

High Power Programmable DC Power Supply

IT6600D Series Programming Guide



Notices

© Itech Electronic, Co., Ltd. 2025

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior permission and written consent from Itech Electronic, Co., Ltd. as governed by international copyright laws.

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).



Note

Visit <https://www.itechate.com/en/support/register.html> to complete product registration by filling out the necessary information to extend the warranty to two (2) years.

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 PRE-PAID. 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 output the dangerous voltage that can cause personal injury, and the operator must always be protected from electric shock. Ensure that the output 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

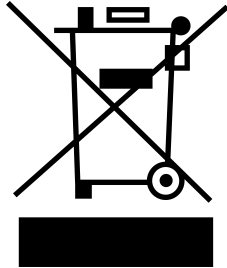

Note

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

Regulation Tag

	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).
	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.
	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 / EN 61326-1 ¹²³

Reference Standards

CISPR 11, Class A

IEC 61000-3-2

IEC 61000-3-3

IEC 61000-4-2

IEC 61000-4-3

IEC 61000-4-4

IEC 61000-4-5

IEC 61000-4-6

IEC 61000-4-11

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



Note

This section provides general and universal terms. For more detailed information on the *Declaration of Conformity*, please contact ITECH personnel.

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	9
1.7 Remote Interface Connections	11
1.8 Multi-channel @chanlist Introduction	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
SYSTem:VOLTage:RZERo <CPD>[, (@chanlist)]	34
SYSTem:ANTI:ACCEss? [(@chanlist)]	34
3 [SOURce] Subsystem	36
[SOURce:]FUNctioN <CPD>[, (@chanlist)]	36

[SOURCE:]FUNCTION? [(@chanlist)]	37
[SOURCE:]FUNCTION:MODE <CPD>[, (@chanlist)]	37
[SOURCE:]FUNCTION:MODE? [(@chanlist)]	38
[SOURCE:]FUNCTION:PRIORITY[:SLEW]:TYPE <CPD>[, (@chanlist)]	39
[SOURCE:]CURRENT[:LEVEL][:IMMEDIATE][:AMPLITUDE] <NRf+>[, (@chanlist)]	40
[SOURCE:]CURRENT[:LEVEL][:IMMEDIATE][:AMPLITUDE]? [MINIMUM MAXIMUM DEFAULT] [, (@chanlist)]	41
[SOURCE:]CURRENT:SLEW[:BOTH] <NRf+>[, (@chanlist)]	41
[SOURCE:]CURRENT:SLEW[:BOTH]? [MINIMUM MAXIMUM DEFAULT][, (@chanlist)]	42
[SOURCE:]CURRENT:SLEW:POSITIVE <NRf+>[, (@chanlist)]	43
[SOURCE:]CURRENT:SLEW:POSITIVE? [MINIMUM MAXIMUM DEFAULT][, (@chanlist)]	44
[SOURCE:]CURRENT:SLEW:NEGATIVE <NRf+>[, (@chanlist)]	45
[SOURCE:]CURRENT:SLEW:NEGATIVE? [MINIMUM MAXIMUM DEFAULT][, (@chanlist)]	46
[SOURCE:]CURRENT[:OVER]:PROTECTION:STATE <Bool>[, (@chanlist)]	46
[SOURCE:]CURRENT[:OVER]:PROTECTION:STATE? [(@chanlist)]	47
[SOURCE:]CURRENT[:OVER]:PROTECTION[:LEVEL] <NRf+>[, (@chanlist)]	48
[SOURCE:]CURRENT[:OVER]:PROTECTION[:LEVEL]? [(@chanlist)]	49
[SOURCE:]CURRENT[:OVER]:PROTECTION:DELAY[:TIME] <NRf+>[, (@chanlist)]	50
[SOURCE:]CURRENT[:OVER]:PROTECTION:DELAY[:TIME]? [MINIMUM MAXIMUM DE- Fault][, (@chanlist)]	50
[SOURCE:]CURRENT:UNDER:PROTECTION:STATE <Bool>[, (@chanlist)]	51
[SOURCE:]CURRENT:UNDER:PROTECTION:STATE? [(@chanlist)]	52
[SOURCE:]CURRENT:UNDER:PROTECTION[:LEVEL] <NRf+>[, (@chanlist)]	53
[SOURCE:]CURRENT:UNDER:PROTECTION[:LEVEL]? [(@chanlist)]	54
[SOURCE:]CURRENT:UNDER:PROTECTION:DELAY[:TIME] <NRf+>[, (@chanlist)]	54
[SOURCE:]CURRENT:UNDER:PROTECTION:DELAY[:TIME]? [MINIMUM MAXIMUM DE- Fault][, (@chanlist)]	55
[SOURCE:]CURRENT:UNDER:PROTECTION:WARM[:TIME] <NRf+>[, (@chanlist)]	56
[SOURCE:]CURRENT:UNDER:PROTECTION:WARM[:TIME]? [MINIMUM MAXIMUM DE- Fault][, (@chanlist)]	57
[SOURCE:]CURRENT:LIMIT[:POSITIVE][:IMMEDIATE][:AMPLITUDE] <NRf+>[, (@chanlist)]	58
[SOURCE:]CURRENT:LIMIT[:POSITIVE][:IMMEDIATE][:AMPLITUDE]? [MINIMUM MAXIMUM DEFAULT][, (@chanlist)]	59
[SOURCE:]VOLTAGE[:LEVEL][:IMMEDIATE][:AMPLITUDE] <NRf+>[, (@chanlist)]	60
[SOURCE:]VOLTAGE[:LEVEL][:IMMEDIATE][:AMPLITUDE]? [MINIMUM MAXIMUM DEFAULT] [, (@chanlist)]	60
[SOURCE:]VOLTAGE:SLEW[:BOTH] <NRf+>[, (@chanlist)]	61
[SOURCE:]VOLTAGE:SLEW[:BOTH]? [MINIMUM MAXIMUM DEFAULT][, (@chanlist)]	62
[SOURCE:]VOLTAGE:SLEW:POSITIVE <NRf+>[, (@chanlist)]	63
[SOURCE:]VOLTAGE:SLEW:POSITIVE? [MINIMUM MAXIMUM DEFAULT][, (@chanlist)]	64
[SOURCE:]VOLTAGE:SLEW:NEGATIVE <NRf+>[, (@chanlist)]	64
[SOURCE:]VOLTAGE:SLEW:NEGATIVE? [MINIMUM MAXIMUM DEFAULT][, (@chanlist)]	65
[SOURCE:]VOLTAGE[:OVER]:PROTECTION:STATE <Bool>[, (@chanlist)]	66
[SOURCE:]VOLTAGE[:OVER]:PROTECTION:STATE? [(@chanlist)]	67
[SOURCE:]VOLTAGE[:OVER]:PROTECTION[:LEVEL] <NRf+>[, (@chanlist)]	68
[SOURCE:]VOLTAGE[:OVER]:PROTECTION[:LEVEL]? [(@chanlist)]	68
[SOURCE:]VOLTAGE[:OVER]:PROTECTION:DELAY[:TIME] <NRf+>[, (@chanlist)]	69

[SOURCE:]VOLTage[:OVER]:PROTection:DELAy[:TIME]? [MINimum MAXimum DE-Fault][, (@chanlist)]	70
[SOURCE:]VOLTage:UNDer:PROTection:STATe <Bool>[, (@chanlist)]	71
[SOURCE:]VOLTage:UNDer:PROTection:STATe? [(@chanlist)]	72
[SOURCE:]VOLTage:UNDer:PROTection[:LEVel] <NRf+>[, (@chanlist)]	72
[SOURCE:]VOLTage:UNDer:PROTection[:LEVel]? [(@chanlist)]	73
[SOURCE:]VOLTage:UNDer:PROTection:DELAy[:TIME] <NRf+>[, (@chanlist)]	74
[SOURCE:]VOLTage:UNDer:PROTection:DELAy[:TIME]? [MINimum MAXimum DE-Fault][, (@chanlist)]	75
[SOURCE:]VOLTage:UNDer:PROTection:WARM[:TIME] <NRf+>[, (@chanlist)]	76
[SOURCE:]VOLTage:UNDer:PROTection:WARM[:TIME]? [MINimum MAXimum DE-Fault][, (@chanlist)]	77
[SOURCE:]VOLTage:LIMit[:POSitive][:IMMediate][:AMPLitude] <NRf+>[, (@chanlist)]	77
[SOURCE:]VOLTage:LIMit[:POSitive][:IMMediate][:AMPLitude]? [MINimum MAXimum DEFAult][, (@chanlist)]	78
[SOURCE:]POWEr:LIMit[:POSitive][:IMMediate][:AMPLitude] <NRf+>[, (@chanlist)]	79
[SOURCE:]POWEr:LIMit[:POSitive][:IMMediate][:AMPLitude]? [MINimum MAXimum DE-Fault][, (@chanlist)]	80
[SOURCE:]POWEr[:OVER]:PROTection:STATe <Bool>[, (@chanlist)]	81
[SOURCE:]POWEr[:OVER]:PROTection:STATe? [(@chanlist)]	82
[SOURCE:]POWEr[:OVER]:PROTection[:LEVel] <NRf+>[, (@chanlist)]	82
[SOURCE:]POWEr[:OVER]:PROTection[:LEVel]? [(@chanlist)]	83
[SOURCE:]POWEr[:OVER]:PROTection:DELAy[:TIME] <NRf+>[, (@chanlist)]	84
[SOURCE:]POWEr[:OVER]:PROTection:DELAy[:TIME]? [MINimum MAXimum DEFAult][, (@chanlist)]	85
[SOURCE:]RESistance[:LEVel][:IMMediate][:AMPLitude] <NRf+>[, (@chanlist)]	86
[SOURCE:]RESistance[:LEVel][:IMMediate][:AMPLitude]? [MINimum MAXimum DE-Fault][, (@chanlist)]	87
[SOURCE:]APPLy <volt>, <curr>[, (@chanlist)]	87
[SOURCE:]EXTErn:PROGrama[:STATe] <Bool>[, (@chanlist)]	88
[SOURCE:]EXTErn:PROGrama[:STATe]? [(@chanlist)]	89
[SOURCE:]EXTErn:PROGrama:CHANnel:MX <NR1>, <NRf>[, (@chanlist)]	90
[SOURCE:]EXTErn:PROGrama:CHANnel:MX? <NR1>[, (@chanlist)]	91
[SOURCE:]EXTErn:PROGrama:CHANnel:MB <NR1>, <NRf>[, (@chanlist)]	91
[SOURCE:]EXTErn:PROGrama:CHANnel:MB? <NR1>[, (@chanlist)]	92
[SOURCE:]FILTEr:LEVel <CPD>[, (@chanlist)]	93
[SOURCE:]FILTEr:LEVel? [(@chanlist)]	94
[SOURCE:]CV:PRiority <priority>[, (@chanlist)]	95
[SOURCE:]CC:PRiority <priority>[, (@chanlist)]	95
[SOURCE:]PROTection:CLEar [(@chanlist)]	96
4 OUTPUT Subsystem	98
OUTPut[:STATe] <Bool>[, (@chanlist)]	98
OUTPut[:STATe]? [(@chanlist)]	99
OUTPut:DELAy:FALL <NRf+>[, (@chanlist)]	99
OUTPut:DELAy:FALL? [MINimum MAXimum DEFAult][, (@chanlist)]	100
OUTPut:DELAy:RISE <NRf+>[, (@chanlist)]	101
OUTPut:DELAy:RISE? [MINimum MAXimum DEFAult][, (@chanlist)]	102
OUTPut:PON:STATe <CPD>	102
OUTPut:PROTection:WDOG[:STATe] <CPD>[, (@chanlist)]	104
OUTPut:PROTection:WDOG:DELAy <NRf+>[, (@chanlist)]	104
OUTPut:MODE <CPD>	105
OUTPut:COUPlE:SYNChronous[:STATe] <Bool>	106
OUTPut:COUPlE:MODE <CPD>	107
OUTPut:COUPlE:SYNChronous:REFerence <NRf+>	108

	OUTPut[:STATe]:COUPlE:DOFFset <NRf+>[,(@chanlist)]	108
5	ABORt Subsystem	110
	ABORt:ACQuire [(@chanlist)]	110
	ABORt:ARB [(@chanlist)]	110
6	INITiate Subsystem	112
	INITiate[:IMMediate]:ACQuire[(@chanlist)]	112
	INITiate:CONTInuous:ARB <Bool>[,(@chanlist)]	113
	INITiate:CONTInuous:ACQuire <Bool>[,(@chanlist)]	114
	INITiate[:IMMediate]:DLOG[(@chanlist)]	114
	INITiate[:IMMediate]:ELOG[(@chanlist)]	115
7	BATTery Subsystem	117
	BATTery:CHARge:VOLTagE <NRf+>[,(@chanlist)]	117
	BATTery:CHARge:VOLTagE? [MINimum MAXimum DEFault][,(@chanlist)]	118
	BATTery:CHARge:CURRent <NRf+>[,(@chanlist)]	118
	BATTery:CHARge:CURRent? [MINimum MAXimum DEFault][,(@chanlist)]	119
	BATTery:SHUT:VOLTagE <NRf+>[,(@chanlist)]	120
	BATTery:SHUT:VOLTagE? [MINimum MAXimum DEFault][,(@chanlist)]	121
	BATTery:SHUT:CURRent <NRf+>[,(@chanlist)]	122
	BATTery:SHUT:CURRent? [MINimum MAXimum DEFault][,(@chanlist)]	122
	BATTery:SHUT:CAPacity <NRf+>[,(@chanlist)]	123
	BATTery:SHUT:CAPacity? [MINimum MAXimum DEFault][,(@chanlist)]	124
	BATTery:SHUT:ENERgy <NRf+>[,(@chanlist)]	125
	BATTery:SHUT:TIME <NRf+>[,(@chanlist)]	126
	BATTery:SHUT:TIME? [MINimum MAXimum DEFault][,(@chanlist)]	126
	BATTery:RUN:STATe <Bool>[,(@chanlist)]	127
	BATTery:SAVe <filename>[,(@chanlist)]	128
	BATTery:RECall <filename>[,(@chanlist)]	129
	BATTery:DELeTe <filename>[,(@chanlist)]	129
	BATTery:TRIG [(@chanlist)]	130
8	CONFIgurable Subsystem	131
	IO:CHANnel:SELEct <port>,<chl>	131
	IO:SELEct <NR1>	131
	IO:SELEct?	132
	IO:DIREction <NRL>,<Bool>	133
	IO:DIREction? <NRL>	134
	IO:REVERse <NRL>,<Bool>	135
	IO:REVERse? <NRL>	136
	IO:PWM[:ENABle] <NRL>,<Bool>	137
	IO:PWM[:ENABle]? <NRL>	138
	IO:PWM:FREQUency <NRL>,<NRf+>	138
	IO:PWM:FREQUency? <NRL>[,][MINimum MAXimum DEFault]	139
	IO:PWM:DUTY <NRL>,<NR1>	140
	IO:PWM:DUTY? <NRL>	141
	IO:PULSe:WIDTh <NRL>,<NRf+>	141
	IO:PULSe:WIDTh? <NRL>[,][MINimum MAXimum DEFault]	142
	IO:TYPE <NRL>,<CPD>	143
	IO:TYPE? <NRL>	144
	IO:OUTPut:LEVel <NRL>,<CPD>	145
	IO:OUTPut:LEVel? <NRL>	146
	IO:INPut:LEVel? <NRL>	147
	[SOURce:]DIGital:RElay:TYPE <NR1>,<CPD>	148
	[SOURce:]DIGital:RElay:LEVel <NR1>,<Bool>	149
9	TRIGger Subsystem	150
	TRIGger:ACQuire[:IMMediate] [(@chanlist)]	150
	TRIGger:ACQuire:MODE <CPD>[,(@chanlist)]	150
	TRIGger:ACQuire:SOURce <CPD>[,(@chanlist)]	151
	TRIGger:ACQuire:VOLTagE:SLOPe <CPD>[,(@chanlist)]	153

TRIGger:ACQuire:VOLTagE[:LEVel] <NRf+>[,(@chanlist)]	154
TRIGger:ACQuire:VOLTagE:HYSterEsis:HIGH <NRf+>[,(@chanlist)]	154
TRIGger:ACQuire:VOLTagE:HYSterEsis:LOW <NRf+>[,(@chanlist)]	155
TRIGger:ACQuire:CURRent:SLOPe <CPD>[,(@chanlist)]	156
TRIGger:ACQuire:CURRent[:LEVel] <NRf+>[,(@chanlist)]	157
TRIGger:ACQuire:CURRent:HYSterEsis:HIGH <NRf+>[,(@chanlist)]	158
TRIGger:ACQuire:CURRent:HYSterEsis:LOW <NRf+>[,(@chanlist)]	159
TRIGger:ARB[:IMMediate] [(@chanlist)]	160
TRIGger:ARB:SOURce <CPD>[,(@chanlist)]	160
TRIGger:DLOG:SOURce <CPD>[,(@chanlist)]	162
TRIGger:DLOG[:IMMediate] [(@chanlist)]	163
TRIGger:DLOG:CURRent:SLOPe <CPD>[,(@chanlist)]	163
TRIGger:DLOG:CURRent[:LEVel] <NRf+>[,(@chanlist)]	164
TRIGger:DLOG:CURRent:HYSterEsis:HIGH <NRf+>[,(@chanlist)]	165
TRIGger:DLOG:CURRent:HYSterEsis:LOW <NRf+>[,(@chanlist)]	166
TRIGger:DLOG:VOLTagE:SLOPe <CPD>[,(@chanlist)]	167
TRIGger:DLOG:VOLTagE[:LEVel] <NRf+>[,(@chanlist)]	168
TRIGger:DLOG:VOLTagE:HYSterEsis:HIGH <NRf+>[,(@chanlist)]	169
TRIGger:DLOG:VOLTagE:HYSterEsis:LOW <NRf+>[,(@chanlist)]	170
TRIGger:ELOG:SOURce <CPD>[,(@chanlist)]	170
TRIGger:ELOG[:IMMediate] [(@chanlist)]	172
TRIGger:ELOG:CURRent:SLOPe <CPD>[,(@chanlist)]	172
TRIGger:ELOG:CURRent[:LEVel] <NRf+>[,(@chanlist)]	173
TRIGger:ELOG:CURRent:HYSterEsis:HIGH <NRf+>[,(@chanlist)]	174
TRIGger:ELOG:CURRent:HYSterEsis:LOW <NRf+>[,(@chanlist)]	175
TRIGger:ELOG:VOLTagE:SLOPe <CPD>[,(@chanlist)]	176
TRIGger:ELOG:VOLTagE[:LEVel] <NRf+>[,(@chanlist)]	177
TRIGger:ELOG:VOLTagE:HYSterEsis:HIGH <NRf+>[,(@chanlist)]	178
TRIGger:ELOG:VOLTagE:HYSterEsis:LOW <NRf+>[,(@chanlist)]	178
10 STATus Subsystem	180
STATus:QUEStionable[:EVENT]? [(@chanlist)]	183
STATus:QUEStionable:ENABle <NR1>[,(@chanlist)]	184
STATus:QUEStionable:PTRansition <NR1>[,(@chanlist)]	185
STATus:QUEStionable:NTRansition <NR1>[,(@chanlist)]	186
STATus:QUEStionable:CONDition? [(@chanlist)]	187
STATus:OPERation[:EVENT]? [(@chanlist)]	188
STATus:OPERation:ENABle <NR1>[,(@chanlist)]	189
STATus:OPERation:PTRansition <NR1>[,(@chanlist)]	190
STATus:OPERation:NTRansition <NR1>[,(@chanlist)]	191
STATus:OPERation:CONDition? [(@chanlist)]	192
STATus:PRESet [(@chanlist)]	193
11 FETCh Subsystem	195
FETCh[:SCALar]:CURRent[:DC]? [(@chanlist)]	195
FETCh[:SCALar]:CURRent[:DC]:HIGH? [(@chanlist)]	196
FETCh[:SCALar]:CURRent[:DC]:LOW? [(@chanlist)]	196
FETCh[:SCALar]:CURRent[:DC]:MAXimum? [(@chanlist)]	197
FETCh[:SCALar]:CURRent[:DC]:MINimum? [(@chanlist)]	198
FETCh[:SCALar]:VOLTagE[:DC]? [(@chanlist)]	199
FETCh[:SCALar]:VOLTagE[:DC]:HIGH? [(@chanlist)]	199
FETCh[:SCALar]:VOLTagE[:DC]:LOW? [(@chanlist)]	200
FETCh[:SCALar]:VOLTagE[:DC]:MAXimum? [(@chanlist)]	201
FETCh[:SCALar]:VOLTagE[:DC]:MINimum? [(@chanlist)]	202
FETCh[:SCALar]:POWER[:DC]? [(@chanlist)]	202
FETCh[:SCALar]? [(@chanlist)]	203
ELOG:STATe?	204
DLOG:STATe?	205
FETCh:ELOG:ARRAy:DATA[:DC]? [(@chanlist)]	205

