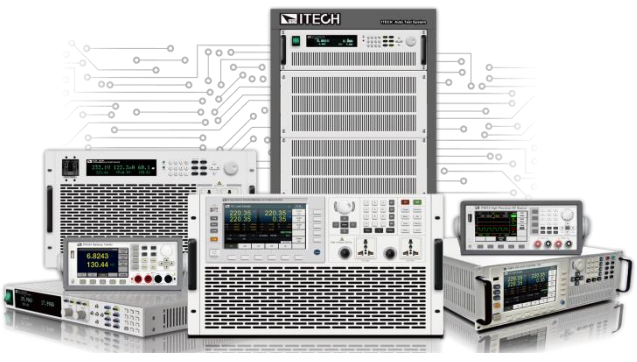
Multi-channel Modular Power System
IT2700 Programming Guide



Model: IT2700
Version: V1.0

Notices

© Itech Electronic, Co., Ltd. 2024
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

IT2700

Revision

1st Edition: March. 8, 2024

Itech Electronic, Co., Ltd.

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 provides this customary commercial license in software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and DFARS 252.227-7015 (Technical Data – Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

Safety Notices

CAUTION

A CAUTION 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 IT2700 series power supply meets all the published specifications at time of shipment from the factory.

Warranty

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

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

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

Limitation of Warranty

This Warranty will be rendered invalid if the product is:

- Damaged resulting from customer-wired circuits or customer-supplied parts or accessories;
- Modified or repaired by customer without authorization;
- Damaged resulting from customer-wired circuits or use in an environment not designated by us;
- 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

Copyright ©ITECH Electronic Co., Ltd.

	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.**
- **Check all marks on the instrument before connecting the instrument to power supply.**
- **Use electric wires of appropriate load. All loading wires should be capable of bearing maximum short-circuit of electronic load without overheating. If there are multiple loads, each pair of the load power cord must be carry out the full rated short-circuit output current of the power securely.**
- **Ensure the voltage fluctuation of mains supply is less than 10% of the working voltage range in order to reduce risks of fire and electric shock.**
- **Do not install alternative parts on the instrument or perform any unauthorized modification.**
- **Do not use the instrument if the detachable cover is removed or loosen.**
- **To prevent the possibility of accidental injuries, be sure to use the power adapter supplied by the manufacturer only.**
- **We do not accept responsibility for any direct or indirect financial damage or loss of profit that might occur when using the instrument.**
- **This instrument is used for industrial purposes, do not apply this product to IT power supply system.**
- **Never use the instrument with a life-support system or any other equipment subject to safety requirements.**

WARNING

- SHOCK HAZARD Ground the Instrument. This product is provided with a protective earth terminal. To minimize shock hazard, the instrument must be connected to the AC mains through a grounded power cable, with the ground wire firmly connected to an electrical ground (safety ground) at the power outlet or distribution box. Any interruption of the protective (grounding)

conductor or disconnection of the protective earth terminal will cause a potential shock hazard that could result in injury or death.

- Before applying power, verify that all safety precautions are taken. All connections must be made with the instrument turned off, and must be performed by qualified personnel who are aware of the hazards involved. Improper actions can cause fatal injury as well as equipment damage.
- SHOCK HAZARD, LETHAL VOLTAGES This product can input the dangerous voltage that can cause personal injury, and the operator must always be protected from electric shock. Ensure that the input electrodes are either insulated or covered using the safety covers provided, so that no accidental contact with lethal voltages can occur.
- Never touch cables or connections immediately after turning off the instrument. Verify that there is no dangerous voltage on the electrodes or sense terminals before touching them.

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

To make accurate measurements, allow the instrument to warm up for 30 min.

Regulatory Markings

CE	The CE mark indicates that the product complies with all the relevant European legal directives. The specific year (if any) affixed refers to the year when the design was approved.
	The instrument complies with the WEEE Directive (2002/96/EC) marking requirement. This affix product label indicates that you must not discard the electrical/electronic product in domestic household waste.

Copyright ©ITECH Electronic Co., Ltd.

	This symbol indicates the time period during which no hazardous or toxic substances are expected to leak or deteriorate during normal use. The expected useful life of the product is 10 years. The product can be used safely during the 10-year Environment Friendly Use Period (EFUP). Upon expiration of the EFUP, the product must be immediately recycled.

Waste Electrical and Electronic Equipment (WEEE) Directive

2002/96/EC Waste Electrical and Electronic Equipment (WEEE) Directive

This product complies with the WEEE Directive (2002/96/EC) marking requirement. This affix product label indicates that you must not discard the electrical/electronic product in domestic household waste.

