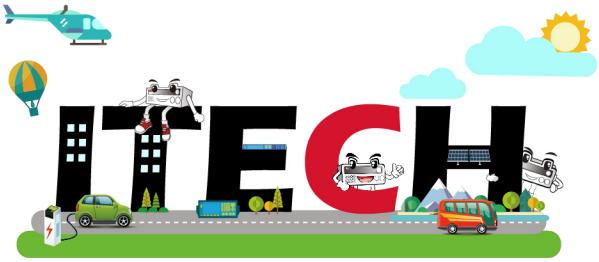
You can chat with us on ITECH website

For more information, pls. visit www.itechate.com

ITECH Web

ITECH News

Click and follow ITECH Electronics



Facebook LinkedIn YouTube