12	MEASure Subsystem	207
	MEASure[:SCALar]:CURRent[:DC]? [(@chanlist)]	207
	MEASure[:SCALar]:CURRent:HIGH? [(@chanlist)]	208
	MEASure[:SCALar]:CURRent:LOW? [(@chanlist)]	208
	MEASure[:SCALar]:CURRent:MAXimum? [(@chanlist)]	209
	MEASure[:SCALar]:CURRent:MINimum? [(@chanlist)]	210
	MEASure[:SCALar]:VOLTage[:DC]? [(@chanlist)]	211
	MEASure[:SCALar]:VOLTage:HIGH? [(@chanlist)]	211
	MEASure[:SCALar]:VOLTage:LOW? [(@chanlist)]	212
	MEASure[:SCALar]:VOLTage:MAXimum? [(@chanlist)]	213
	MEASure[:SCALar]:VOLTage:MINimum? [(@chanlist)]	214
	MEASure[:SCALar]:POWER[:DC]? [(@chanlist)]	214
	MEASure:ARRay:CURRent[:DC]? [(@chanlist)]	215
	MEASure:ARRay:VOLTage[:DC]? [(@chanlist)]	216
	MEASure[:SCALar]? [(@chanlist)]	217
13	PARAllel Subsystem	218
	PARAllel:ROLE <CPD>	218
	PARAllel:NUMBer <NR1>	219
14	SENSe Subsystem	220
	[SOURce:]REMOte:SENSe[:STATe] <Bool>[, (@chanlist)]	220
	SENSe:ELOG:FUNcTION:CURRent <Bool>[, (@chanlist)]	221
	SENSe:ELOG:FUNcTION:VOLTage <Bool>[, (@chanlist)]	222
	SENSe:ELOG:FUNcTION:POWER <Bool>[, (@chanlist)]	222
	SENSe:ELOG:FUNcTION:PEAK <Bool>[, (@chanlist)]	223
	SENSe:ELOG:PERiod <NRf+>[, (@chanlist)]	224
	SENSe:ELOG:TIME <NRf+>[, (@chanlist)]	225
	SENSe:DLOG:FUNcTION:CURRent <Bool>[, (@chanlist)]	226
	SENSe:DLOG:FUNcTION:VOLTage <Bool>[, (@chanlist)]	227
	SENSe:DLOG:FUNcTION:POWER <Bool>[, (@chanlist)]	228
	SENSe:DLOG:FUNcTION:PEAK <Bool>[, (@chanlist)]	229
	SENSe:DLOG:PERiod <NRf+>[, (@chanlist)]	229
	SENSe:DLOG:TIME <NRf+>[, (@chanlist)]	230
	SENSe:FILTer:LEVel <CPD>[, (@chanlist)]	231
	[SOURce:]REMOte:SENSe:THReshold <level>[, (@chanlist)]	232
15	ARB Subsystem	233
	ARB:COUNT <NRf+>[, (@chanlist)]	233
	ARB:TERMinate:LAST <CPD>[, (@chanlist)]	234
	ARB:FUNcTION:TYPE <CPD>[, (@chanlist)]	234
	ARB:FUNcTION:SHAPE <CPD>[, (@chanlist)]	235
	ARB:PULSe:STARt[:LEVel] <NRf+>[, (@chanlist)]	236
	ARB:PULSe:STARt:TIME <NRf+>[, (@chanlist)]	237
	ARB:PULSe:STARt:SLEW <NRf+>[, (@chanlist)]	238
	ARB:PULSe:TOP[:LEVel] <NRf+>[, (@chanlist)]	239
	ARB:PULSe:TOP:TIME <NRf+>[, (@chanlist)]	240
	ARB:PULSe:END:TIME <NRf+>[, (@chanlist)]	240
	ARB:PULSe:END:SLEW <NRf+>[, (@chanlist)]	241
	ARB:SINusoid:AMPLitude <NRf+>[, (@chanlist)]	242
	ARB:SINusoid:OFFSet <NRf+>[, (@chanlist)]	243
	ARB:SINusoid:FREQuency <NRf+>[, (@chanlist)]	244
	ARB:SWEep:AMPLitude <NRf+>[, (@chanlist)]	244
	ARB:SWEep:OFFSet <NRf+>[, (@chanlist)]	245
	ARB:SWEep:STARt <NRf+>[, (@chanlist)]	246
	ARB:SWEep:END <NRf+>[, (@chanlist)]	247
	ARB:SWEep:STEP[:LEVel] <NRf+>[, (@chanlist)]	248
	ARB:SWEep:STEP:WIDTh <NRf+>[, (@chanlist)]	248
	ARB:UDEfined:COUNT <NR1>[, (@chanlist)]	249
	ARB:UDEfined:LEVel <NR1>[, (@chanlist)]	250

ARB:UDEFined:DWELI <NR1>,<NRf+>[(,@chanlist)]	251
ARB:UDEFined:SLEW <NR1>,<NRf+>[(,@chanlist)]	252
ARB:UDEFined:TRlgout <CPD>[(,@chanlist)]	253
ARB:CDWell:POINts <NR1>[(,@chanlist)]	254
ARB:CDWell[:LEVel] <NRf+>,<NRf+>,<NR1>[(,@chanlist)]	255
ARB:CDWell:DWELI <NRf+>[(,@chanlist)]	256
ARB:CDWell:STOP [(,@chanlist)]	256
ARB:SEQuence:LENGth <NR1>[(,@chanlist)]	257
ARB:SEQuence:SELEct <NR1>[(,@chanlist)]	258
ARB:SEQuence:COUNt <NR1>[(,@chanlist)]	259
ARB:SEQuence:FUNcTION <CPD>[(,@chanlist)]	260
ARB:SEQuence:PACing <CPD>[(,@chanlist)]	261
ARB:SEQuence:PULSe:STARt[:LEVel] <NRf+>[(,@chanlist)]	262
ARB:SEQuence:PULSe:STARt:TIME <NRf+>[(,@chanlist)]	263
ARB:SEQuence:PULSe:STARt:SLEW <NRf+>[(,@chanlist)]	263
ARB:SEQuence:PULSe:TOP:LEVel <NRf+>[(,@chanlist)]	264
ARB:SEQuence:PULSe:TOP:TIME <NRf+>[(,@chanlist)]	265
ARB:SEQuence:PULSe:END:TIME <NRf+>[(,@chanlist)]	266
ARB:SEQuence:PULSe:END:SLEW <NRf+>[(,@chanlist)]	267
ARB:SEQuence:SINusoid:AMPLitude <NRf+>[(,@chanlist)]	268
ARB:SEQuence:SINusoid:OFFSet <NRf+>[(,@chanlist)]	269
ARB:SEQuence:SINusoid:FREQuency <NRf+>[(,@chanlist)]	269
ARB:SEQuence:UDEFined:COUNt <count>[(,@chanlist)]	270
ARB:SEQuence:UDEFined:LEVel <NR1>,<NRf+>[(,@chanlist)]	271
ARB:SEQuence:UDEFined:DWELI <NR1>,<NRf+>[(,@chanlist)]	272
ARB:SEQuence:UDEFined:SLEW <NR1>,<NRf+>[(,@chanlist)]	273
ARB:SAVE <NR1>[(,@chanlist)]	274
ARB:RECall <NR1>[(,@chanlist)]	275
ARB:EXE:STATe? [(,@chanlist)]	275
ARB:SEQuence:CDWell:POINts <NR1>[(,@chanlist)]	276
ARB:SEQuence:CDWell:LEVel <NRf+>,<NRf+>,<NR1>[(,@chanlist)]	277
ARB:SEQuence:CDWell:WIDth <NRf+>[(,@chanlist)]	278
INITiate[:IMMediate]:ARB [(,@chanlist)]	279
ARB:ZOOM:COF <NRf>[(,@chanlist)]	279
ARB:ZOOM:LIMIt:POSitive <NR1>[(,@chanlist)]	280
ARB:ZOOM:LIMIt:NEGative <NR1>[(,@chanlist)]	281
ARB:PAUSe:STATe <Bool>[(,@chanlist)]	282
16 IEEE-488 Common Commands	283
*CLS	283
*ESE <NR1>	284
*ESE?	285
*ESR?	285
*IDN?	286
*OPC	287
*OPC?	288
*RST	289
*SRE <NR1>	291
*SRE?	292
*STB?	293
*TRG	294
*SAV <NR1>[(,@chanlist)]	294
*RCL <NR1>[(,@chanlist)]	295
*TST?	296
*WAI	297
*PSC <Bool>	298
*PSC?	298
17 Format Subsystem	300

17.1	FORMat[:data] <CPD>	300
17.2	DLOG:EXPort:FORMat[:data] <CPD>[,(@chanlist)].....	301
18	Example.....	302
18.1	Dlog.....	302
18.2	Elog.....	302
18.3	List Function.....	303
18.4	Battery mode charge test command.....	304
19	Error Messages	305

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](#)
- ◆ [Multi-channel @chanlist Introduction](#)

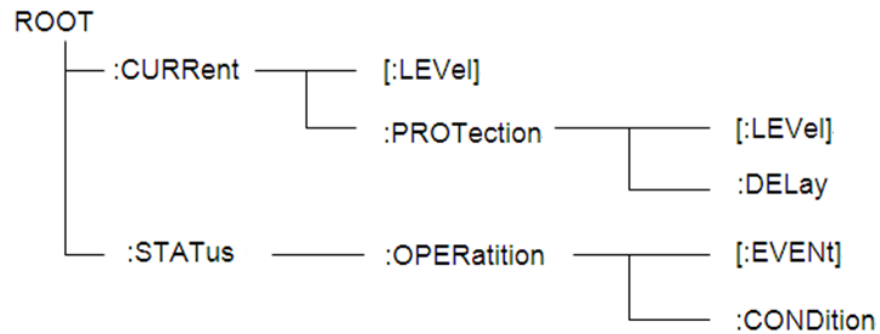
1.1 Overview

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.

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:

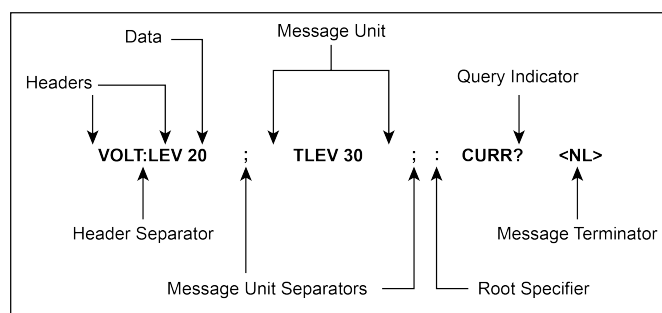
- Set up the proper number of variables for the returned data. For example, if you are reading back a measurement array, you must dimension the array according to the number of measurements that you have placed in the measurement buffer.
- Read back all the results of a query before sending another command 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 an 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 output setting value, the command SYST:REM must be sent to the instrument after finishing the connection between the instrument and PC.

1.8 Multi-channel @chanlist Introduction

Most of the commands in this programming guide support the use of the @Chanlist parameter. When using this parameter, please pay attention to its proper usage to ensure the command takes effect as expected.

For example, in the OUTPut[:STATe] <Bool>[,(@chanlist)] command:

- OUTP 1,(@1,2) turns on the output of both channel 1 and channel 2;
- OUTP 1,(@1) turns on the output of channel 1 only;
- OUTP 1,(@2) turns on the output of channel 2 only;
- OUTP 1 turns on the output of channel 1 by default.

In series or parallel mode, the @chanlist parameter is not required, and the command format should be OUTP 1.

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.

SYS^Tem:CLEar

This command is used to clear the error queue.

Group

System

Syntax

SYS^T:CLE

Arguments

None

Default Value

None

Returns

None

Example

SYS^T:CLE

Also see

None

SYS^Tem: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

ExampleSet 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

SYSTEM:VOLTage:RZERo <CPD>[,(@chanlist)]

This command is used to set the voltage fast zeroing in source mode.

Group

System

Syntax

SYSTEM:VOLTage:RZERo <CPD>[,(@chanlist)]

Arguments

<CPD>[,(@chanlist)]

<CPD>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Example

SYST:VOLT:RZER 0,(@2)

Query Syntax

SYSTEM:VOLTage:RZERo? [(@chanlist)]

Returns

0|1

SYSTEM:ANTI:ACCeSS? [(@chanlist)]

This command is used to check if the sense wire is reversed, 0 indicates not reversed, 1 indicates reversed.

Group

System

Syntax

SYSTem:ANTI:ACCEss? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<NR1>

Example

SYSTem:ANTI:ACCEss? (@2)

3 [SOURce] Subsystem

The SOURce keyword is optional in many commands that set parameters for a source or output, such as **[SOURce:]CURRENT <value>**.

[SOURce:]FUNcTion <CPD>[,(@chanlist)]

This command is used to set the working mode of the power supply.

- VOLTage: Indicates that the power supply is operating in CV priority mode.
- CURRent: Indicates that the power supply is operating in CC priority mode.

Group

Source

Syntax

[SOURce:]FUNcTion <CPD>[,(@chanlist)]

Arguments

<CPD>[,(@chanlist)]

<CPD>: VOLTage|CURRent

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

VOLTage

Returns

None

Example

FUNcTion CURR,(@2)

Also see

[SOURce:]FUNcTion? [(@chanlist)]

[SOURce:]FUNction? [(@chanlist)]

This command is used to query the working mode of the power supply.

Group

Source

Syntax

[SOURce:]FUNction? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

CRD

Example

FUNction? [(@chanlist)]

Also see

None

[SOURce:]FUNction:MODE <CPD>[,(@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]FUNction:MODE <CPD>[,(@chanlist)]

Arguments

<CPD>[,(@chanlist)]

<CPD>: FIXed|LIST|BATTery|FUEL|STEP|EXTAnalog|ARB

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

FIXed

Returns

None

Example

FUNC:MODE LIST,(@2)

Also see

[SOURce:]FUNction:MODE? [(@chanlist)]

[SOURce:]FUNction:MODE? [(@chanlist)]

This command is used to query the function mode of the power supply.

Group

SOURce

Syntax

[SOURce:]FUNction:MODE? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<CRD>

Example

FUNC:MODE? (@2)

Also see

None

**[SOURce:]FUNction:PRiority[:SLEW]:TYPE
<CPD>[,(@chanlist)]**

This command is used to set the slew mode.

Group

SOURce

Syntax**[SOURce:]FUNction:PRiority[:SLEW]:TYPE <CPD>[,(@chanlist)]****Arguments**

<CPD>[,(@chanlist)]

<CPD>: TIME|STANdard

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

TIME

Example

FUNC:PRI:TYPE TIME,(@2)

Query Syntaxes**[SOURce:]FUNction:PRiority[:SLEW]:TYPE? [(@chanlist)]**

Returns

<CPD>

[SOURce:]CURRENT[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

This command is used to set the output current value Iset in CC priority mode.

Group

SOURce

Syntax

[SOURce:]CURRENT[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 1% of the rated current of the instrument

Returns

None

Example

CURRENT MIN,(@2)

Also see

[SOURce:]CURRENT[:LEVel][:IMMediate][:AMPLitude]? [(@chanlist)]

[SOURce:]CURRENT[:LEVel][:IMMediate][:AMPLitude]? [MINimum|MAXimum|DEFault][, (@chanlist)]

This command is used to query the output current value lset in CC priority mode.

Group

SOURce

Syntax

[SOURce:]CURRENT[:LEVel][:IMMediate][:AMPLitude]? [MINimum|MAXimum|DEFault][, (@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][, (@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<NRf+>

Example

CURRENT? MAX,(@2)

Also see

None

[SOURce:]CURRENT:SLEW[:BOTH] <NRf+>[, (@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]CURRent:SLEW[:BOTH] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0.100s(Time slew); IMAX/100(A/ms)(Standard slew).

Returns

None

Example

CURR:SLEW MIN,(@2)

Also see

[SOURce:]CURRent:SLEW[:BOTH]? [,(@chanlist)]

[SOURce:]CURRent:SLEW[:BOTH]? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the current rise and fall slews.

Group

SOURce

Syntax

[SOURce:]CURRent:SLEW[:BOTH]? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

CURR:SLEW? MAX,(@2)

Also see

None

[SOURce:]CURRent:SLEW:POSitive <NRf+>[, (@chanlist)]

This command is used to set the current rise slew.

Group

SOURce

Syntax

[SOURce:]CURRent:SLEW:POSitive <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0.100s(Time slew); IMAX/100(A/ms)(Srandard slew).

Returns

None

Example

CURR:SLEW:POS MIN,(@2)

Also see

[SOURce:]CURRent:SLEW:POSitive? [(@chanlist)]

[SOURce:]CURRent:SLEW:POSitive? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the current rise slew.

Group

SOURce

Syntax

[SOURce:]CURRent:SLEW:POSitive? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

CURR:SLEW:POS? MAX,(@2)

Also see

None

[SOURce:]CURRent:SLEW:NEGative <NRf+>[, (@chanlist)]

This command is used to set the current fall slew.

Group

SOURce

Syntax**[SOURce:]CURRent:SLEW:NEGative <NRf+>[, (@chanlist)]****Arguments**

<NRf+>[, (@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0.100s(Time slew); IMAX/100(A/ms)(Standard slew).

Returns

None

Example**CURR:SLEW:NEG MIN,(@2)****Also see****[SOURce:]CURRent:SLEW:NEGative? [(@chanlist)]**

[SOURce:]CURRent:SLEW:NEGative? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the current fall slew.

Group

SOURce

Syntax

[SOURce:]CURRent:SLEW:NEGative? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

CURR:SLEW:NEG? MAX,@2

Also see

None

[SOURce:]CURRent[:OVER]:PROTection:STATe <Bool>[,(@chanlist)]

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

Group

SOURce

Syntax**[SOURce:]CURRent[:OVER]:PROTection:STATe <Bool>[,(@chanlist)]****Arguments**

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Returns

None

Example**CURR:PROT:STAT 1,(@2)****Also see****[SOURce:]CURRent[:OVER]:PROTection:STATe? [(@chanlist)]**

[SOURce:]CURRent[:OVER]:PROTection:STATe? [(@chanlist)]

This command is used to query the status of overcurrent protection: enabled or disabled.