Product Category

With reference to the equipment classifications described in the Annex 1 of the WEEE Directive, this instrument is classified as a “Monitoring and Control Instrument”.

To return this unwanted instrument, contact your nearest ITECH office.

Copyright ©ITECH Electronic Co., Ltd.

Compliance Information

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

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

Conforms with the following product standards:

EMC Standard

IEC 61326-1:2012/ EN 61326-1:2013 [1][2][3]
Reference Standards
CISPR 11:2009+A1:2010/ EN 55011:2009+A1:2010 (Group 1, Class A)
IEC 61000-4-2:2008/ EN 61000-4-2:2009
IEC 61000-4-3:2006+A1:2007+A2:2010/ EN 61000-4-3:2006+A1:2008+A2:2010
IEC 61000-4-4:2004+A1:2010/ EN 61000-4-4:2004+A1:2010
IEC 61000-4-5:2005/ EN 61000-4-5:2006
IEC 61000-4-6:2008/ EN 61000-4-6:2009
IEC 61000-4-11:2004/ EN 61000-4-11:2004

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/ EN 61010-1:2010

Copyright ©ITECH Electronic Co., Ltd.

CONTENT

Quality Certification and Assurance iii
Warranty iii
Limitation of Warranty iii
Safety Symbols iii
Safety Precautions iv
Environmental Conditions v
Regulatory Markings v
Waste Electrical and Electronic Equipment (WEEE) Directive vi
Compliance Information vii

Chapter1 SCPI Introduction 1

1.1 Overview 1
1.2 Command Type of SCPI 1
1.3 Message Type of SCPI 3
1.4 Response Data Type 4
1.5 Command Format 5
1.6 Data Type 7
1.7 Remote Interface Connection 8

Chapter2 Example of Common Commands 1

Example 1:Identify the power supply 1
Example 2:Common output commands 1

Chapter3 SCPI Status Register 3

Chapter4 STATus Commands 6

STATus:QUEStionable[:EVENt]? (<@chanlist>) 6
STATus:QUEStionable:CONDition? (<@chanlist>) 6
STATus:QUEStionable:ENABle <value>,(<@chanlist>) 6
STATus:QUEStionable:NTRansition <value>, (<@chanlist>) 7
STATus:QUEStionable:PTRansition <value>, (<@chanlist>) 8
STATus:OPERation[:EVENt]? (<@chanlist>) 8
STATus:OPERation:CONDition? (<@chanlist>) 8
STATus:OPERation:ENABle <value>,(<@chanlist>) 9
STATus:OPERation:NTRansition <value>,(<@chanlist>) 9
STATus:OPERation:PTRansition <value>,(<@chanlist>) 10
STATus:PRESet 10

Chapter5 DC power Commands 12

SYSTem:VERSion? *12*
SYSTem:ERRor[:NEXT]? *12*
SYSTem:CHANnel[:COUNt]? *13*
SYSTem:CHANnel:MODel? (@<chanlist>) *13*
SYSTem:CHANnel:OPTion? (@<chanlist>) *13*
SYSTem:CHANnel:DESCription? (@<chanlist>) *13*
SYSTem:CHANnel:SERial? (@<chanlist>) *13*
SYSTem:CHANnel:STATus? (@<chanlist>) *14*
SYSTem:CHANnel:STATus:SUMMary? *14*
SYSTem:CHANnel:VERSion? (@<chanlist>) *14*
SYSTem:CHANnel:CATegory? (@<chanlist>) *14*
SYSTem:COMMunicate:RLSTate LOCal|REMote|RWLock *15*
SYSTem:REMote *15*
SYSTem:LOCal *15*
SYSTem:RWLock *15*
SYSTem:KEY <0-255> *16*
SYSTem:PRESet *16*
SYSTem:POSetup <CPD> *16*
SYSTem:CLEar *17*

Copyright ©ITECH Electronic Co., Ltd.