Group

SOURce

Syntax**[SOURce:]CURRent[:OVER]:PROTection:STATe? [(@chanlist)]**

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<Bool>

Example

CURR:PROT:STAT? (@2)

Also see

None

[SOURce:]CURRent[:OVER]:PROTection[:LEVel] <NRf+>[,(@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]CURRent[:OVER]:PROTection[:LEVel] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: rated current of the instrument

Returns

None

Example

CURR:PROT 10,(@2)

Also see

[SOURce:]CURRent[:OVER]:PROTection[:LEVel]? [(@chanlist)]

[SOURce:]CURRent[:OVER]:PROTection[:LEVel]? [(@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]CURRent[:OVER]:PROTection[:LEVel]? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

CURR:PROT? (@2)

Also see

None

[SOURce:]CURRent[:OVER]:PROTection:DELaY[:TIME] <NRf+>[,(@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]CURRent[:OVER]:PROTection:DELaY[:TIME] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 60s

Returns

None

Example

CURR:PROT:DEL MIN,(@2)

Also see

[SOURce:]CURRent[:OVER]:PROTection:DELaY[:TIME]? [(@chanlist)]

[SOURce:]CURRent[:OVER]:PROTection:DELaY[:TIME]? [MINimum|MAXimum|DEFault][,(@chanlist)]

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

Group

SOURce

Syntax**[SOURce:]CURRent[:OVER]:PROTection:DELAy[:TIME]? [MINimum|MAXimum|DEFault][,(@chanlist)]****Arguments**

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example**CURR:PROT:DEL? DEF,(@2)****Also see**

None

**[SOURce:]CURRent:UNDer:PROTection:STATe
<Bool>[,(@chanlist)]**

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

Group

SOURce

Syntax**[SOURce:]CURRent:UNDer:PROTection:STATe <Bool>[,(@chanlist)]**

Arguments

<Bool>[,(@chanlist)]
<Bool>: 0|OFF|1|ON
(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Returns

None

Example

CURR:UND:PROT:STAT 1,(@2)

Also see

[SOURce:]CURRent:UNDer:PROTection:STATe? [(@chanlist)]

[SOURce:]CURRent:UNDer:PROTection:STATe? [(@chanlist)]

This command is used to query the status of undercurrent protection: enabled or disabled.

Group

SOURce

Syntax

[SOURce:]CURRent:UNDer:PROTection:STATe? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<Bool>

Example

CURR:UND:PROT:STAT? (@2)

Also see

None

[SOURce:]CURRent:UNDer:PROTection[:LEVel] <NRf+>[,(@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]CURRent:UNDer:PROTection[:LEVel] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 0

Returns

None

Example

CURR:UND:PROT 10,(@2)

Also see

[SOURce:]CURRent:UNDer:PROTection[:LEVel]? [(@chanlist)]

[SOURce:]CURRent:UNDer:PROTection[:LEVel]? [(@chanlist)]

This command is used to query the limit value of undercurrent protection.

Group

SOURce

Syntax

[SOURce:]CURRent:UNDer:PROTection[:LEVel]? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

CURR:UND:PROT? (@2)

Also see

None

[SOURce:]CURRent:UNDer:PROTection:DELaY[:TIME] <NRf+>[,(@chanlist)]

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

Group

SOURce

Syntax**[SOURce:]CURRent:UNDer:PROTection:DELay[:TIME] <NRf+>[,
(@chanlist)]****Arguments**

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 60s

Returns

None

Example**CURR:UND:PROT:DEL 10,(@2)****Also see****[SOURce:]CURRent:UNDer:PROTection:DELay[:TIME]? [(@chanlist)]****[SOURce:]CURRent:UNDer:PROTection:DELay[:
TIME]? [MINimum|MAXimum|DEFault][,
(@chanlist)]**

This command is used to query the delay time of undercurrent protection.

Group

SOURce

Syntax

[SOURce:]CURRent:UNDer:PROTection:DELaY[:TIME]? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

Default Value

None

Returns

<NRf+>

Example

CURR:UND:PROT:DEL? MIN,(@2)

Also see

None

[SOURce:]CURRent:UNDer:PROTection:WARM[:TIME] <NRf+>[,(@chanlist)]

This command is used to set the warm-up time of undercurrent protection. This time is set to prevent the instrument from triggering protection when the current is rising. Because this transient condition should not be considered as an undercurrent fault and there is no need to trigger protection.

Group

SOURce

Syntax

[SOURce:]CURRent:UNDer:PROTection:WARM[:TIME] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0s

Returns

None

Example

CURR:UND:PROT:WARM MIN,(@2)

Also see

[SOURce:]CURRent:UNDER:PROTection:WARM[:TIME]? [(@chanlist)]

[SOURce:]CURRent:UNDER:PROTection:WARM[:TIME]? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the warm-up time of undercurrent protection.

Group

SOURce

Syntax

[SOURce:]CURRent:UNDER:PROTection:WARM[:TIME]? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example
CURR:UND:PROT:WARM? DEF,(@2)
Also see

None

[SOURce:]CURRent:LIMit[:POSitive][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

This command is used to set the current upper limit value Ilim in CV priority mode.

Group

SOURce

Syntax
[SOURce:]CURRent:LIMit[:POSitive][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]
Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 1% of the rated current of the instrument

Returns

None

Example

CURR:LIM MAX,(@2)

Also see

**[SOURce:]CURRent:LIMit[:POSitive][:IMMediate][:AMPLitude]?
[(@chanlist)]**

**[SOURce:]CURRent:LIMit[:POSitive][:IMMediate]
[:AMPLitude]? [MINimum|MAXimum|DEFault][,
(@chanlist)]**

This command is used to query the current upper limit value in CV priority mode.

Group

SOURce

Syntax

[SOURce:]CURRent:LIMit[:POSitive][:IMMediate][:AMPLitude]? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<NRf+>

Example

CURR:LIM? MAX,(@2)

Also see

None

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

This command is used to set the output voltage value Vset in CV priority mode.

Group

SOURce

Syntax

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault

Returns

None

Example

VOLT 50,(@2)

Also see

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? [(@chanlist)]

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the output voltage value Vset in CV priority mode.

Group

SOURce

Syntax**[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? [MINimum|MAXimum|DEFault][,(@chanlist)]****Arguments**

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example**VOLT? MAX,(@2)****Also see**

None

[SOURce:]VOLTage:SLEW[:BOTH] <NRf+>[,(@chanlist)]

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

Group

SOURce

Syntax**[SOURce:]VOLTage:SLEW[:BOTH] <NRf+>[,(@chanlist)]**

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0.100s(Time slew); VMAX/100(V/ms)(Standard slew).

Returns

None

Example

VOLT:SLEW MIN,(@2)

Also see

[SOURce:]VOLTage:SLEW[:BOTH]? [(@chanlist)]

[SOURce:]VOLTage:SLEW[:BOTH]? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the voltage rise and fall slews.

Group

SOURce

Syntax

[SOURce:]VOLTage:SLEW[:BOTH]? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<NRf+>

Example

VOLT:SLEW? MAX,(@2)

Also see

None

[SOURce:]VOLTage:SLEW:POSitive <NRf+>[, (@chanlist)]

This command is used to set the voltage rise slew.

Group

SOURce

Syntax

[SOURce:]VOLTage:SLEW:POSitive <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0.100s(Time slew); VMAX/100(V/ms)(Standard slew).

Returns

None

Example

VOLT:SLEW:POS MIN,(@2)

Also see

[SOURce:]VOLTage:SLEW:POSitive? [(@chanlist)]

[SOURce:]VOLTage:SLEW:POSitive? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the voltage rise slew.

Group

SOURce

Syntax

[SOURce:]VOLTage:SLEW:POSitive? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<NRf+>

Example

VOLT:SLEW:POS? MAX,(@2)

Also see

None

[SOURce:]VOLTage:SLEW:NEGative <NRf+>[,(@chanlist)]

This command is used to set the voltage fall slew.

Group

SOURce

Syntax

[SOURce:]VOLTage:SLEW:NEGative <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0.100s(Time slew); VMAX/100(V/ms)(Standard slew).

Returns

None

Example

VOLT:SLEW:NEG MIN,(@2)

Also see

[SOURce:]VOLTage:SLEW:NEGative? [(@chanlist)]

[SOURce:]VOLTage:SLEW:NEGative? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the voltage drop slew.

Group

SOURce

Syntax

[SOURce:]VOLTage:SLEW:NEGative? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<NRf+>

Example

VOLT:SLEW:NEG? MAX,(@2)

Also see

None

[SOURce:]VOLTage[:OVER]:PROTection:STATe <Bool>[,(@chanlist)]

Enable or disable overvoltage protection. If overvoltage protection is enabled and the output enters a voltage limit, the output is disabled.

Group

SOURce

Syntax

[SOURce:]VOLTage[:OVER]:PROTection:STATe <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Returns

None

Example**VOLT:PROT:STAT 1,(@2)****Also see****[SOURce:]VOLTage[:OVER]:PROTection:STATe? [(@chanlist)]**

[SOURce:]VOLTage[:OVER]:PROTection:STATe? [(@chanlist)]

This command is used to query the status of overvoltage protection: enabled or disabled.

Group

SOURce

Syntax**[SOURce:]VOLTage[:OVER]:PROTection:STATe? [(@chanlist)]****Arguments**

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<Bool>

Example**VOLT:PROT:STAT? (@2)****Also see**

None

[SOURce:]VOLTage[:OVER]:PROTection[:LEVel] <NRf+>[,(@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]VOLTage[:OVER]:PROTection[:LEVel] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: rated voltage of the instrument

Returns

None

Example

VOLT:PROT 100,(@2)

Also see

[SOURce:]VOLTage[:OVER]:PROTection[:LEVel]? [(@chanlist)]

[SOURce:]VOLTage[:OVER]:PROTection[:LEVel]? [(@chanlist)]

This command is used to query the limit value of overvoltage protection.

Group

SOURce

Syntax**[SOURce:]VOLTage[:OVER]:PROTection[:LEVel]? [(@chanlist)]****Arguments**

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

VOLT:PROT? (@2)

Also see

None

[SOURce:]VOLTage[:OVER]:PROTection:DELAy[:TIME] <NRf+>[,(@chanlist)]

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

Group

SOURce

Syntax**[SOURce:]VOLTage[:OVER]:PROTection:DELAy[:TIME] <NRf+>[,(@chanlist)]****Arguments**

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 60S

Returns

None

Example

VOLT:PROT:DEL MIN,(@2)

Also see

[SOURce:]VOLTage[:OVER]:PROTection:DELay[:TIME]? [(@chanlist)]

[SOURce:]VOLTage[:OVER]:PROTection:DELay[:TIME]? [MINimum|MAXimum|DEFault][, (@chanlist)]

This command is used to query the delay time of overvoltage protection.

Group

SOURce

Syntax

[SOURce:]VOLTage[:OVER]:PROTection:DELay[:TIME]? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

VOLT:PROT:DEL? DEF,(@2)

Also see

None

[SOURce:]VOLTage:UNDER:PROTection:STATe <Bool>[,(@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]VOLTage:UNDER:PROTection:STATe <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Returns

None

Example

VOLT:UND:PROT:STAT 1,(@2)

Also see

[SOURce:]VOLTage:UNDer:PROTection:STATe? [(@chanlist)]

[SOURce:]VOLTage:UNDer:PROTection:STATe? [(@chanlist)]

This command is used to query the status of undervoltage protection: enabled or disabled.

Group

SOURce

Syntax

[SOURce:]VOLTage:UNDer:PROTection:STATe? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<Bool>

Example

VOLT:UND:PROT:STAT? (@2)

Also see

None

[SOURce:]VOLTage:UNDer:PROTection[:LEVel] <NRf+>[,(@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]VOLTage:UNDER:PROTection[:LEVel] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 0

Returns

None

Example

VOLT:UND:PROT 10,(@2)

Also see

[SOURce:]VOLTage:UNDER:PROTection[:LEVel]? [(@chanlist)]

[SOURce:]VOLTage:UNDER:PROTection[:LEVel]? [(@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]VOLTage:UNDER:PROTection[:LEVel]? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<NRf+>

Example

VOLT:UND:PROT? (@2)

Also see

None

[SOURce:]VOLTage:UNDER:PROTection:DELay[: TIME] <NRf+>[,(@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]VOLTage:UNDER:PROTection:DELay[:TIME] <NRf+>[,
(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 60s

Returns

None

Example

```
VOLT:UND:PROT:DEL 10,(@2)
```

Also see

```
[SOURce:]VOLTage:UNDer:PROTection:DELaY[:TIME]? [(@chanlist)]
```

```
[SOURce:]VOLTage:UNDer:PROTection:DELaY[:TIME]? [MINimum|MAXimum|DEFault][, (@chanlist)]
```

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

Group

SOURce

Syntax

```
[SOURce:]VOLTage:UNDer:PROTection:DELaY[:TIME]? [MINimum|MAXimum|DEFault][,(@chanlist)]
```

Arguments

```
[MINimum|MAXimum|DEFault][,(@chanlist)]
```

```
(@chanlist): (@1)|(@2)|(@1,2)
```

Default Value

None

Returns

<NRf+>

Example

```
VOLT:UND:PROT:DEL? MIN,(@2)
```

Also see

None

[SOURce:]VOLTage:UNDER:PROTection:WARM[:TIME] <NRf+>[,(@chanlist)]

This command is used to set the warm-up time of undervoltage protection. This time is set to prevent the instrument from triggering protection when the voltage is rising. Because this transient condition should not be considered as an undervoltage fault and there is no need to trigger protection.

Group

SOURce

Syntax

[SOURce:]VOLTage:UNDER:PROTection:WARM[:TIME] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 60s

Returns

None

Example

VOLT:UND:PROT:WARM MIN,(@2)

Also see

[SOURce:]VOLTage:UNDER:PROTection:WARM[:TIME]? [(@chanlist)]

[SOURce:]VOLTage:UNDER:PROTection:WARM[:TIME]? [MINimum|MAXimum|DEFault][, (@chanlist)]

This command is used to query the warm-up time of undervoltage protection.

Group

SOURce

Syntax

[SOURce:]VOLTage:UNDER:PROTection:WARM[:TIME]? [MINimum|MAXimum|DEFault][, (@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][, (@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

VOLT:UND:PROT:WARM? DEF,(@2)

Also see

None

[SOURce:]VOLTage:LIMit[:POSitive][:IMMEDIATE] [:AMPLitude] <NRf+>[, (@chanlist)]

This command is used to set the voltage upper limit value V_{lim} in CC priority mode.

Group

SOURce

Syntax**[SOURce:]VOLTage:LIMit[:POSitive][:IMMEDIATE][:AMPLitude] <NRf+>[, (@chanlist)]****Arguments**

<NRf+>[, (@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault

Returns

None

Example**VOLT:LIM MAX,(@2)****Also see****[SOURce:]VOLTage:LIMit[:POSitive][:IMMEDIATE][:AMPLitude]?
[(@chanlist)]****[SOURce:]VOLTage:LIMit[:POSitive][:IMMEDIATE][:AMPLitude]? [MINimum|MAXimum|DEFault][, (@chanlist)]**

This command is used to query the voltage upper limit value in CC priority mode.

Group

SOURce

Syntax

[SOURce:]VOLTage:LIMit[:POSitive][:IMMEDIATE][:AMPLitude]? [MINimum|MAXimum|DEFAULT][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFAULT][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<NRf+>

Example

VOLT:LIM? MAX,(@2)

Also see

None

[SOURce:]POWER:LIMit[:POSitive][:IMMEDIATE][:AMPLitude] <NRf+>[,(@chanlist)]

This command is used to set the power upper limit value Plim.

Group

SOURce

Syntax

[SOURce:]POWER:LIMit[:POSitive][:IMMEDIATE][:AMPLitude] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFAULT<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: Rated power value of the instrument

Returns

None

Example

POW:LIM MAX,(@2)

Also see

**[SOURce:]POWER:LIMit[:POSitive][:IMMEDIATE][:AMPLitude]?
[(@chanlist)]**

[SOURce:]POWER:LIMit[:POSitive][:IMMEDIATE][:AMPLitude]? [MINimum|MAXimum|DEFault][, (@chanlist)]

This command is used to query the power upper limit value.

Group

SOURce

Syntax

[SOURce:]POWER:LIMit[:POSitive][:IMMEDIATE][:AMPLitude]? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

POW:LIM? MAX,(@2)

Also see

None

[SOURce:]POWER[:OVER]:PROTection:STATe <Bool>[,(@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]POWER[:OVER]:PROTection:STATe <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Returns

None

Example

POW:PROT:STAT 1,(@2)

Also see

[SOURce:]POWER[:OVER]:PROTection:STATe? [(@chanlist)]

[SOURce:]POWer[:OVER]:PROTection:STATe? [(@chanlist)]

This command is used to query the status of power protection: enabled or disabled.

Group

SOURce

Syntax

[SOURce:]POWer[:OVER]:PROTection:STATe? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<Bool>

Example

POW:PROT:STAT? (@2)

Also see

None

[SOURce:]POWer[:OVER]:PROTection[:LEVel] <NRf+>[(@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]POWer[:OVER]:PROTection[:LEVel] <NRf+>[(,@chanlist)]

Arguments

<NRf+>[(,@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: rated power of the instrument

Returns

None

Example

POW:PROT 100,(@2)

Also see

[SOURce:]POWer[:OVER]:PROTection[:LEVel]? [(,@chanlist)]

[SOURce:]POWer[:OVER]:PROTection[:LEVel]? [(,@chanlist)]

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

Group

SOURce

Syntax

[SOURce:]POWer[:OVER]:PROTection[:LEVel]? [(,@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

POW:PROT? (@2)

Also see

None

**[SOURce:]POWER[:OVER]:PROTection:DELaY[:
TIME] <NRf+>[,(@chanlist)]**

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

Group

SOURce

Syntax**[SOURce:]POWER[:OVER]:PROTection:DELaY[:TIME] <NRf+>[,
(@chanlist)]****Arguments**

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 60s

Returns

None

Example

```
POW:PROT:DEL MIN,(@2)
```

Also see

```
[SOURce:]POWER[:OVER]:PROTection:DELAy[:TIME]? [(@chanlist)]
```

```
[SOURce:]POWER[:OVER]:PROTection:DELAy[:  
TIME]? [MINimum|MAXimum|DEFault][,  
(@chanlist)]
```

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

Group

SOURce

Syntax

```
[SOURce:]POWER[:OVER]:PROTection:DELAy[:TIME]? [MINimum|MAXi-  
mum|DEFault][,(@chanlist)]
```

Arguments

```
[MINimum|MAXimum|DEFault][,(@chanlist)]
```

```
(@chanlist): (@1)|(@2)|(@1,2)
```

Returns

```
<NRf+>
```

Example

```
POW:PROT:DEL? DEF,(@2)
```

Also see

None

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

This command is used to set the internal resistance of the power supply in CV priority mode. Only used in CV priority mode.

Group

SOURce

Syntax

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0

Returns

None

Example

RES 100,(@2)

Also see

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude]? [(@chanlist)]

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude]? [MINimum|MAXimum|DEFault][, (@chanlist)]

This command is used to query the internal resistance of the power supply in CV priority mode. Only used in CV priority mode.

Group

SOURce

Syntax

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude]? [MINimum|MAXimum|DEFault][, (@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][, (@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<NRf+>

Example

RES? MAX,(@2)

Also see

None

[SOURce:]APPLy <volt>, <curr>[, (@chanlist)]

This command is used to configure Vs and I+ in CV priority mode, or Is and Vh in CC priority mode.

Group

SOURce

Syntax

[SOURce:]APPLy <volt>,<curr>[,(@chanlist)]

Arguments

<volt>,<curr>[,(@chanlist)]

<volt>: MINimum|MAXimum|DEFault|<MIN~MAX>

<curr>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

<volt>: VMAX/100

<curr>: IMAX/100

Example

APPL 10,5,(@2)

Query Syntaxes

[SOURce:]APPLy? [(@chanlist)]

Returns

<NRf+>,<NRf+>

[SOURce:]EXTErn:PROGrama[:STATe] <Bool>[, (@chanlist)]

Enable or disable the external analog function.

Group

SOURce

Syntax

[SOURce:]EXTErn:PROGrama[:STATe] <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Returns

None

Example

EXT:PROG 1,(@2)

Also see

[SOURce:]EXTErn:PROGrama[:STATe]? [(@chanlist)]

[SOURce:]EXTErn:PROGrama[:STATe]? [(@chanlist)]

This command is used to query the status of the external analog function: enabled or disabled.

Group

SOURce

Syntax

[SOURce:]EXTErn:PROGrama[:STATe]? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<Bool>

Example

```
EXT:PROG? (@2)
```

Also see

None

[SOURce:]EXTErn:PROGrama:CHANnel:MX <NR1>,<NRf>[,(@chanlist)]

This command is used to set the linear calibration coefficient of the channel (1/2/3). The instruction needs to set two parameters, separated by commas, the channel number in front of the comma, and the linear calibration coefficient after the comma.

Group

SOURce

Syntax

```
[SOURce:]EXTErn:PROGrama:CHANnel:MX <NR1>,<NRf>[,(@chanlist)]
```

Arguments

- <NR1>, Settings: 1, 2, 3
- <NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>
- (@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault

Returns

None

Example

```
EXT:PROG:CHAN:MX 1,10,(@2)
```

Also see

[SOURce:]EXTErn:PROGrama:CHANnel:MX? <NR1>[,(@chanlist)]

[SOURce:]EXTErn:PROGrama:CHANnel:MX? <NR1>[,(@chanlist)]

This command is used to query the linear calibration coefficient of the channel (1/2/3).

Group

SOURce

Syntax

[SOURce:]EXTErn:PROGrama:CHANnel:MX? <NR1>[,(@chanlist)]

Arguments

<NR1>[,(@chanlist)]

<NR1>, Setting: 1,2,3

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<NRf+>

Example

EXT:PROG:CHAN:MX? 2,(@2)

Also see

None

[SOURce:]EXTErn:PROGrama:CHANnel:MB <NR1>,<NRf>[,(@chanlist)]

This command is used to set the offset calibration coefficient of the channel (1/2/3). The instruction needs to set two parameters, separated by commas, the

channel number in front of the comma, and the offset calibration coefficient after the comma.

Group

SOURce

Syntax

[SOURce:]EXTErn:PROGram:CHANnel:MB <NR1>,<NRf>[,(@chanlist)]

Arguments

- <NR1>, Settings: 1, 2, 3
- <NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>
- (@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault

Returns

None

Example

EXT:PROG:CHAN:MB 1,20,(@2)

Also see

[SOURce:]EXTErn:PROGram:CHANnel:MB? <NR1>[,(@chanlist)]

[SOURce:]EXTErn:PROGram:CHANnel:MB? <NR1>[,(@chanlist)]

This command is used to query the offset calibration coefficient of the channel (1/2/3).

Group

SOURce

Syntax

[SOURce:]EXTErn:PROGram:CHANnel:MB? <NR1>[,(@chanlist)]

Arguments

<NR1>[,(@chanlist)]

<NR1>, Settings: 1, 2, 3

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

EXT:PROG:CHAN:MB? 2,(@2)

Also see

None

[SOURce:]FILTEr:LEVEl <CPD>[,(@chanlist)]

This command is used to set the filter level.

Group

SOURce

Syntax

[SOURce:]FILTEr:LEVEl <CPD>[,(@chanlist)]

Arguments

<CPD>[,(@chanlist)]

<CPD>: LOW|MEDIum|FAST

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

LOW

Returns

None

Example

FILT:LEV MED,(@2)

Also see

[SOURce:]FILTer:LEVel? [(@chanlist)]

[SOURce:]FILTer:LEVel? [(@chanlist)]

This command is used to query the filter level.

Group

SOURce

Syntax

[SOURce:]FILTer:LEVel? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<CRD>

Example

FILT:LEV? (@2)

Also see

None

[SOURce:]CV:PRiority <priority>[,(@chanlist)]

This command is used to set CV loop speed.

Group

SOURce

Syntax

[SOURce:]CV:PRiority <priority>[,(@chanlist)]

Arguments

<priority>: HIGH|LOW

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

HIGH

Example

CV:PRI HIGH,(@2)

Query Syntax

[SOURce:]CV:PRiority? [(@chanlist)]

Returns

HIGH|LOW

[SOURce:]CC:PRiority <priority>[,(@chanlist)]

This command is used to set CC loop speed.

Group

SOURce

Syntax

[SOURce:]CC:PRiority <priority>[,(@chanlist)]

Arguments

<priority>: HIGH|LOW

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

HIGH

Example

CC:PRI HIGH,(@2)

Query Syntax

[SOURce:]CC:PRiority? [(@chanlist)]

Returns

HIGH|LOW

[SOURce:]PROTection:CLEar [(@chanlist)]

This command is used to clear the protection status.

Group

SOURce

Syntax

[SOURce:]PROTection:CLEar [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

None

Example

PROT:CLE (@2)

Also see

None

4 OUTPut Subsystem

The Output subsystem controls the output state, power-on, protection, and relay functions.

OUTPut[:STAtE] <Bool>[,(@chanlist)]

Enable or disable the output.

Group

OUTPut

Syntax

OUTPut[:STAtE] <Bool>[,(@chanlist)]

Arguments

<CPD>

0|OFF|1|ON

Default Value

0|OFF

Returns

None

Example

OUTP 1,(@2)

Also see

OUTPut[:STAtE]? [(@chanlist)]

OUTPut[:STATe]? [(@chanlist)]

This command is used to query the status of the output: enabled or disabled.

Group

OUTPut

Syntax

OUTPut[:STATe]? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<CRD>

Example

OUTP? (@2)

Also see

None

OUTPut:DELAy:FALL <NRf+>[,(@chanlist)]

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

Group

OUTPut

Syntax

OUTPut:DELAy:FALL <NRf+>[,(@chanlist)]

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<value>

Setting range: MIN to MAX

Default Value

DEFault: 0

Returns

None

Example

OUTP:DEL:FALL 6,(@2)

Also see

OUTPut:DELay:FALL? [(@chanlist)]

OUTPut:DELay:FALL? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the delay time for the output to be turned off.

Group

OUTPut

Syntax

OUTPut:DELay:FALL? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault]

Default Value

None

Returns

<NRf+>

Example

OUTP:DEL:FALL? MAX,(@2)

Also see

None

OUTPut:DELay:RISE <NRf+>[,(@chanlist)]

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

Group

OUTPut

Syntax

OUTPut:DELay:RISE <NRf+>[,(@chanlist)]

Arguments

<NRf+>

MINimum|MAXimum|DEFault|<value>

Setting range: MIN to MAX

Default Value

DEFault: 0

Returns

None

Example

OUTP:DEL:RISE 6,(@2)

Also see

OUTPut:DELay:RISE? [(@chanlist)]

OUTPut:DELay:RISE? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the delay time for the output to be turned on.

Group

OUTPut

Syntax

OUTPut:DELay:RISE? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault]

Default Value

None

Returns

<NRf+>

Example

OUTP:DEL:RISE? MAX,(@2)

Also see

None

OUTPut:PON:STATe <CPD>

This command is used to control several parameter settings and the output status (on or off) when the instrument is powered on.

- RST: The default value indicates that the factory initialization value is displayed when the instrument is powered on. The specific parameters are described in [*RST](#).
- LAST_ON: Indicates when powered on, the instrument will remain the same parameter settings as last time you turned off the instrument.
- LAST_OFF: Indicates when powered on, the instrument will remain the same settings as last time you turned off the instrument, but the output state is **Off**.


Note

The parameters involved in LAST_ON and LAST_OFF are those related to the ***SAV 0** command.

Group

OUTPut

Syntax

OUTPut:PON:STATe <CPD>

Arguments

<CPD>

RST|LAST_ON|LAST_OFF

Default Value

RST

Returns

None

Example

OUTP:PON:STAT LAST_ON

Also see

OUTPut:PON:STATe?

OUTPut:PROTection:WDOG[:STATe] <CPD>[, (@chanlist)]

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

Group

OUTPut

Syntax

OUTPut:PROTection:WDOG[:STATe] <CPD>[,(@chanlist)]

Arguments

<CPD>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Example

OUTP:PROT:WDOG 1,(@2)

Query Syntax

OUTPut:PROTection:WDOG[:STATe]? [(@chanlist)]

Returns

0|1

OUTPut:PROTection:WDOG:DELay <NRf+>[, (@chanlist)]

Sets the watchdog delay time. When the watchdog timer is enabled, the output 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 output will still shut down after the time period has elapsed. Programmed values can range from 1 to 3600 seconds in 1 second increments.

Group

OUTPut

Syntax**OUTPut:PROTection:WDOG:DELay <NRf+>[,(@chanlist)]****Arguments**

<NRf+>: <value>|MIN|MAX|DEF

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEF: 0s

Example**OUTP:PROT:WDOG:DEL 600,(@2)****Query Syntax****OUTPut:PROTection:WDOG:DELay? [(@chanlist)]****Returns**

<NRf>

OUTPut:MODE <CPD>

This command is used to set the output mode of the machine.

Group

OUTPut

Syntax**OUTPut:MODE <CPD>**

Arguments

SERies|PARAllel|DUAL|ONE_CHANNEL

Default Value

DUAL

Example**OUTPut:MODE DUAL****Query Syntaxes****OUTPut:MODE?****Returns**

SERies|PARAllel|DUAL|ONE_CHANNEL

OUTPut:COUPlE:SYNChronous[:STATe] <Bool>

This command is used to enable or disable the dual-channel synchronous switch.

Group

OUTPut

Syntax**OUTPut:COUPlE:SYNChronous[:STATe] <Bool>****Arguments**

0|OFF|1|ON

Default Value

0|OFF

Example**OUTP:COUP:SYNCh ON**

Query Syntaxes

OUTPut:COUPle:SYNChronous[:STATe]?

Returns

0|1

OUTPut:COUPle:MODE <CPD>

This command is used to set the synchronization mode of dual channels.

Group

OUTPut

Syntax

OUTPut:COUPle:MODE <CPD>

Arguments

OUTPut|TRACK|DUPLicate

Default Value

OUTPut

Example

OUTPut:COUPle:MODE OUTPut

Query Syntaxes

OUTPut:COUPle:MODE?

Returns

OUTPut|TRACK|DUPLicate

OUTPut:COUPle:SYNChronous:REFerence <NRf +>

Set the voltage proportional relationship among 2-channel.

Group

OUTPut

Syntax

OUTPut:COUPle:SYNChronous:REFerence <NRf+>

Arguments

<NRf+>

Default Value

1

Example

OUTPut:COUPle:SYNChronous:REFerence 1

Query Syntaxes

OUTPut:COUPle:SYNChronous:REFerence?

Returns

<NRf+>

OUTPut[:STATe]:COUPle:DOFFset <NRf+>[, (@chanlist)]

This command is used to set the internally programmable delay time.

Group

OUTPut

Syntax

OUTPut[:STATe]:COUPle:DOFFset <NRf+>[,(@chanlist)]

Arguments

<MIN ~ MAX>|MINimum|MAXimum|DEFault

Default Value

0.001s

Example

OUTPut[:STATe]:COUPle:DOFFset 0.002,(@2)

Query Syntaxes

OUTPut[:STATe]:COUPle:DOFFset? [(@chanlist)]

Returns

<NRf+>

5 ABORt Subsystem

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

ABORt:ACQuire [(@chanlist)]

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

Group

ABORt

Syntax

ABORt:ACQuire [(@chanlist)]

Arguments

None

Default Value

None

Returns

None

Example

ABOR:ACQ (@1,2)

Also see

None

ABORt:ARB [(@chanlist)]

Cancel the execution of the ARB subsystem instructions.

Group

ABORt

Syntax**ABORt:ARB [(@chanlist)]****Arguments**

None

Default Value

None

Returns

None

Example**ABOR:ARB (@1,2)****Also see**

None

6 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[(@chanlist)]

Initiates the measurement trigger system.

Group

INITiate

Syntax

INITiate[:IMMediate]:ACQuire[(@chanlist)]

Arguments

None

Default Value

None

Returns

None

Example

INIT:ACQ(@1)

Also see

None

INITiate:CONTInuous:ARB <Bool>[,(@chanlist)]

This command is used to enable or disable the ARB continuous trigger function. Use occasion: Single-waveform mode, when a trigger signal is received, a waveform output is generated.

Group

INITiate

Syntax

INITiate:CONTInuous:ARB <Bool>[,(@chanlist)]

Arguments

<Bool>

0|OFF|1|ON

Default Value

0|OFF

Returns

None

Example

INIT:CONT:ARB 1,(@2)

Also see

INITiate:CONTInuous:ARB?[(@chanlist)]

INITiate:CONTinuous:ACQuire <Bool>[, (@chanlist)]

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

Group

INITiate

Syntax

INITiate:CONTinuous:ACQuire <Bool>[,(@chanlist)]

Arguments

<Bool>

0|OFF|1|ON

Default Value

1|ON

Returns

None

Example

INIT:CONT:ACQ 0,(@1)

Also see

INITiate:CONTinuous:ACQuire?[(@chanlist)]

INITiate[:IMMediate]:DLOG[(@chanlist)]

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[(@chanlist)]

Arguments

None

Default Value

None

Returns

None

Example

INIT:DLOG(@1)

Also see

None

INITiate[:IMMediate]:ELOG[(@chanlist)]

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[(@chanlist)]

Arguments

None

Default Value

None

Returns

None

Example`INIT:ELOG(@1)`**Also see**

None

7 BATTery Subsystem

Suitable for battery testing related functions.

BATTery:CHARge:VOLTage <NRf+>[,(@chanlist)]

This command is used to set the battery charging voltage value.

Group

BATTery

Syntax

BATTery:CHARge:VOLTage <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 0

Returns

None

Example

BATT:CHAR:VOLT 20,(@1)

Also see

BATTery:CHARge:VOLTage? [(@chanlist)]

BATTery:CHARge:VOLTage? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the battery charging voltage value.

Group

BATTery

Syntax

BATTery:CHARge:VOLTage? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

BATT:CHAR:VOLT? MAX,(@1)

Also see

None

BATTery:CHARge:CURRent <NRf+>[,(@chanlist)]

This command is used to set the battery charging current value.

Group

BATTery

Syntax

BATTERY:CHARGE:CURRENT <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 0

Returns

None

Example

BATT:CHAR:CURR 2,(@2)

Also see

BATTERY:CHARGE:CURRENT? [(@chanlist)]

BATTERY:CHARGE:CURRENT? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the battery charging current value.

Group

BATTERY

Syntax

BATTERY:CHARGE:CURRENT? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example**BATT:CHAR:CURR? MAX,(@2)****Also see**

None

BATTery:SHUT:VOLTage <NRf+>[,(@chanlist)]

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

Group

BATTery

Syntax**BATTery:SHUT:VOLTage <NRf+>[,(@chanlist)]****Arguments**

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 0

Returns

None

Example

BATT:SHUT:VOLT 200,(@1)

Also see

BATTery:SHUT:VOLTage? [(@chanlist)]

BATTery:SHUT:VOLTage? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the voltage value of the battery test cutoff.

Group

BATTery

Syntax

BATTery:SHUT:VOLTage? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

BATT:SHUT:VOLT? MAX,(@1)

Also see

None

BATTery:SHUT:CURRent <NRf+>[,(@chanlist)]

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

Group

BATTery

Syntax

BATTery:SHUT:CURRent <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 0

Returns

None

Example

BATT:SHUT:CURR 5,(@2)

Also see

BATTery:SHUT:CURRent? [(@chanlist)]

BATTery:SHUT:CURRent? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the current value of the battery test cutoff.

Group

BATTery

Syntax

BATTERY:SHUT:CURRENT? [MINimum|MAXimum|DEFAULT][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFAULT][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

BATT:SHUT:CURR? MAX,(@2)

Also see

None

BATTERY:SHUT:CAPacity <NRf+>[,(@chanlist)]

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

Group

BATTERY

Syntax

BATTERY:SHUT:CAPacity <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFAULT|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 1 AH

Returns

None

Example**BATT:SHUT:CAP 50,(@2)****Also see****BATTery:SHUT:CAPacity? [(@chanlist)]**

BATTery:SHUT:CAPacity? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the capacity value of the battery test cutoff.

Group

BATTery

Syntax**BATTery:SHUT:CAPacity? [MINimum|MAXimum|DEFault][,(@chanlist)]****Arguments**

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

BATT:SHUT:CAP? MAX,(@1)

Also see

None

BATTery:SHUT:ENERgy <NRf+>[,(@chanlist)]

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

Group

BATTery

Syntax

BATTery:SHUT:ENERgy <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 0WH

Example

BATT:SHUT:ENER 50,(@1)

Query Syntax

BATTery:SHUT:ENERgy? [(@chanlist)]

Returns

NRf

BATTERY:SHUT:TIME <NRf+>[,(@chanlist)]

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

Group

BATTERY

Syntax

BATTERY:SHUT:TIME <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MINimum|MAXimum|DEFault|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DEFault: 20 s

Returns

None

Example

BATTERY:SHUT:TIME 3,(@1)

Also see

BATTERY:SHUT:TIME? [(@chanlist)]

BATTERY:SHUT:TIME? [MINimum|MAXimum|DEFault][,(@chanlist)]

This command is used to query the time when the battery test is cut off.

Group

BATTERY

Syntax

BATTery:SHUT:TIME? [MINimum|MAXimum|DEFault][,(@chanlist)]

Arguments

[MINimum|MAXimum|DEFault][,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<NRf+>

Example

BATT:SHUT:TIME? MAX,(@1)

Also see

None

BATTery:RUN:STATe <Bool>[,(@chanlist)]

Set the RUN/STOP for battery testing.

Group

BATTery

Syntax

BATTery:RUN:STATe <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Example**BATTery:RUN:STATe 1,(@2)****Query Syntaxes**

BATTery:RUN:STATe? [(@chanlist)]

Returns

0|1

BATTery:SAVe <filename>[,(@chanlist)]

Save the battery file with the specified filename .

Group

BATTery

Syntax**BATTery:SAVe <filename>[,(@chanlist)]****Arguments**

<filename>[,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Example**BATTery:SAVe List1,(@1)**

BATTery:RECall <filename>[,(@chanlist)]

Open the battery file with the specified filename.