SYSTem:BEEPer:IMMediate 17
SYSTem:BEEPer:CLICk <0|1|OFF|ON> 17
SYSTem:BEEPer:ALARm <0|1|OFF|ON> 18
SYSTem:COMMunicate:USB:TYPE <DEVice|HOST> 18
SYSTem:COMMunicate:SELect <TMC|VCP> 19
SYSTem:COMMunicate:GPIB:ADDRess <NR1> 19
SYSTem:COMMunicate:SERial:BAUDrate <CPD> 19
SYSTem:COMMunicate:LAN:IP[:CONFiguration] <ip> 20
SYSTem:COMMunicate:LAN:IP[:CONFiguration]:MODE <CPD> 20
SYSTem:COMMunicate:LAN:SMASk <SPD> 21
SYSTem:COMMunicate:LAN:DGATeway <SPD> 21
SYSTem:COMMunicate:LAN:RAWSocketport <port> 22
SYSTem:OPERation:CATalog? 22
SYSTem:OPERation:MODE <CPD> 22
SYSTem:BRIGhtness:LEVel <NR1> 23
SYSTem:TOUCh[:STATe] <CPD> 23
SYSTem:LANGuage <CPD> 24
SYSTem:SOFT:KEYBoard[:STATe] <CPD> 24
SYSTem:KNOB:IMMediate:EFFective <CPD> 24
SYSTem:RUN:TIME? 25
SYSTem:GROup:DEFine (@<chanlist>) 25
SYSTem:GROup:DELete (@<chanlist>) 25
SYSTem:GROup:DELete:ALL 26
SYSTem:GROup:APPLy 26
SYSTem:REBoot 26
SYSTem:SECurity:IMMediate 26
SYSTem:SOURce:SLEW:MODE <RATE|TIME> 27
SYSTem:LOAD:CURRent:SYMBol <POSitive|NEGative> 27
5.2 ABORt Subsystem 27
ABORt:ACQuire (@<chanlist>) 27
ABORt[:ALL] 28
ABORt:DLOG 28
ABORt:ELOG (@<chanlist>) 28
ABORt:TRANsient (@<chanlist>) 28
5.3 INITiate Subsystem 29
INITiate[:IMMediate]:ACQuire (@<chanlist>) 29
INITiate[:IMMediate]:DLOG <"filename"> 29
INITiate[:IMMediate]:ELOG (@<chanlist>) 29
INITiate[:IMMediate]:TRANsient (@<chanlist>) 30
INITiate:CONTinuous[:ACQuire] 0|OFF|1|ON,(@<chanlist>) 30
INITiate:CONTinuous:TRANsient 0|OFF|1|ON,(@<chanlist>) 30
5.4 SOURce Subsystem 31
[SOURce:]FUNCtion CURRent|VOLTage, (@<chanlist>) 31
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>) 31
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>) 32
[SOURce:]VOLTage:LIMit:HIGH[:IMMediate][:AMPLitude] <value>,(@<chanlist>) 32
[SOURce:]VOLTage:LIMit:LOW[:IMMediate][:AMPLitude] <value>,(@<chanlist>) 33
[SOURce:]VOLTage:LIMit:COUPle 0|OFF|1|ON,(@<chanlist>) 33
[SOURce:]VOLTage:PROTection[:LOCal][:POSitive][:LEVel] <value>,(@<chanlist>) 34
[SOURce:]VOLTage:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>) 34
[SOURce:]VOLTage:PROTection:DELay[:TIME] <value>, (@<chanlist>) 35
[SOURce:]VOLTage:UNDer:PROTection[:POSitive][:LEVel] <value>,(@<chanlist>) 35
[SOURce:]VOLTage:UNDer:PROTection:STATe <0|OFF|1|ON>,(@<chanlist>) 36
[SOURce:]VOLTage:UNDer:PROTection:DELay[:TIME] <value>,(@<chanlist>) 36
[SOURce:]VOLTage:UNDer:PROTection:WARM[:TIME] <value>,(@<chanlist>) 37
[SOURce:]VOLTage:RESistance[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>) 37
[SOURce:]VOLTage:RESistance:STATe <0|OFF|1|ON>, (@<chanlist>) 38
[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>) 38
[SOURce:]VOLTage:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>) 39

Copyright ©ITECH Electronic Co., Ltd.