Group

BATTery

Syntax

BATTery:RECall <filename>[,(@chanlist)]

Arguments

<filename>[,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Example

BATTery:RECall List1,@2

BATTery:DElete <filename>[,(@chanlist)]

Delete the battery file with the specified filename.

Group

BATTery

Syntax

BATTery:DElete <filename>[,(@chanlist)]

Arguments

<filename>[,(@chanlist)]

(@chanlist): (@1)|(@2)|(@1,2)

Example

BATTery:DElete List1,@2

BATTery:TRIG [(@chanlist)]

Trigger battery testing function.

Group

BATTery

Syntax

BATTery:TRIG [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Example

BATTery:TRIG (@2)

8 CONFigurable Subsystem

Applies to digital I/O function.

IO:CHANnel:SElect <port>,<chl>

This command applies the IO pin configuration to the selected channel.

Group

CONFigurable

Syntax

IO:CHANnel:SElect <port>,<chl>

Arguments

<port>: Used to specify the pin number. Pins 2 to 7 correspond to number 1 to 6.

<chl>: Used to specify the channel. CH1|CH2|ALL

Example

Select pin 3: **IO:CHAN:SEL 2,CH1**

Query Syntaxes

IO:CHANnel:SElect? <port>

Returns

<1~6>,<chl>

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

ExampleSelect 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

ExampleSet 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 2 correspond to number 0 to 1.
- <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 1, 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 1

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 1, 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 1, 0 to 100

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 1

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. It can be set as the first function and the second function: The default first function of the 7 pins is PSClear, PASSta, OFFSta, EXTTrig (TOUT), INHLiv (INHLat), SYON, SYOFF; IO1 and IO2 also have PWM functions; IO1 to IO6 all have IORD and IOWR functions.

- <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:OUTPut:LEVeI <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:LEVeI <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:LEVeI? <NRL>

IO:OUTPut:LEVeI? <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:LEVeI? <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:LEVeI? <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:LEVeI? <NRL>****Arguments**

<NRL>
0 to 6

Default Value

None

Returns

<CRD>
0|LOW|1|HIGH

Example

IO:INP:LEV? 1

Also see

None

[SOURce:]DIGital:RElay:TYPE <NR1>,<CPD>

This command is used to set the operating mode of the IT-E179 relay card.

Relay port number	Function 1 (default function)	Function 2
Relay 1	-	Output (common relay)
Relay 2	Sense1 (control sense1)	Output (common relay)
Relay 3	Sense2 (control sense2)	Output (common relay)
Relay 4	KP (parallel output)	Output (common relay)
Relay 5	KS (series output)	Output (common relay)

Group

SOURce

Syntax

[SOURce:]DIGital:RElay:TYPE <NR1>,<CPD>

Arguments

<NR1>,<CPD>

<NR1>: 0 ~ 4

<CPD>: OUTPut|Sense1|Sense2|KP|KS

Example

DIG:REI:TYPE 0,OUTPut

Query Syntaxes

[SOURce:]DIGital:RElay:TYPE? <NR1>

Returns

OUTPut|Sense|KP|KS

[SOURce:]DIGital:RElay:LEVel <NR1>,<Bool>

This command is used to sets the suction state of the IT-E179 relay card.

Group

SOURce

Syntax

[SOURce:]DIGital:RElay:LEVel <NR1>,<Bool>

Arguments

<NR1>,<Bool>

<NR1>: 0 ~ 4

<Bool>: 0|OFF|1|ON

Example

DIG:REI:LEV 3,1

Query Syntaxes

[SOURce:]DIGital:RElay:LEVel? <NR1>

Returns

0|1|

9 TRIGger Subsystem

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

TRIGger:ACQuire[:IMMediate] [(@chanlist)]

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] [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Example

TRIG:ACQ (@2)

Query Syntax

None

Returns

None

TRIGger:ACQuire:MODE <CPD>[,(@chanlist)]

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>[,(@chanlist)]

Arguments

<CPD>[,(@chanlist)]

<CPD>: AUTO|NORMal

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

NORMal

Example

TRIGger:ACQuire:MODE AUTO,(@2)

Query Syntax

TRIGger:ACQuire:MODE? [(@chanlist)]

Returns

AUTO|NORMal

TRIGger:ACQuire:SOURce <CPD>[,(@chanlist)]

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 **CON-Figurable** 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>[,(@chanlist)]
Arguments

<CPD>[,(@chanlist)]

<CPD>: VOLTage|CURRent|EXTernal|BUS|MANual|IMMediate

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

IMMediate

Example
TRIGger:ACQuire:SOURce EXTernal,(@2)
Query Syntax
TRIGger:ACQuire:SOURce? [(@chanlist)]
Returns

VOLTage|CURRent|EXTernal|BUS|MANual|IMMediate

TRIGger:ACQUIRE:VOLTage:SLOPe <CPD>[, (@chanlist)]

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>[, (@chanlist)]

Arguments

<CPD>[, (@chanlist)]

<CPD>: POSitive|NEGative|EITHer

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

POSitive

Example

TRIGger:ACQUIRE:VOLTage:SLOPe NEGative,(@2)

Query Syntax

TRIGger:ACQUIRE:VOLTage:SLOPe? [(@chanlist)]

Returns

POSitive|NEGative|EITHer

TRIGger:ACQuire:VOLTage[:LEVel] <NRf+>[, (@chanlist)]

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

Group

TRIGger

Syntax

TRIGger:ACQuire:VOLTage[:LEVel] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:ACQuire:VOLTage[:LEVel] 100,(@2)

Query Syntax

TRIGger:ACQuire:VOLTage[:LEVel]? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:ACQuire:VOLTage:HYSTeresis:HIGH <NRf+>[,(@chanlist)]

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

Group

TRIGger

Syntax

TRIGger:ACQuire:VOLTage:HYSTeresis:HIGH <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:ACQuire:VOLTage:HYSTeresis:HIGH 100,(@2)

Query Syntax

TRIGger:ACQuire:VOLTage:HYSTeresis:HIGH? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:ACQuire:VOLTage:HYSTeresis:LOW <NRf+>[,(@chanlist)]

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

Group

TRIGger

Syntax

TRIGger:ACQuire:VOLTage:HYSTeresis:LOW <NRf+>[,(@chanlist)]

Arguments

`<NRf+>[,(@chanlist)]`
`<NRf+>: MIN|MAX|DEF|<MIN to MAX>`
`(@chanlist): (@1)|(@2)|(@1,2)`

Default Value

MIN

Example

`TRIGger:ACQuire:VOLTage:HYSTeresis:LOW 10,(@2)`

Query Syntax

`TRIGger:ACQuire:VOLTage:HYSTeresis:LOW? [MIN|MAX|DEF][,(@chanlist)]`

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:ACQuire:CURRent:SLOPe <CPD>[,(@chanlist)]

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>[,(@chanlist)]

Arguments

<CPD>[,(@chanlist)]

<CPD>: POSitive|NEGative|EITHer

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

POSitive

Example

TRIGger:ACQuire:CURRent:SLOPe NEGative,(@2)

Query Syntax

TRIGger:ACQuire:CURRent:SLOPe? [(@chanlist)]

Returns

POSitive|NEGative|EITHer

TRIGger:ACQuire:CURRent[:LEVel] <NRf+>[,(@chanlist)]

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

Group

TRIGger

Syntax

TRIGger:ACQuire:CURRent[:LEVel] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:ACQuire:CURRent[:LEVel] 10,(@2)

Query Syntax

TRIGger:ACQuire:CURRent[:LEVel]? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:ACQuire:CURRent:HYSTeresis:HIGH <NRf+>[,(@chanlist)]

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

Group

TRIGger

Syntax

TRIGger:ACQuire:CURRent:HYSTeresis:HIGH <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

```
TRIGger:ACQuire:CURRent:HYSTeresis:HIGH 10,(@2)
```

Query Syntax

```
TRIGger:ACQuire:CURRent:HYSTeresis:HIGH? [MIN|MAX|DEF][,]  
(@chanlist)]
```

Returns

```
MIN|MAX|DEF|<MIN to MAX>
```

TRIGger:ACQuire:CURRent:HYSTeresis:LOW <NRf+>[,(@chanlist)]

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

Group

```
TRIGger
```

Syntax

```
TRIGger:ACQuire:CURRent:HYSTeresis:LOW <NRf+>[,(@chanlist)]
```

Arguments

```
<NRf+>[,(@chanlist)]
```

```
<NRf+>: MIN|MAX|DEF|<MIN to MAX>
```

```
(@chanlist): (@1)|(@2)|(@1,2)
```

Default Value

```
MIN
```

Example

```
TRIGger:ACQuire:CURRent:HYSTeresis:LOW 1,(@2)
```

Query Syntax

TRIGger:ACQuire:CURRent:HYSteresis:LOW? [MIN|MAX|DEF][, (@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:ARB[:IMMediate] [(@chanlist)]

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] [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Example

TRIG:ARB (@2)

Query Syntax

None

Returns

None

TRIGger:ARB:SOURce <CPD>[,(@chanlist)]

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

- IMMEDIATE: A trigger is generated immediately.

- **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 **CON-Figurable** subsystem.

- **BUS**: Triggered by the command (***TRG**).
- **MANual**: Manual triggering, that is, the ARB subsystem function is triggered by the **TRIG:ARB**.

Group

TRIGger

Syntax

TRIGger:ARB:SOURce <CPD>[,(@chanlist)]

Arguments

<CPD>[,(@chanlist)]

<CPD>: IMMEDIATE|MANual|BUS|EXTernal

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MANual

Example

TRIGger:ARB:SOURce EXTernal,(@2)

Query Syntax

TRIGger:ARB:SOURce? [(@chanlist)]

Returns

IMMEDIATE|MANual|BUS|EXTernal

TRIGger:DLOG:SOURce <CPD>[,(@chanlist)]

This command is used to set the trigger source of the Dlog function.

- **IMMEDIATE**: Generate a trigger signal immediately.
- **EXTERNAL**: External trigger, that is, when pin 4 of the digital I/O interface receives a fixed pulse signal, the Dlog 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 introduction of the **CONFIGURABLE** subsystem.

- **BUS**: Command (***TRG**) triggered
- **MANUAL**: Manual trigger, that is, the command **TRIG:DLOG** generates a trigger.
- **VOLTAGE**: Voltage rising or falling edge trigger.
- **CURRENT**: Current rising or falling edge trigger.

Group

TRIGger

Syntax

TRIGger:DLOG:SOURce <CPD>[,(@chanlist)]

Parameter

<CPD>[,(@chanlist)]

<CPD>: **IMMEDIATE|MANUAL|BUS|VOLTAGE|CURRENT|EXTERNAL**

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MANUAL

Example

TRIG:DLOG:SOUR EXT,(@2)

Query Syntax

TRIGger:DLOG:SOURce? [(@chanlist)]

Returns

IMMediate|MANual|BUS|VOLTage|CURRent|EXTernal

TRIGger:DLOG[:IMMediate] [(@chanlist)]

This command is used to generate a trigger for the Dlog function.

Group

TRIGger

Syntax

TRIGger:DLOG[:IMMediate] [(@chanlist)]

Parameter

(@chanlist): (@1)|(@2)|(@1,2)

Example

TRIG:DLOG (@2)

Query Syntax

None

Returns

None

TRIGger:DLOG:CURRent:SLOPe <CPD>[, (@chanlist)]

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

- POSitive: rising edge
- NEGative: Falling edge

- EITHer: Either rising or falling edge


Note

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

Group

TRIGger

Syntax

TRIGger:DLOG:CURRent:SLOPe <CPD>[,(@chanlist)]

Parameter

<CPD>[,(@chanlist)]

<CPD>: POSitive|NEGative|EITHer

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

POSitive

Example

TRIGger:DLOG:CURRent:SLOPe NEGative,(@2)

Query Syntax

TRIGger:DLOG:CURRent:SLOPe? [(@chanlist)]

Returns

POSitive|NEGative|EITHer

TRIGger:DLOG:CURRent[:LEVeI] <NRf+>[, (@chanlist)]

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

Group

TRIGger

Syntax
TRIGger:DLOG:CURRent[:LEVel] <NRf+>[,(@chanlist)]
Parameter

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example
TRIGger:DLOG:CURRent[:LEVel] 10,@2
Query Syntax
TRIGger:DLOG:CURRent[:LEVel]? [MIN|MAX|DEF][,(@chanlist)]
Returns

MIN|MAX|DEF|<Value>

TRIGger:DLOG:CURRent:HYSTeresis:HIGH <NRf+>[,(@chanlist)]

This command is used to set the Dlog current trigger upper threshold.

Group

TRIGger

Syntax
TRIGger:DLOG:CURRent:HYSTeresis:HIGH <NRf+>[,(@chanlist)]

Parameter

<NRf+>[,(@chanlist)]
 <NRf+>: MIN|MAX|DEF|<MIN to MAX>
 (@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:DLOG:CURRent:HYSTeresis:HIGH 10,(@2)

Query Syntax

TRIGger:DLOG:CURRent:HYSTeresis:HIGH? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:DLOG:CURRent:HYSTeresis:LOW <NRf+>[,(@chanlist)]

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

Group

TRIGger

Syntax

TRIGger:DLOG:CURRent:HYSTeresis:LOW <NRf+>[,(@chanlist)]

Parameter

<NRf+>[,(@chanlist)]
 <NRf+>: MIN|MAX|DEF|<MIN to MAX>
 (@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:DLOG:CURREnt:HYSteresis:LOW 1,(@2)

Query Syntax

TRIGger:DLOG:CURREnt:HYSteresis:LOW? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:DLOG:VOLTage:SLOPe <CPD>[, (@chanlist)]

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

- POSitive: Rising edge
- NEGative: Falling edge
- EITHer: Either rising or falling edge



Note

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

Group

TRIGger

Syntax

TRIGger:DLOG:VOLTage:SLOPe <CPD>[,(@chanlist)]

Parameter

<CPD>[,(@chanlist)]

<CPD>: POSitive|NEGative|EITHer

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

POSitive

Example

TRIGger:DLOG:VOLTage:SLOPe NEGative,(@2)

Query Syntax

TRIGger:DLOG:VOLTage:SLOPe? [(@chanlist)]

Reruns

POSitive|NEGative|EITHer

TRIGger:DLOG:VOLTage[:LEVel] <NRf+>[, (@chanlist)]

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

Group

TRIGger

Syntax

TRIGger:DLOG:VOLTage[:LEVel] <NRf+>[,(@chanlist)]

Parameter

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:DLOG:VOLTage[:LEVel] 10,(@2)

Query Syntax

TRIGger:DLOG:VOLTage[:LEVel]? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:DLOG:VOLTage:HYSTeresis:HIGH <NRf +>[,(@chanlist)]

This command is used to set the Dlog voltage trigger upper threshold.

Group

TRIGger

Syntax

TRIGger:DLOG:VOLTage:HYSTeresis:HIGH <NRf+>[,(@chanlist)]

Parameter

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:DLOG:VOLTage:HYSTeresis:HIGH 80,(@2)

Query Syntax

TRIGger:DLOG:VOLTage:HYSTeresis:HIGH? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:DLOG:VOLTage:HYSTeresis:LOW <NRf+>[,(@chanlist)]

This command is used to set the Dlog voltage trigger lower threshold.

Group

TRIGger

Syntax

TRIGger:DLOG:VOLTage:HYSTeresis:LOW <NRf+>[,(@chanlist)]

Parameter

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:DLOG:VOLTage:HYSTeresis:LOW 1,(@2)

Query Syntax

TRIGger:DLOG:VOLTage:HYSTeresis:LOW? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:ELOG:SOURce <CPD>[,(@chanlist)]

This command is used to set the trigger source of the Elog function.

- IMMEDIATE: Generate a trigger signal immediately.
- EXTERNAL: External trigger, that is, when pin 4 of the digital I/O interface receives a fixed pulse signal, the Elog 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 introduction of the **CONFigurable** subsystem.

- **BUS**: Command (***TRG**) triggered
- **MANual**: Manual trigger, that is, command **TRIG:ELOG** to generate a trigger.
- **VOLTage**: Voltage rising or falling edge trigger.
- **CURRent**: Current rising or falling edge trigger.

Group

TRIGger

Syntax
TRIGger:ELOG:SOURce <CPD>[,(@chanlist)]

Parameter

<CPD>[,(@chanlist)]

 <CPD>: **IMMediate**|**MANual**|**BUS**|**VOLTage**|**CURRent**|**EXTernal**

(@chanlist): (@1)|(@2)|(@1,2)

Default Value
MANual
Example
TRIG:ELOG:SOUR EXT,(@2)
Query Syntax
TRIGger:ELOG:SOURce? [(@chanlist)]
Returns
IMMediate|**MANual**|**BUS**|**VOLTage**|**CURRent**|**EXTernal**

TRIGger:ELOG[:IMMediate] [(@chanlist)]

This command is used to generate a trigger for the Elog function.

Group

TRIGger

Syntax

TRIGger:ELOG[:IMMediate] [(@chanlist)]

Parameter

(@chanlist): (@1)|(@2)|(@1,2)

Example

TRIG:ELOG (@2)

Query Syntax

None

Returns

None

TRIGger:ELOG:CURRent:SLOPe <CPD>[, (@chanlist)]

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

- POSitive: Rising edge
- NEGative: Falling edge
- EITHer: Either rising or falling edge



Note

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

Group

TRIGger

Syntax**TRIGger:ELOG:CURRent:SLOPe <CPD>[,(@chanlist)]****Parameter**

<CPD>[,(@chanlist)]

<CPD>: POSitive|NEGative|EITHer

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

POSitive

Example**TRIGger:ELOG:CURRent:SLOPe NEGative,(@2)****Query Syntax****TRIGger:ELOG:CURRent:SLOPe? [(@chanlist)]****Returns**

POSitive|NEGative|EITHer

**TRIGger:ELOG:CURRent[:LEVe] <NRf+>[,
(@chanlist)]**

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

Group

TRIGger

Syntax**TRIGger:ELOG:CURRent[:LEVe] <NRf+>[,(@chanlist)]**

Parameter

<NRf+>[,(@chanlist)]
 <NRf+>: MIN|MAX|DEF|<MIN to MAX>
 (@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:ELOG:CURRent[:LEVel] 10,(@2)

Query Syntax

TRIGger:ELOG:CURRent[:LEVel]? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:ELOG:CURRent:HYSTeresis:HIGH <NRf+>[,(@chanlist)]

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

Group

TRIGger

Syntax

TRIGger:ELOG:CURRent:HYSTeresis:HIGH <NRf+>[,(@chanlist)]

Parameter

<NRf+>[,(@chanlist)]
 <NRf+>: MIN|MAX|DEF|<MIN to MAX>
 (@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:ELOG:CURRent:HYSTeresis:HIGH 10,(@2)

Query Syntax

TRIGger:ELOG:CURRent:HYSTeresis:HIGH? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:ELOG:CURRent:HYSTeresis:LOW <NRf +>[,(@chanlist)]

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

Group

TRIGger

Syntax

TRIGger:ELOG:CURRent:HYSTeresis:LOW <NRf+>[,(@chanlist)]

Parameter

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:ELOG:CURRent:HYSTeresis:LOW 1,(@2)

Query Syntax

TRIGger:ELOG:CURRent:HYSTeresis:LOW? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:ELOG:VOLTage:SLOPe <CPD>[, (@chanlist)]

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

- POSitive: Rising edge
- NEGative: Falling edge
- EITHer: Either rising or falling edge



Note

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

Group

TRIGger

Syntax

TRIGger:ELOG:VOLTage:SLOPe <CPD>[,(@chanlist)]

Parameter

<CPD>[,(@chanlist)]

<CPD>: POSitive|NEGative|EITHer

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

POSitive

Example

TRIGger:ELOG:VOLTage:SLOPe NEGative,(@2)

Query Syntax

TRIGger:ELOG:VOLTage:SLOPe? [(@chanlist)]

Returns

POSitive|NEGative|EITHer

TRIGger:ELOG:VOLTage[:LEVel] <NRf+>[, (@chanlist)]

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

Group

TRIGger

Syntax

TRIGger:ELOG:VOLTage[:LEVel] <NRf+>[,(@chanlist)]

Parameter

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:ELOG:VOLTage[:LEVel] 10,(@2)

Query Syntax

TRIGger:ELOG:VOLTage[:LEVel]? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<Value>

TRIGger:ELOG:VOLTage:HYSTeresis:HIGH <NRf+>[,(@chanlist)]

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

Group

TRIGger

Syntax

TRIGger:ELOG:VOLTage:HYSTeresis:HIGH <NRf+>[,(@chanlist)]

Parameter

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:ELOG:VOLTage:HYSTeresis:HIGH 80,(@2)

Query Syntax

TRIGger:ELOG:VOLTage:HYSTeresis:HIGH? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

TRIGger:ELOG:VOLTage:HYSTeresis:LOW <NRf+>[,(@chanlist)]

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

Group

TRIGger

Syntax

TRIGger:ELOG:VOLTage:HYSTeresis:LOW <NRf+>[,(@chanlist)]

Parameter

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

TRIGger:ELOG:VOLTage:HYSTeresis:LOW 1,(@2)

Query Syntax

TRIGger:ELOG:VOLTage:HYSTeresis:LOW? [MIN|MAX|DEF][,(@chanlist)]

Returns

MIN|MAX|DEF|<MIN to MAX>

10 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
OV	0	1	Overvoltage Protection
OC	1	2	Overcurrent Protection

OP	2	4	Overpower Protection
UV	3	8	Undervoltage Protection
OT	4	16	Over Temperature Protection.
UC	5	32	Undercurrent Protection.
Errsense	6	64	Sense Fault
Share	7	128	Current sharing fault.
Rvs	8	256	The output is reversed.
INH	9	512	Externally inhibited output.
PS	10	1024	Fault protection bit (protect shutdown).
OSC	11	2048	Loop oscillation failure.
UNR	12	4096	Unknown internal fault of the instrument.
Operation Status Bit			
Log_Mode	0	1	LOG mode(0=dlog, 1=elog)
OPT-WTG	1	2	Waiting for trigger (indicates the trigger status of meter, battery, and arb).
DLOG-WTG	2	4	Waiting for a trigger (indicates the trigger status of Dlog).
OPT_Pause	3	8	The current running function is in a paused state.
OPT-Active	4	16	Being executed (indicates the operating status of meter, battery, and arb).
DLOG-Active	5	32	DLOG has been triggered and is being executed.
OFF	6	64	The [on/off] of the instrument is off.
CC	7	128	Constant current
CV	8	256	Constant voltage
CP	9	512	Constant power

Bit description of the standard status register			
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]? [(@chanlist)]

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]? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<bit value>

Example

STATus:QUEStionable[:EVENT]? (@2)

Also see

None

STATus:QUEStionable:ENABLE <NR1>[, (@chanlist)]

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>[,(@chanlist)]

Arguments

<NR1>: A decimal value corresponding to the binary weighted sum of the register's bits. Setting range: 0 to 65535.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0

Example

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

Query Syntax

STATus:QUEStionable:ENABle? [(@chanlist)]

Returns

<NR1>

STATus:QUEStionable:PTRansition <NR1>[,(@chanlist)]

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>[,(@chanlist)]

Arguments

<NR1>: A decimal value corresponding to the binary weighted sum of the register's bits. Setting range: 0 to 65535.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0

Example

Enable bit 3 and 4 in the questionable PTR register: **STATus:QUEStionable:PTRansition 24,(@2)**

Query Syntax

STATus:QUEStionable:PTRansition? [(@chanlist)]

Returns

<NR1>

STATus:QUEStionable:NTRansition <NR1>[,(@chanlist)]

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>[,(@chanlist)]

Arguments

<NR1>: A decimal value corresponding to the binary weighted sum of the register's bits. Setting range: 0 to 65535.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0

Example

Enable bit 3 and 4 in the questionable NTR register: **STATus:QUEStionable:NTRansition 24,(@2)**

Query Syntax

STATus:QUEStionable:NTRansition? [(@chanlist)]

Returns

<NR1>

STATus:QUEStionable:CONDition? [(@chanlist)]

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? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Returns

<bit value>

Example

STATus:QUEStionable:CONDition? (@2)

Also see

None

STATus:OPERation[:EVENT]? [(@chanlist)]

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]? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<bit value>

Example

STATus:OPERation[:EVENT]? (@2)

Also see

None

STATus:OPERation:ENABLE <NR1>[,(@chanlist)]

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>[,(@chanlist)]

Arguments

<NR1>: A decimal value corresponding to the binary weighted sum of the register's bits. Setting range: 0 to 65535.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0

Example

Enable bit 3 and 4 in the enable register: **STATus:OPERation:ENABle 24,(@2)**

Query Syntax

STATus:OPERation:ENABle? [(@chanlist)]

Returns

<NR1>

STATus:OPERation:PTRansition <NR1>[, (@chanlist)]

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>[,(@chanlist)]

Arguments

<NR1>: A decimal value corresponding to the binary weighted sum of the register's bits. Setting range: 0 to 65535.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0

Example

Enable bit 3 and 4 in the PTR register: **STATus:OPERation:PTRansition 24, (@2)**

Query Syntax

STATus:OPERation:PTRansition? [(@chanlist)]

Returns

<NR1>

STATus:OPERation:NTRansition <NR1>[, (@chanlist)]

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>[,(@chanlist)]

Arguments

<NR1>: A decimal value corresponding to the binary weighted sum of the register's bits. Setting range: 0 to 65535.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0

Example

Enable bit 3 and 4 in the NTR register: **STATus:OPERation:NTRansition 24, (@2)**

Query Syntax

STATus:OPERation:NTRansition? [(@chanlist)]

Returns

<NR1>

STATus:OPERation:CONDition? [(@chanlist)]

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? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Returns

<bit value>

Example

STATus:OPERation:CONDition? (@2)

Also see

None

STATus:PRESet [(@chanlist)]

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 [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Returns

None

Example

Preset the Operation and Questionable registers: **STATus:PRESet (@2)**

Also see

None

11 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]? [(@chanlist)]

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

Group

FETCh

Syntax

FETCh[:SCALAr]:CURRent[:DC]? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETCh:CURR? (@1)

Also see

None

FETCh[:SCALar]:CURRent[:DC]:HIGH? [(@chanlist)]

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

Group

FETCh

Syntax

FETCh[:SCALar]:CURRent[:DC]:HIGH? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETC:CURR:HIGH? (@1)

Also see

None

FETCh[:SCALar]:CURRent[:DC]:LOW? [(@chanlist)]

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

Group

FETCh

Syntax

FETCh[:SCALar]:CURRent[:DC]:LOW? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETC:CURR:LOW? (@1)

Also see

None

FETCh[:SCALar]:CURRent[:DC]:MAXimum? [(@chanlist)]

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

Group

FETCh

Syntax

FETCh[:SCALar]:CURRent[:DC]:MAXimum? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETC:CURR:MAX? (@1)

Also see

None

FETCh[:SCALar]:CURRent[:DC]:MINimum? [(@chanlist)]

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

Group

FETCh

Syntax

FETCh[:SCALar]:CURRent[:DC]:MINimum? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETC:CURR:MIN? (@1)

Also see

None

FETCh[:SCALar]:VOLTage[:DC]? [(@chanlist)]

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

Group

FETCh

Syntax

FETCh[:SCALar]:VOLTage[:DC]? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETC:VOLT? (@1)

Also see

None

FETCh[:SCALar]:VOLTage[:DC]:HIGH? [(@chanlist)]

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

Group

FETCh

Syntax

FETCh[:SCALar]:VOLTage[:DC]:HIGH? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example**FETC:VOLT:HIGH? (@1)****Also see**

None

FETCh[:SCALar]:VOLTage[:DC]:LOW? [(@chanlist)]

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

Group

FETCh

Syntax**FETCh[:SCALar]:VOLTage[:DC]:LOW? [(@chanlist)]****Arguments**

None

Default Value

None

Returns

<NRf+>

Example

FETC:VOLT:LOW? (@1)

Also see

None

FETCh[:SCALar]:VOLTage[:DC]:MAXimum? [(@chanlist)]

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

Group

FETCh

Syntax

FETCh[:SCALar]:VOLTage[:DC]:MAXimum? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETC:VOLT:MAX? (@1)

Also see

None

FETCh[:SCALar]:VOLTage[:DC]:MINimum? [(@chanlist)]

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

Group

FETCh

Syntax

FETCh[:SCALar]:VOLTage[:DC]:MINimum? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

FETC:VOLT:MIN? (@1)

Also see

None

FETCh[:SCALar]:POWER[:DC]? [(@chanlist)]

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

Group

FETCh

Syntax

FETCh[:SCALar]:POWER[:DC]? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example**FETC:POW? (@1)****Also see**

None

FETCh[:SCALar]? [(@chanlist)]

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

Group

FETCh

Syntax**FETCh[:SCALar]? [(@chanlist)]****Arguments**

None

Default Value

None

Returns

<NRf+>

Example

FETC? (@1)

Also see

None

ELOG:STATe?

This command is used to query the running status of ELOG. 0 means stop, 1 means run, and 2 means waiting for trigger. If it returns to 2, use **trig:elog** to trigger the running of ELOG.

Group

ELOG

Syntax

ELOG:STATe?

Arguments

None

Default Value

None

Returns

<SRD>

Example

ELOG:STATe?

Related Commands

None

DLOG:STATe?

This command is used to query the running status of DLOG. 0 means stop, 1 means run, and 2 means waiting for trigger. If it returns to 2, use **TRIG:DLOG** to trigger the running of DLOG. The USB interface on the front panel and the USB interface on the rear panel cannot be occupied at the same time.

Group

DLOG

Syntax

DLOG:STATe?

Arguments

None

Default Value

None

Returns

<SRD>

Example

DLOG:STATe?

Related Commands

None

FETCh:ELOG:ARRay:DATA[:DC]? [(@chanlist)]

Query cache data.

Group

FETCh

Syntax

FETCh:ELOG:ARRay:DATA[:DC]? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<Block>

Example

FETC:ELOG:ARR:DATA? (@1)

Also see

None

12 MEASure Subsystem

Measure commands measure the output voltage or current. They trigger the acquisition of new data before returning the reading. Measurements are performed by digitizing the instantaneous output 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]? [(@chanlist)]

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

Group

MEASure

Syntax

MEASure[:SCALar]:CURRent[:DC]? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEAS:CURRE? (@2)

Also see

None

MEASure[:SCALar]:CURRent:HIGH? [(@chanlist)]

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

Group

MEASure

Syntax

MEASure[:SCALar]:CURRent:HIGH? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEAS:CURR:HIGH? (@2)

Also see

None

MEASure[:SCALar]:CURRent:LOW? [(@chanlist)]

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

Group

MEASure

Syntax

MEASure[:SCALar]:CURRent:LOW? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEAS:CURR:LOW? (@2)

Also see

None

MEASure[:SCALar]:CURRent:MAXimum? [(@chanlist)]

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

Group

MEASure

Syntax

MEASure[:SCALar]:CURRent:MAXimum? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example**MEAS:CURR:MAX? (@2)****Also see**

None

MEASure[:SCALar]:CURRent:MINimum? [(@chanlist)]

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

Group

MEASure

Syntax**MEASure[:SCALar]:CURRent:MINimum? [(@chanlist)]****Arguments**

None

Default Value

None

Returns

<NRf+>

Example**MEAS:CURR:MIN? (@2)****Also see**

None

MEASure[:SCALar]:VOLTage[:DC]? [(@chanlist)]

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

Group

MEASure

Syntax

MEASure[:SCALar]:VOLTage[:DC]? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEAS:VOLT? (@2)

Also see

None

MEASure[:SCALar]:VOLTage:HIGH? [(@chanlist)]

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

Group

MEASure

Syntax

MEASure[:SCALar]:VOLTage:HIGH? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEAS:VOLT:HIGH? (@2)

Also see

None

MEASure[:SCALar]:VOLTage:LOW? [(@chanlist)]

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

Group

MEASure

Syntax

MEASure[:SCALar]:VOLTage:LOW? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example**MEAS:VOLT:LOW? (@2)****Also see**

None

MEASure[:SCALar]:VOLTage:MAXimum? [(@chanlist)]

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

Group

MEASure

Syntax**MEASure[:SCALar]:VOLTage:MAXimum? [(@chanlist)]****Arguments**

None

Default Value

None

Returns

<NRf+>

Example**MEAS:VOLT:MAX? (@2)****Also see**

None

MEASure[:SCALar]:VOLTage:MINimum? [(@chanlist)]

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

Group

MEASure

Syntax

MEASure[:SCALar]:VOLTage:MINimum? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEAS:VOLT:MIN? (@2)

Also see

None

MEASure[:SCALar]:POWer[:DC]? [(@chanlist)]

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

Group

MEASure

Syntax

MEASure[:SCALar]:POWer[:DC]? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<NRf+>

Example

MEAS:POW? (@2)

Also see

None

MEASure:ARRay:CURRent[:DC]? [(@chanlist)]

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]? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<Block>

Example

Returns the measured current array: **MEAS:ARR:CURR? (@2)**

Also see

None

MEASure:ARRay:VOLTage[:DC]? [(@chanlist)]

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]? [(@chanlist)]

Arguments

None

Default Value

None

Returns

<Block>

Example

Returns the measured voltage array: **MEAS:ARR:VOLT? (@2)**

Also see

None

MEASure[:SCALar]? [(@chanlist)]

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

Group

MEASure

Syntax**MEASure[:SCALar]? [(@chanlist)]****Arguments**

None

Default Value

None

Returns

<NRf+>

Example**MEAS? (@2)****Also see**

None

13 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>

14 SENSE Subsystem

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

[SOURce:]REMOte:SENSe[:STATe] <Bool>[, (@chanlist)]

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>[, (@chanlist)]

Arguments

<Bool>[, (@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Example

REM:SENS 1,(@2)

Query Syntax

[SOURce:]REMOte:SENSe[:STATe]? [(@chanlist)]

Returns

<Bool>

SENSe:ELOG:FUNction:CURRent <Bool>[, (@chanlist)]

This command is used to control whether to enable the current data buffer recording function.

Group

SENSe

Syntax

SENSe:ELOG:FUNction:CURRent <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1|ON

Example

SENS:ELOG:FUNC:CURR 1,(@2)

Query Syntax

SENSe:ELOG:FUNction:CURRent? [(@chanlist)]

Returns

<Bool>

SENSE:ELOG:FUNCTION:VOLTage <Bool>[, (@chanlist)]

This command is used to control whether to enable the function of voltage data buffer recording.

Group

SENSe

Syntax

SENSe:ELOG:FUNCTION:VOLTage <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1|ON

Example

SENSe:ELOG:FUNC:VOLT 1,(@2)

Query Syntax

SENSe:ELOG:FUNCTION:VOLTage? [(@chanlist)]

Returns

<Bool>

SENSe:ELOG:FUNCTION:POWER <Bool>[, (@chanlist)]

This command is used to control whether to enable the power data buffer recording function.

Group

SENSe

Syntax

SENSe:ELOG:FUNCtion:POWER <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1|ON

Example

SENS:ELOG:FUNC:POW 1,(@2)

Query Syntax

SENSe:ELOG:FUNCtion:POWER? [(@chanlist)]

Returns

<Bool>

SENSe:ELOG:FUNCtion:PEAK <Bool>[,(@chanlist)]

This command is used to control whether to enable the function of buffering and recording the maximum and minimum values of voltage and current.

Group

SENSe

Syntax

SENSe:ELOG:FUNCtion:PEAK <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]
 <Bool>: 0|OFF|1|ON
 (@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Example

SENSE:ELOG:FUNC:PEAK 1,(@2)

Query Syntax

SENSe:ELOG:FUNction:PEAK? [(@chanlist)]

Returns

<Bool>

SENSe:ELOG:PERiod <NRf+>[,(@chanlist)]

This command is used to set the data sampling interval of the ELOG buffer. The default value is 0.001S. The setting range is 0.0001-100. Unit: seconds.

Group

SENSe

Syntax

SENSe:ELOG:PERiod <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]
 <NRf+>: MIN|MAX|DEF|<MIN~MAX>
 (@chanlist): (@1)|(@2)|(@1,2)

Default Value

0.001S

Example

SENS:ELOG:PER 0.1,(@2)

Query Syntax

SENSe:ELOG:PERiod? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

SENSe:ELOG:TIME <NRf+>[,(@chanlist)]

This command is used to set the duration of ELOG cache. The default value is 0S, which means to keep caching until it receives an instruction **ABORT:ELOG** to stop caching . Unit: seconds.

Group

SENSe

Syntax

SENSe:ELOG:TIME <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0

Example

SENS:ELOG:TIME 100,(@2)

Query Syntax

SENSe:ELOG:TIME? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

SENSe:DLOG:FUNCTION:CURRENT <Bool>[, (@chanlist)]

This command is used to control whether to enable the current data recording function. Before using DLOG, a USB flash drive must be inserted into the front panel of the instrument.

Group

SENSe

Syntax

SENSe:DLOG:FUNCTION:CURRENT <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1|ON

Example

SENS:DLOG:FUNC:CURR 1,(@2)

Query Syntax

SENSe:DLOG:FUNCTION:CURRENT? [(@chanlist)]

Returns

<Bool>

SENSe:DLOG:FUNCtion:VOLTage <Bool>[, (@chanlist)]

This command is used to control whether to enable the function of voltage data recording. Before using DLOG, a USB flash drive must be inserted into the front panel of the instrument.

Group

SENSe

Syntax

SENSe:DLOG:FUNCtion:VOLTage <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1|ON

Example

SENS:DLOG:FUNC:VOLT 1,(@2)

Query Syntax

SENSe:DLOG:FUNCtion:VOLTage? [(@chanlist)]

Returns

<Bool>

SENSe:DLOG:FUNCtion:POWer <Bool>[, (@chanlist)]

This command is used to control whether to enable the function of power data recording. Before using DLOG, a USB flash drive must be inserted into the front panel of the instrument.

Group

SENSe

Syntax

SENSe:DLOG:FUNCtion:POWer <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1|ON

Example

SENS:DLOG:FUNC:POW 1,(@2)

Query Syntax

SENSe:DLOG:FUNCtion:POWer? [(@chanlist)]

Returns

<Bool>

SENSe:DLOG:FUNCtion:PEAK <Bool>[, (@chanlist)]

This command is used to control whether to enable the function of recording the maximum and minimum values of voltage and current. Before using DLOG, a USB flash drive must be inserted into the front panel of the instrument.

Group

SENSe

Syntax

SENSe:DLOG:FUNCtion:PEAK <Bool>[,(@chanlist)]

Arguments

<Bool>[,(@chanlist)]

<Bool>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Example

SENS:DLOG:FUNC:PEAK 1,(@2)

Query Syntax

SENSe:DLOG:FUNCtion:PEAK? [(@chanlist)]

Returns

<Bool>

SENSe:DLOG:PERiod <NRf+>[,(@chanlist)]

This command is used to set the sampling interval of the data logging function. The default value is 0.001s. Setting range: 0.0001-1000. Unit: seconds.