[SOURce:]VOLTage:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>) 39
[SOURce:]VOLTage:SLEW[:POSitive]:MAXimum 0|OFF|1|ON,(@<chanlist>) 39
[SOURce:]VOLTage:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>) 40
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>) 40
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>) 41
[SOURce:]CURRent:LIMit[:POSitive][:IMMediate][:AMPLitude] <value>, (@<chanlist>) 41
[SOURce:]CURRent:LIMit:NEGative[:IMMediate][:AMPLitude] <value>, (@<chanlist>) 42
[SOURce:]CURRent:LIMit:COUPle 0|OFF|1|ON, (@<chanlist>) 42
[SOURce:]CURRent:PROTection[:LEVel][:POSitive] <level>,(@<chanlist>) 43
[SOURce:]CURRent:PROTection[:LEVel]:NEGative <level>,(@<chanlist>) 43
[SOURce:]CURRent:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>) 44
[SOURce:]CURRent:PROTection:DELay[:TIME] <value>, (@<chanlist>) 44
[SOURce:]CURRent:UNDer:PROTection[:LEVel][:POSitive] <level>,(@<chanlist>) 44
[SOURce:]CURRent:UNDer:PROTection[:LEVel]:NEGative <level>,(@<chanlist>) 45
[SOURce:]CURRent:UNDer:PROTection:STATe <0|OFF|1|ON>,(@<chanlist>) 45
[SOURce:]CURRent:UNDer:PROTection:DELay[:TIME] <value>,(@<chanlist>) 45
[SOURce:]CURRent:UNDer:PROTection:WARM[:TIME] <value>,(@<chanlist>) 46
[SOURce:]CURRent:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>) 46
[SOURce:]CURRent:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>) 46
[SOURce:]CURRent:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>) 47
[SOURce:]CURRent:SLEW[:POSitive]:MAXimum 0|OFF|1|ON,(@<chanlist>) 47
[SOURce:]CURRent:SLEW:NEGative:MAXimum 0|OFF|1|ON,(@<chanlist>) 47
[SOURce:]POWer:LIMit[:POSitive][:IMMediate][:AMPLitude] <value>, (<@chanlist>) 48
[SOURce:]POWer:LIMit:NEGative[:IMMediate][:AMPLitude]<value>, (<@chanlist>) 48
[SOURce:]POWer:LIMit:COUPle 0|OFF|1|ON, (@<chanlist>) 49
[SOURce:]POWer:PROTection:DELay[:TIME] <value>, (@<chanlist>) 49
[SOURce:]POWer:PROTection:STATe 0|OFF|1|ON, (@<chanlist>) 50
[SOURce:]POWer:PROTection[:LEVel][:POSitive] <value>,(@<chanlist>) 50
[SOURce:]POWer:PROTection[:LEVel]:NEGative <value>,(@<chanlist>) 50
[SOURce:]POWer:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>) 51
[SOURce:]POWer:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>) 51
[SOURce:]POWer:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>) 51
[SOURce:]POWer:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>) 52
[SOURce:]POWer:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>) 52
[SOURce:]SINK:RESistance[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>) 53
[SOURce:]SINK:RESistance:STATe 0|OFF|1|ON,(@<chanlist>) 53
5.5 OUTPut Subsystem 54
INPut|OUTPut[:STATe] 0|OFF|1|ON , (@<chanlist>) 54
INPut|OUTPut[:STATe]:COUPle:CHANnel ALL|NONE|[<channel>{,<channel>}] 54
INPut|OUTPut[:STATe]:COUPle:DOFFset <value> 54
INPut|OUTPut[:STATe]:COUPle:DOFFset:MODE AUTO|MANual 55
INPut|OUTPut[:STATe]:COUPle:MAX:DOFFset? 55
INPut|OUTPut[:STATe]:DELay:FALL <value>, (@<chanlist>) 55
INPut|OUTPut[:STATe]:DELay:RISE <value>, (@<chanlist>) 56
INPut|OUTPut:INHibit:COUPle:CHANnel ALL|NONE|[<channel> {,<channel>}] 56
INPut|OUTPut:INHibit:MODE LATChing|LIVE|OFF 57
INPut|OUTPut:PON:STATe RST|LAST|LOFF 57
INPut|OUTPut:PROTection:CLEar (@<chanlist>) 58
INPut|OUTPut:PROTection:COUPle 0|OFF|1|ON 58
INPut|OUTPut:PROTection:COUPle:CHANnel ALL|NONE|[<channel>{,<channel>}] 58
INPut|OUTPut:PROTection:WDOG[:STATe] 0|OFF|1|ON 59
INPut|OUTPut:PROTection:WDOG:DELay <value> 59
INPut|OUTPut:PROTection:FOLDback[:MODE] <OFF|CC|CV>,(@<chanlist>) 59
INPut|OUTPut:PROTection:FOLDback:DELay <value>,(@<chanlist>) 60
INPut|OUTPut:RELay:LOCK[:STATe] 0|1|OFF|ON,(@<chanlist>) 60
INPut|OUTPut:REGulation:SPEed <LOW|HIGH1|HIGH2|HIGH3> (@<chanlist>) 61
INPut|OUTPut:FUNCtion:MODE
<FIXed|ARB|BCHarge|BEMulation|TRANsient|OCP|OPP|SEQuence|CDARB|SSWeep|LIST|BDIScharge>,(@<chanlist>) 61

Copyright ©ITECH Electronic Co., Ltd.