Group

SENSe

Syntax

SENSe:DLOG:PERiod <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0.001s

Example

SENS:DLOG:PER 0.1,(@2)

Query Syntax

SENSe:DLOG:PERiod? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

SENSe:DLOG:TIME <NRf+>[,(@chanlist)]

This command is used to set the duration of DLOG data recording. The default value is 0S, which means to keep recording until it receives a command **ABORT:DLOG** to stop recording. Unit: seconds.

Group

SENSe

Syntax

SENSe:DLOG:TIME <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]
 <NRf+>: MIN|MAX|DEF|<MIN~MAX>
 (@chanlist): (@1)|(@2)|(@1,2)

Default Value

0

Example

SENS:DLOG:TIME 100,@2

Query Syntax

SENSe:DLOG:TIME? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

SENSe:FILTer:LEVel <CPD>[,(@chanlist)]

This command is used to set the measurement frequency.

Group

SENSe

Syntax

SENSe:FILTer:LEVel <CPD>[,(@chanlist)]

Arguments

<CPD>[,(@chanlist)]
 <CPD>: SLOW|MEDIum|FAST
 (@chanlist): (@1)|(@2)|(@1,2)

Default Value

Med

Example

SENS:FILT:LEV MED,(@2)

Query Syntaxes

SENSe:FILTer:LEVel? [(@chanlist)]

Returns

SLOW|MEDIum|FAST

[SOURce:]REMOte:SENSe:THReshold <level>[, (@chanlist)]

Set the threshold for Sense protection.

Group

SENSe

Syntax

[SOURce:]REMOte:SENSe:THReshold <level>[, (@chanlist)]

Arguments

<level>[, (@chanlist)]

<level>: MINimum|MAXimum|DEFault|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Example

[SOURce:]REMOte:SENSe:THReshold 2,(@2)

Query Syntaxes

[SOURce:]REMOte:SENSe:THReshold? [(@chanlist)]

Returns

<NRf>

15 ARB Subsystem

The ARB subsystem contains instructions for setting various waveform-related parameters, including: pulse waveform, custom waveform (LIST function), sine wave, CDWell waveform (i.e. constant-dwell arbitrary waveform).

ARB:COUNT <NRf+>[,(@chanlist)]

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+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1

Example

Programs a repeat count of 10: **ARB:COUNT 10,(@2)**

Query Syntax

ARB:COUNT? [MIN|MAX][,(@chanlist)]

Returns

<NRf+>

ARB:TERMinate:LAST <CPD>[,(@chanlist)]

Selects the output setting and controls whether [On/Off] is set to OFF after the ARB ends.

Group

ARB

Syntax

ARB:TERMinate:LAST <CPD>[,(@chanlist)]

Arguments

<CPD>[,(@chanlist)]

<CPD>: NORMal|LAST|OFF|0|1|2

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|NORMal

Example

ARB:TERMinate:LAST 1,(@2)

Query Syntax

ARB:TERMinate:LAST? [(@chanlist)]

Returns

0|1|2

ARB:FUNcTion:TYPE <CPD>[,(@chanlist)]

Specifies either a voltage or current ARB. Only one type of ARB may be output at a time. The selection must match the CC or CV priority mode.

- CURRent: Current ARB
- VOLTage: Voltage ARB

Group

ARB

Syntax

ARB:FUNCTION:TYPE <CPD>[,(@chanlist)]

Arguments

<CPD>[,(@chanlist)]

<CPD>: CURRent|VOLTage

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

VOLTage

Example

ARB:FUNCTION:TYPE CURRent,(@1)

Query Syntax

ARB:FUNCTION:TYPE? [(@chanlist)]

Returns

CURRent|VOLTage

ARB:FUNCTION:SHAPE <CPD>[,(@chanlist)]

This command is used to set the function of the ARB.

- PULSe: Pulse waveform (square wave)
- SINusoid: Sine Wave
- SWEepsine: Sweep sine
- UDEFined: Custom Waveforms (LIST)
- CDWell: Constantly Residing Arbitrary Waveforms
- SEQuence: Sequence setting, which can be a combination of the above several waveforms.

Group

ARB

Syntax
ARB:FUNCTION:SHAPE <CPD>[,(@chanlist)]
Arguments

<CPD>[,(@chanlist)]

<CPD>: PULSe|SINusoid|SWEepsine|UDEFinEd|CDWell|SEQUence

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Example
ARB:FUNCTION:SHAPE CDWell,(@2)
Query Syntax
ARB:FUNCTION:SHAPE? [(@chanlist)]
Returns

PULSe|SINusoid|SWEepsine|UDEFinEd|CDWell|SEQUence

ARB:PULSe:START[:LEVeL] <NRf+>[,(@chanlist)]

This command is used to set the pulse signal starting amplitude.

Group

ARB

Syntax
ARB:PULSe:START[:LEVeL] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]
 <NRf+>: MIN|MAX|DEF|<MIN~MAX>
 (@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:PULSe:STARt[:LEVel] 10,(@2)

Query Syntax

ARB:PULSe:STARt[:LEVel]? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:PULSe:STARt:TIME <NRf+>[,(@chanlist)]

This command is used to set the delay of the initial phase of the pulse, in ms.

Group

ARB

Syntax

ARB:PULSe:STARt:TIME <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]
 <NRf+>: MIN|MAX|DEF|<MIN~MAX>
 (@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:PULSe:STARt:TIME 1,(@2)

Query Syntax

ARB:PULSe:STARt:TIME? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:PULSe:STARt:SLEW <NRf+>[,(@chanlist)]

This command is used to set the pulse start slope.

Group

ARB

Syntax

ARB:PULSe:STARt:SLEW <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:PULSe:STARt:SLEW 1,(@2)

Query Syntax

ARB:PULSe:STARt:SLEW? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:PULSe:TOP[:LEVeI] <NRf+>[,(@chanlist)]

This command is used to set the amplitude at the top of the pulse signal (waveform).

Group

ARB

Syntax

ARB:PULSe:TOP[:LEVeI] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:PULSe:TOP[:LEVeI] MAX,(@2)

Query Syntax

ARB:PULSe:TOP[:LEVeI]? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:PULSe:TOP:TIME <NRf+>[,(@chanlist)]

This command is used to set the time width of the top (high level) of the pulse signal.

Group

ARB

Syntax

ARB:PULSe:TOP:TIME <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:PULSe:TOP:TIME MAX,(@2)

Query Syntax

ARB:PULSe:TOP:TIME? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:PULSe:END:TIME <NRf+>[,(@chanlist)]

This command is used to set the end delay of the pulse signal.

Group

ARB

Syntax

ARB:PULSe:END:TIME <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:PULSe:END:TIME MAX,@2

Query Syntax

ARB:PULSe:END:TIME? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:PULSe:END:SLEW <NRf+>[,(@chanlist)]

This command is used to set the slope of the end of the pulse signal.

Group

ARB

Syntax

ARB:PULSe:END:SLEW <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:PULSe:END:SLEW MAX,(@2)

Query Syntax

ARB:PULSe:END:SLEW? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SINusoid:AMPLitude <NRf+>[,(@chanlist)]

This command is used to set the amplitude of the sinusoidal signal.

Group

ARB

Syntax

ARB:SINusoid:AMPLitude <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SINusoid:AMPLitude MAX,(@2)

Query Syntax

ARB:SINusoid:AMPLitude? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SINusoid:OFFSet <NRf+>[,(@chanlist)]

This command is used to set the offset of the sinusoidal signal.

Group

ARB

Syntax

ARB:SINusoid:OFFSet <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SINusoid:OFFSet MAX,@2

Query Syntax

ARB:SINusoid:OFFSet? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SINusoid:FREQuency <NRf+>[,(@chanlist)]

This command is used to set the frequency of the sinusoidal signal.

Group

ARB

Syntax

ARB:SINusoid:FREQuency <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SINusoid:FREQuency MAX,(@2)

Query Syntax

ARB:SINusoid:FREQuency? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SWEep:AMPLitude <NRf+>[,(@chanlist)]

This command is used to set the amplitude of the sweep wave.

Group

ARB

Syntax

ARB:SWEep:AMPLitude <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SWEep:AMPLitude MAX,@2

Query Syntax

ARB:SWEep:AMPLitude? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SWEep:OFFSet <NRf+>[,(@chanlist)]

This command is used to set the offset of the sweep wave.

Group

ARB

Syntax

ARB:SWEep:OFFSet <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SWEep:OFFSet MAX,(@2)

Query Syntax

ARB:SWEep:OFFSet? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SWEep:STARt <NRf+>[,(@chanlist)]

This command is used to set the starting frequency of the sweep wave.

Group

ARB

Syntax

ARB:SWEep:STARt <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SWEep:STARt MAX,(@2)

Query Syntax

ARB:SWEep:STARt? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SWEep:END <NRf+>[,(@chanlist)]

This command is used to set the end frequency of the sweep wave.

Group

ARB

Syntax

ARB:SWEep:END <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SWEep:END MAX,@2

Query Syntax

ARB:SWEep:END? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SWEep:STEP[:LEVel] <NRf+>[,(@chanlist)]

This command is used to set the step value of the sweep wave.

Group

ARB

Syntax

ARB:SWEep:STEP[:LEVel] <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SWEep:STEP[:LEVel] MAX,@2

Query Syntax

ARB:SWEep:STEP[:LEVel]? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SWEep:STEP:WIDTh <NRf+>[,(@chanlist)]

This command is used to set the time width of a single step of the sweep wave.
Unit: seconds.

Group

ARB

Syntax

ARB:SWEep:STEP:WIDTh <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SWEep:STEP:WIDTh MAX,(@2)

Query Syntax

ARB:SWEep:STEP:WIDTh? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:UDEFined:COUNT <NR1>[,(@chanlist)]

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>[,(@chanlist)]

Arguments

<NR1>, setting range: 1 to 200.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1

Example

ARB:UDEFined:COUNT 6,(@2)

Query Syntax

ARB:UDEFined:COUNT? [(@chanlist)]

Returns

<NR1>

ARB:UDEFined:LEVel <NR1>,<NRf+>[,(@chanlist)]

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+>[,(@chanlist)]

Arguments

<NR1>, used to specify which step of the custom waveform, setting range: 1 to 200.

<NRf+>: MIN|MAX|DEF|<MIN to MAX>. Used to specify the voltage/current value of the present step.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1,MIN

Example

ARB:UDEFined:LEVel 2,10,(@2)

Query Syntax

ARB:UDEFined:LEVel? <NR1>[,(@chanlist)]

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

Returns

<NRf+>

ARB:UDEFined:DWELI <NR1>,<NRf+>[,(@chanlist)]

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+>[,(@chanlist)]

Arguments

<NR1>, used to specify which step of the custom waveform, setting range: 1 to 200.

<NRf+>: MIN|MAX|DEF|<MIN to MAX>. Used to specify the time width of the present step.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1,MIN

Example

ARB:UDEFined:DWELI 2,3,(@2)

Query Syntax

ARB:UDEFined:DWELI? <NR1>[,(@chanlist)]

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

Returns

<NRf+>

ARB:UDEFined:SLEW <NR1>,<NRf+>[,(@chanlist)]

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+>[,(@chanlist)]

Arguments

<NR1>, used to specify which step of the custom waveform, setting range: 1 to 200.

<NRf+>: MIN|MAX|DEF|<MIN to MAX>. Used to specify the slope of the present step.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1,MIN

Example

ARB:UDEFined:SLEW 2,0.5,@2

Query Syntax

ARB:UDEFined:SLEW? <NR1>[,(@chanlist)]

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

Returns

<NRf+>

ARB:UDEFined:TRlgout <CPD>[,(@chanlist)]

This command is used to control whether the external trigger function is turned on. The external trigger function is applicable to multiple units synchronous control scenarios, that is, by connecting the outer ring fiber interfaces TX and RX, the List synchronization trigger between multiple units is realized.

- 0|OFF: No
- 1|ON: Yes

Group

ARB

Syntax

ARB:UDEFined:TRlgout <CPD>[,(@chanlist)]

Arguments

<CPD>[,(@chanlist)]

<CPD>: 0|OFF|1|ON

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0|OFF

Example

ARB:UDEFined:TRlgout 1,@2

Query Syntax

ARB:UDEFined:TRlgout? [(@chanlist)]

Returns

<CRD>

ARB:CDWell:POINTs <NR1>[,(@chanlist)]

This command is used to set the number of points of the CDWell waveform. ARB commands program the constant-dwell arbitrary waveforms. Constant-dwell waveforms can have up to 65,535 points assigned to them, with the same dwell time for each point.

Group

ARB

Syntax**ARB:CDWell:POINTs <NR1>[,(@chanlist)]****Arguments**

<NR1>, setting range: 1 to 65535.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1

Example**ARB:CDWell:POINTs 6,(@2)****Query Syntax****ARB:CDWell:POINTs? [(@chanlist)]****Returns**

<NR1>

ARB:CDWell[:LEVel] <NRf+>,<NRf+>,<NR1>[, (@chanlist)]

This command is used to set the high and low amplitude of a step (point) in the CDWell waveform, that is, the level of a certain point in the ARB, and the value is specified in amps or volts. The minimum and maximum values depend on the device rating. The command contains 3 parameters, the first <NRf+> indicates a high amplitude, the second <NRf+> indicates a low amplitude, the third parameter <NR1> indicates the specified step (point), and the setting value of the first parameter must be greater than the second parameter.

Group

ARB

Syntax

ARB:CDWell[:LEVel] <NRf+>,<NRf+>,<NR1>[,(@chanlist)]

Arguments

<NRf+>: MIN|MAX|DEF|<MIN to MAX>. Used to specify the high/low amplitude of a point.

<NR1>, used to specify which point of the CDWell waveform, range: 1 to 65535.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MAX,MIN,1

Example

ARB:CDWell[:LEVel] 100,10,2,(@2)

Query Syntax

ARB:CDWell[:LEVel]? <NR1>[,(@chanlist)]

Query by specifying a point number to return the high and low amplitudes corresponding to the point in the CDWell waveform.

Returns

<NRf+>,<NRf+>

ARB:CDWell:DWELI <NRf+>[,(@chanlist)]

This command is used to set the time width of each point in the CDWell waveform. That is, the dwell time of each point, in seconds.

Group

ARB

Syntax

ARB:CDWell:DWELI <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:CDWell:DWELI 5,(@2)

Query Syntax

ARB:CDWell:DWELI? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:CDWell:STOP [(@chanlist)]

This command is used to terminate the output of the CDWell waveform and terminates after all the transmitted data has been executed. Unlike the **ABORT:** **ARB** command, **ABORT:ARB** is stopped immediately.

Group

ARB

Syntax
ARB:CDWell:STOP [(@chanlist)]
Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Example
ARB:CDWell:STOP (@2)
Query Syntax

None

Returns

None

ARB:SEQuence:LENGth <NR1>[,(@chanlist)]

This command is used to set the total length of the Sequence, which is the total number of steps including multiple waveforms.

Group

ARB

Syntax
ARB:SEQuence:LENGth <NR1>[,(@chanlist)]
Arguments

<NR1>, setting range: 1 to 200.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1

Example

ARB:SEQuence:LENGth 10,(@2)

Query Syntax

ARB:SEQuence:LENGth? [(@chanlist)]

Returns

<NR1>

ARB:SEQuence:SElect <NR1>[,(@chanlist)]

This command is used to specify which waveform in the Sequence will be edited. This value cannot exceed the sequence length set in **ARB:SEQuence:LENGth <NR1>[,(@chanlist)]**.

Group

ARB

Syntax

ARB:SEQuence:SElect <NR1>[,(@chanlist)]

Arguments

<NR1>, setting range: 1 to 200.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1

Example

ARB:SEQuence:SElect 8,(@2)

Query Syntax

ARB:SEquence:SElect? [(@chanlist)]

Returns

<NR1>

ARB:SEquence:COUNt <NR1>[,(@chanlist)]

This command is used to set the number of repetitions of the Nth step selected in the Sequence.

Group

ARB

Syntax

ARB:SEquence:COUNt <NR1>[,(@chanlist)]

Arguments

<NR1>, setting range: 1 to 65535.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1

Example

ARB:SEquence:COUNt 3,(@2)

Query Syntax

ARB:SEquence:COUNt? [(@chanlist)]

Returns

<NR1>

ARB:SEQuence:FUNcTion <CPD>[,(@chanlist)]

This command is used to set the waveform type of the Nth step that is selected in the Sequence.

- CDArb: CDWell waveform
- SINusoid: Sinusoidal waveform
- UDEFined: User-defined LIST waveform
- PULSe: Pulse waveform

Group

ARB

Syntax

ARB:SEQuence:FUNcTion <CPD>[,(@chanlist)]

Arguments

<CPD>[,(@chanlist)]

<CPD>: CDArb|SINusoid|UDEFined|PULSe

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

None

Example

ARB:SEQuence:FUNcTion UDEFined,(@2)

Query Syntax

ARB:SEQuence:FUNcTion? [(@chanlist)]

Returns

CDArb|SINusoid|UDEFined|PULSe

ARB:SEQuence:PACing <CPD>[,(@chanlist)]

This command is used to control whether the Nth step selected in the Sequence is to generate a trigger signal or wait for the trigger to continue the loop after each loop is completed. Applicable to the scenario where multiple units are connected in parallel, that is, by connecting the outer ring fiber interfaces TX and RX, the trigger signal generated by the master triggers the operation of other slaves to realize the sequence synchronization trigger between multiple units.

- DWELI: Receive trigger signal
- TRIG: Generate trigger signal

Group

ARB

Syntax

ARB:SEQuence:PACing <CPD>[,(@chanlist)]

Arguments

<CPD>: DWELI|TRIG

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

DWELI

Example

ARB:SEQuence:PACing TRIG,(@2)

Query Syntax

ARB:SEQuence:PACing? [(@chanlist)]

Returns

DWELI|TRIG

ARB:SEQuence:PULSe:STARt[:LEVel] <NRf+>[, (@chanlist)]

This command is used to set the initial amplitude of the pulse signal of the selected Nth step of the Sequence. The premise is that **ARB:SEQuence:FUNCTion** has been set to **PULSe**, and the subsequent pulse signal related instructions are the same, and will not be explained again in this manual.

Group

ARB

Syntax

ARB:SEQuence:PULSe:STARt[:LEVel] <NRf+>[, (@chanlist)]

Arguments

<NRf+>[, (@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SEQuence:PULSe:STARt[:LEVel] 10,(@2)

Query Syntax

ARB:SEQuence:PULSe:STARt[:LEVel]? [MIN|MAX|DEF][, (@chanlist)]

Returns

<NRf+>

ARB:SEQuence:PULSe:STARt:TIME <NRf+>[, (@chanlist)]

This command is used to set the delay time of the initial pulse of the selected Nth step of the Sequence, in ms.

Group

ARB

Syntax

ARB:SEQuence:PULSe:STARt:TIME <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SEQuence:PULSe:STARt:TIME 10,(@2)

Query Syntax

ARB:SEQuence:PULSe:STARt:TIME? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SEQuence:PULSe:STARt:SLEW <NRf+>[, (@chanlist)]

This command is used to set the pulse start slope of the selected Nth step of the Sequence.

Group

ARB

Syntax
ARB:SEQuence:PULSe:STARt:SLEW <NRf+>[,(@chanlist)]
Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example
ARB:SEQuence:PULSe:STARt:SLEW 1,(@2)
Query Syntax
ARB:SEQuence:PULSe:STARt:SLEW? [MIN|MAX|DEF][,(@chanlist)]
Returns

<NRf+>

ARB:SEQuence:PULSe:TOP:LEVel <NRf+>[,(@chanlist)]

This command is used to set the amplitude at the top of the pulse signal (waveform) of the selected Nth step of the Sequence.

Group

ARB

Syntax
ARB:SEQuence:PULSe:TOP:LEVel <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]
 <NRf+>: MIN|MAX|DEF|<MIN~MAX>
 (@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SEQuence:PULSe:TOP:LEVel MAX,(@2)

Query Syntax

ARB:SEQuence:PULSe:TOP:LEVel? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SEQuence:PULSe:TOP:TIME <NRf+>[,(@chanlist)]

This command is used to set the top (high level) time width of the pulse signal of the selected Nth step of the Sequence.

Group

ARB

Syntax

ARB:SEQuence:PULSe:TOP:TIME <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]
 <NRf+>: MIN|MAX|DEF|<MIN~MAX>
 (@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SEQUence:PULSe:TOP:TIME MAX,(@2)

Query Syntax

ARB:SEQUence:PULSe:TOP:TIME? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SEQUence:PULSe:END:TIME <NRf+>[,(@chanlist)]

This command is used to set the pulse signal end delay of the selected Nth step of the Sequence.

Group

ARB

Syntax

ARB:SEQUence:PULSe:END:TIME <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SEQUence:PULSe:END:TIME MAX,(@2)

Query Syntax

ARB:SEQuence:PULSe:END:TIME? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SEQuence:PULSe:END:SLEW <NRf+>[,(@chanlist)]

This command is used to set the end slope of the pulse signal of the selected Nth step of the Sequence.

Group

ARB

Syntax

ARB:SEQuence:PULSe:END:SLEW <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SEQuence:PULSe:END:SLEW MAX,(@2)

Query Syntax

ARB:SEQuence:PULSe:END:SLEW? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SEquence:SINusoid:AMPLitude <NRf+>[, (@chanlist)]

This command is used to set the amplitude of the sinusoidal signal of the selected Nth step of the Sequence. The premise is that **ARB:SEquence:FUNCTION** has been set to **SINusoid**, and the subsequent sine signal related instructions are the same, and will not be explained again in this manual.

Group

ARB

Syntax

ARB:SEquence:SINusoid:AMPLitude <NRf+>[, (@chanlist)]

Arguments

<NRf+>[, (@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SEquence:SINusoid:AMPLitude MAX,(@2)

Query Syntax

ARB:SEquence:SINusoid:AMPLitude? [MIN|MAX|DEF][, (@chanlist)]

Returns

<NRf+>

ARB:SEquence:SINusoid:OFFSet <NRf+>[, (@chanlist)]

This command is used to set the offset of the sinusoidal signal of the selected Nth step of the Sequence.

Group

ARB

Syntax

ARB:SEquence:SINusoid:OFFSet <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SEquence:SINusoid:OFFSet MAX,(@2)

Query Syntax

ARB:SEquence:SINusoid:OFFSet? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SEquence:SINusoid:FREQuency <NRf+>[, (@chanlist)]

This command is used to set the frequency of the sinusoidal signal of the selected Nth step of the Sequence.

Group

ARB

Syntax

ARB:SEquence:SINusoid:FREQuency <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN~MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SEquence:SINusoid:FREQuency MAX,(@2)

Query Syntax

ARB:SEquence:SINusoid:FREQuency? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

ARB:SEquence:UDEFinEd:COUNt <count>[,(@chanlist)]

Set the number of steps in the list within the sequence.

Group

ARB

Syntax

ARB:SEquence:UDEFinEd:COUNt <count>[,(@chanlist)]

Arguments

<count>: <1 ~ 200>|MINimum|MAXimum|DEFault

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1

Example

ARB:SEQuence:UDEfined:COUnT 9,@2

Query Syntax

ARB:SEQuence:UDEfined:COUnT? [(@chanlist)]

Returns

<NR1>

ARB:SEQuence:UDEfined:LEVel <NR1>,<NRf+> [,(@chanlist)]

This command is used to set the amplitude corresponding to the X step of the user-defined waveform of the selected Nth step of the Sequence. The premise is that **ARB:SEQuence:FUnCtion** has been set to **UDEfined**, and subsequent user-defined waveform related instructions are the same, and will not be explained again in this manual.

Group

ARB

Syntax

ARB:SEQuence:UDEfined:LEVel <NR1>,<NRf+>[,(@chanlist)]

Arguments

<NR1>, used to specify which step of the user-defined waveform, the setting range is 1 to 200.

<NRf+>: MIN|MAX|DEF|<MIN to MAX>. Used to set the voltage/current value of the specified step.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1,MIN

Example

ARB:SEQuence:UDEFined:LEVel 2,10,(@2)

Query Syntax

ARB:SEQuence:UDEFined:LEVel? <NR1>[,(@chanlist)]

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

Returns

<NRf+>

ARB:SEQuence:UDEFined:DWELI <NR1>,<NRf+>[,(@chanlist)]

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

Group

ARB

Syntax

ARB:SEQuence:UDEFined:DWELI <NR1>,<NRf+>[,(@chanlist)]

Arguments

<NR1>, used to specify which step of the user-defined waveform, the setting range is 1 to 200.

<NRf+>: MIN|MAX|DEF|<MIN to MAX>. Used to set the time width of the specified step.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1,MIN

Example

ARB:SEquence:UDEFined:DWELI 2,3,@2)

Query Syntax

ARB:SEquence:UDEFined:DWELI? <NR1>[,(@chanlist)]

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

Returns

<NRf+>

ARB:SEquence:UDEFined:SLEW <NR1>,<NRf+>[,(@chanlist)]

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

Group

ARB

Syntax

ARB:SEquence:UDEFined:SLEW <NR1>,<NRf+>[,(@chanlist)]

Arguments

<NR1>, used to specify which step of the user-defined waveform, the setting range is 1 to 200.

<NRf+>: MIN|MAX|DEF|<MIN to MAX>. Used to set the slope of the selected step.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1,MIN

Example

ARB:SEQuence:UDEFined:SLEW 2,0.5,@2

Query Syntax

ARB:SEQuence:UDEFined:SLEW? <NR1>[,(@chanlist)]

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

Returns

<NRf+>

ARB:SAVE <NR1>[,(@chanlist)]

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

Group

ARB

Syntax

ARB:SAVE <NR1>[,(@chanlist)]

Arguments

<NR1>, the setting range is from 1 to 10.

(@chanlist): (@1)|(@2)|(@1,2)

Example

ARB:SAVE 2[,(@chanlist)]

Query Syntax

None

Returns

None

ARB:RECall <NR1>[,(@chanlist)]

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>[,(@chanlist)]

Arguments

<NR1>, the setting range is from 1 to 10.

(@chanlist): (@1)|(@2)|(@1,2)

Example

ARB:RECall 2,(@2)

Query Syntax

None

Returns

None

ARB:EXE:STATe? [(@chanlist)]

This command is used to query the status of the ARB: 0 means non-ARB mode, 1 means ARB operation is finished, 2 means ARB is waiting to be triggered, and 3 means ARB is running.

Group

ARB

Syntax

ARB:EXE:STATe? [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Example

ARB:EXE:STATe? (@2)

Returns

0|1|2|3

ARB:SEQuence:CDWell:POINts <NR1>[, (@chanlist)]

This command is used to set the number of points of the CDWell waveform of the selected Nth step of the Sequence. The premise is that **ARB:SEQuence:FUNCTion** has been set to **CDArb**. You must set this number of points firstly, then you can set the other sequence cdarb parameters.

Group

ARB

Syntax

ARB:SEQuence:CDWell:POINts <NR1>[,(@chanlist)]

Arguments

<NR1>, the setting range is from 1 to 65535.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1

Example

ARB:SEQuence:CDWell:POINts 6,(@2)

Query Syntax

ARB:SEQuence:CDWell:POINts? [,(@chanlist)]

Returns

<NR1>

ARB:SEQuence:CDWell:LEVel <NRf+>,<NRf+>,<NR1>[,(@chanlist)]

This command is used to set the high and low amplitude of a step (point) in the CDWell waveform of the selected Nth step of the Sequence, that is, the level of a certain point, and the value is specified in amps or volts. The minimum and maximum values depend on the device rating. The command contains 3 parameters, the first <NRf+> indicates a high amplitude, the second <NRf+> indicates a low amplitude, the third parameter <NR1> indicates the specified step (point), and the setting value of the first parameter must be greater than the second parameter.

Group

ARB

Syntax

ARB:SEQuence:CDWell:LEVel <NRf+>,<NRf+>,<NR1>[,(@chanlist)]

Arguments

<NRf+>: MIN|MAX|DEF|<MIN to MAX>. Used to specify the high/low amplitude of a point.

<NR1>, used to specify which point of the CDWell waveform, range: 1 to 65535.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MAX,MIN,1

Example

ARB:SEQuence:CDWell:LEVel 100,10,2,(@2)

Query Syntax

ARB:SEQuence:CDWell:LEVel? <NR1>[,(@chanlist)]

Query by specifying a point number to return the high and low amplitudes corresponding to the point in the CDWell waveform.

Returns

<NRf+>,<NRf+>

ARB:SEQuence:CDWell:WIDth <NRf+>[,(@chanlist)]

This command is used to set the time width of each point in the CDWell waveform of the selected Nth step of the Sequence. That is, the dwell time of each point, in seconds.

Group

ARB

Syntax

ARB:SEQuence:CDWell:WIDth <NRf+>[,(@chanlist)]

Arguments

<NRf+>[,(@chanlist)]

<NRf+>: MIN|MAX|DEF|<MIN to MAX>

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

MIN

Example

ARB:SEQuence:CDWell:WIDth 5,(@2)

Query Syntax

ARB:SEQuence:CDWell:WIDth? [MIN|MAX|DEF][,(@chanlist)]

Returns

<NRf+>

INITiate[:IMMediate]:ARB [(@chanlist)]

This command is used to initialize the ARB to wait for a trigger. In the process of waveform output, if you want to switch the ARB type, you must execute the **ABORt:ARB** command firstly to terminate the previous ARB waveform.

Group

ARB

Syntax

INITiate[:IMMediate]:ARB [(@chanlist)]

Arguments

(@chanlist): (@1)|(@2)|(@1,2)

Example

INITiate[:IMMediate]:ARB (@2)

Query Syntax

None

Returns

None

ARB:ZOOM:COF <NRf>[,(@chanlist)]

This command is used to set the ARB scaling factor to change the amplitude of the ARB. The default scaling factor is 1.

Group

ARB

Syntax

ARB:ZOOM:COF <NRf>[,(@chanlist)]

Arguments

<NRf>, the setting range is 0.002 to 16384.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

1

Example

ARB:ZOOM:COF 5,(@2)

Query Syntax

ARB:ZOOM:COF? [(@chanlist)]

Returns

<NRf>

ARB:ZOOM:LIMIt:POSitive <NR1>[,(@chanlist)]

This command is used to set the upper limit of ARB scaling.

Group

ARB

Syntax

ARB:ZOOM:LIMIt:POSitive <NR1>[,(@chanlist)]

Arguments

<NR1>, the setting range is -32768 to 32767.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

32767

Example

ARB:ZOOM:LIMI:POS 5,(@2)

Query Syntax

ARB:ZOOM:LIMIt:POSitive? [(@chanlist)]

Returns

<NR1>

ARB:ZOOM:LIMIt:NEGative <NR1>[,(@chanlist)]

This command is used to set the lower limit of ARB scaling.

Group

ARB

Syntax

ARB:ZOOM:LIMIt:NEGative <NR1>[,(@chanlist)]

Arguments

<NR1>, the setting range is -32768 to 32767.

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

32767

Example

ARB:ZOOM:LIMI:NEG 5,(@2)

Query Syntax

ARB:ZOOM:LIMIt:NEGative? [(@chanlist)]

Returns

<NR1>

ARB:PAUSE:STATE <Bool>[,(@chanlist)]

This command is used to control whether the ARB is suspended. 1 means pause and 0 means start (continue).

Group

ARB

Syntax**ARB:PAUSE:STATE <Bool>[,(@chanlist)]****Arguments**

<Bool>[,(@chanlist)]

<Bool>: 0|1

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

0

Example**ARB:PAUSE:STATE 1,(@2)****Query Syntax****ARB:PAUSE:STATE? [(@chanlist)]****Returns**

0|1

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>[,(@chanlist)]**

Save the instrument **[On/Off]** status or several parameter settings to 100 non-volatile memories, and the position can be set from 1 to 100. When shipped, locations 1 through 100 are empty.

Group

None

Syntax

***SAV <NR1>[,(@chanlist)]**

Arguments

<NR1>

The setting range is from 1 to 100.

Default Value

None

Returns

None

Example

***SAV 2,(@1)**

Also see

***RCL <NR1>[,(@chanlist)]**

***RCL <NR1>[,(@chanlist)]**

Recalls a saved instrument state. This restores the instrument to a state that was previously stored in locations 1 through 100 with the ***SAV** command.

Group

None

Syntax

***RCL <NR1>[,(@chanlist)]**

Arguments

<NR1>

The setting range is from 1 to 100.

Default Value

None

Returns

None

Example`*RCL 1,(@2)`**Also see**`*SAV <NR1>[,(@chanlist)]`***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 [19 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 Format Subsystem

17.1 FORMat[:data] <CPD>

This command sets the data format of ELOG.

- ASCii
ASCII format data
- BINary
Binary data

Group

FORMat

Syntax

FORMat[:data] <CPD>

Parameter

<CPD>
ASCii|BINary

Default Value

ASCii

Example

FORMat[:data] <CPD>

Query Syntax

FORMat[:data]?

Returns

<CPD>

17.2 DLOG:EXPort:FORMat[:data] <CPD>[, (@chanlist)]

This command sets the data format of DLOG.

Group

FORMat

Syntax

DLOG:EXPort:FORMat[:data] <CPD>[,(@chanlist)]

Parameter

<CPD>: TDMS|CSV

(@chanlist): (@1)|(@2)|(@1,2)

Default Value

TDMS

Example

DLOG:EXPort:FORMat CSV,(@2)

Query Syntax

DLOG:EXPort:FORMat[:data]? [(@chanlist)]

Returns

<CPD>

18 Example

Demonstrate the order of sending commands for a specific function.

18.1 Dlog

```

sys:rem //Enter remote mode

ABORt:DLOG //To prevent the current Dlog from running,
you can execute the stop command first.

SENSe:DLOG:PERiod 0.001 //Set the sampling interval of data logging

SENSe:DLOG:FUNction:CURRent on //Open current data logging

SENSe:DLOG:FUNction:VOLTagE on //Open voltage data logging

SENSe:DLOG:FUNction:PEAK on //Open peak data recording

SENSe:DLOG:TIME 3 //Set the data logging time to 3 seconds

curr 2 //Set current 2A

volt 3.2 //Set voltage 3.2V

output 1 //Turn on On/Off

INITiate:DLOG //Start data logging

DLOG:STATe? //Query the Dlog status. If the query is 2,
use TRIG:DLOG to trigger the operation.
U disk and TMC cannot be used at the same time.
  
```

18.2 Elog

```

sys:rem //Enter remote mode
  
```

```

abort:elog //To prevent the current Elog from running,
you can execute the stop command first.

format BINary //Select binary data

SENSE:ELOG:PERiod 0.001 //Set the buffer sampling interval to 1ms.

SENSE:ELOG:FUNction:volt 1 //Open voltage data buffer

SENSE:ELOG:FUNction:curr 1 //Open current data buffer

SENSE:ELOG:FUNction:PEAK 0 //Open the peak data cache

curr 2 //Set current 2A

volt 3.2 //Set voltage 3.2V

output 1 //Turn on On/Off

init:elog //Start data cache

elog:stat? //Query elog status. If the query is 2,
use trig:elog to trigger the operation.

FETCh:ELOG:ARRay:DATA? //Query cache data
  
```

18.3 List Function

```

SYSTEM:REMOte //Enter remote mode
TRIGger:ARB:SOURce BUS //Set the trigger source of the list.
FUNCTION:MODE LIST //Enter list mode
ARB:FUNCTION:SHAPE UDEFined //list
ARB:FUNCTION:TYPE VOLTage //Set the operation mode of list
ARB:UDEFined:COUNT 1 //Set the total steps of list
ARB:UDEFined:LEVel 1,10 //Set the first step voltage value
ARB:UDEFined:DWELl 1,5 //Set the first step delay time
ARB:UDEFined:SLEW 1,0.1 //Set the slope of the first step
ARB:COUNT 3 //Set the number of repetitions of the list
ARB:TERMinate:LAST 1 //Set the end state of list
ARB:SAVE 1 //Set save address
ARB:RECall 1 //Recall list
OUTPut 1 //Turn the output on
  
```

```
INITiate:ARB //Initialize the ARB to wait for a trigger
TRIGger:ARB //Trigger run list
```

18.4 Battery mode charge test command

Battery Charging Command:

```
SYST:REM // Enter remote state.
BATT:MODE CHAR // Select charge mode.
BATT:CHAR:VOLT 50 // Set charge voltage to 50V.
BATT:CHAR:CURR 20 // Set charge current to 20A.
BATT:SHUT:VOLT 49 // Set charge cutoff voltage at 49V.
BATT:SHUT:CURR 0.1 // Set charge cutoff current at 0.1A.
BATT:SHUT:CAP 50 // Set cutoff capacity at 50Ah.
BATT:SHUT:TIME 5000 // Set cutoff time at 5000 seconds.
FUNC:MODE BATT // Execute RUN/RESET.
FUNC:MODE FIX // Execute STOP.
OUTP 0 // Turn off output.
```

19 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

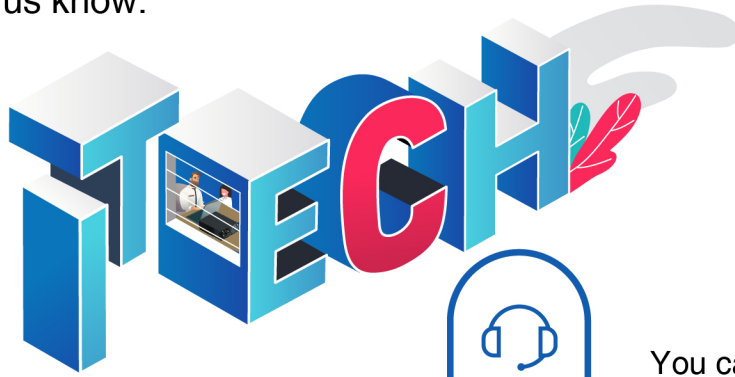
Error Code	Error Message	Description
11	CVDAC selftest 1	In CV mode, DAC channel 1 self-test failed.
12	CVDAC selftest 2	In CV mode, DAC channel 2 self-test failed.
13	CCDAC selftest 1	In CC mode, DAC channel 1 self-test failed.
14	CCDAC selftest 2	In CC mode, DAC channel 2 self-test failed.
20	Input 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.
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

Error Code	Error Message	Description
		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.
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.

Error Code	Error Message	Description
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.



You can chat with us on ITECH website



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



Click and follow ITECH Electronics