INPut|OUTPut:OFF:VOLTage:RTZero <0||OFF|1|ON>,(@<chanlist>) 62
5.6 FETCh & MEASureSubsystem 62
FETCh[:SCALar]:CURRent[:DC]? (@<chanlist>) 62
MEASure[:SCALar]:CURRent[:DC]? (@<chanlist>) 62
FETCh[:SCALar]:CURRent:ACDC? (@<chanlist>) 63
MEASure[:SCALar]:CURRent:ACDC? (@<chanlist>) 63
FETCh[:SCALar]:CURRent:MAXimum? (@<chanlist>) 63
MEASure[:SCALar]:CURRent:MAXimum? (@<chanlist>) 63
FETCh[:SCALar]:CURRent:MINimum? (@<chanlist>) 64
MEASure[:SCALar]:CURRent:MINimum? (@<chanlist>) 64
FETCh[:SCALar]:POWer[:DC]? (@<chanlist>) 64
MEASure[:SCALar]:POWer[:DC]? (@<chanlist>) 64
FETCh[:SCALar]:POWer:MAXimum? (@<chanlist>) 65
MEASure[:SCALar]:POWer:MAXimum? (@<chanlist>) 65
FETCh[:SCALar]:POWer:MINimum? (@<chanlist>) 65
MEASure[:SCALar]:POWer:MINimum? (@<chanlist>) 65
FETCh[:SCALar]:VOLTage[:DC]? (@<chanlist>) 66
MEASure[:SCALar]:VOLTage[:DC]? (@<chanlist>) 66
FETCh[:SCALar]:VOLTage:ACDC? (@<chanlist>) 66
MEASure[:SCALar]:VOLTage:ACDC? (@<chanlist>) 66
FETCh[:SCALar]:VOLTage:MINimum? (@<chanlist>) 67
MEASure[:SCALar]:VOLTage:MINimum? (@<chanlist>) 67
FETCh[:SCALar]:VOLTage:MAXimum? (@<chanlist>) 67
MEASure[:SCALar]:VOLTage:MAXimum? (@<chanlist>) 67
FETCh[:SCALar]:AHOur? (@<chanlist>) 68
MEASure[:SCALar]:AHOur? (@<chanlist>) 68
FETCh[:SCALar]:WHOur? (@<chanlist>) 68
MEASure[:SCALar]:WHOur? (@<chanlist>) 68
FETCh[:SCALar]:TIME? (@<chanlist>) 69
MEASure[:SCALar]:TIME? (@<chanlist>) 69
FTECh[:SCALar][:ALL]? (@<chanlist>) 69
MEASure[:SCALar][:ALL]? (@<chanlist>) 69
FETCh:ARRay:CURRent[:DC]? (@<chanlist>) 70
MEASure:ARRay:CURRent[:DC]? (@<chanlist>) 70
FETCh:ARRay:VOLTage[:DC]? (@<chanlist>) 70
MEASure:ARRay:VOLTage[:DC]? (@<chanlist>) 70
FETCh:ARRay:POWer[:DC]? (@<chanlist>) 71
MEASure:ARRay:POWer[:DC]? (@<chanlist>) 71
SENSe[:REMote][:STATe] 0|1|OFF|ON, (@<channlist>) 71
SENSe:SWEep:OFFSet:POINts <value>, (@<chanlist>) 72
SENSe:SWEep:POINts <value>, (@<chanlist>) 72
SENSe:SWEep:TINTerval <value>, (@<chanlist>) 73
SENSe:NPLCycles? [MINimum|MAXimum,](@<chanlist>) 73
SENSe:APERture <time|MINimum|MAXimum>,(@<chanlist>) 74
SENSe:MEASure:TIME:MODE <NPLC|APERture>,(@<chanlist>) 74
SENSe:AHOur:CLEar (@<chanlist>) 75
SENSe:WHOur:CLEar (@<chanlist>) 75
SENSe:AHOur:CLEar:AUTO <0|1|OFF|ON>,(@<chanlist>) 75
SENSe:WHOur:CLEar:AUTO <0|1|OFF|ON>,(@<chanlist>) 76
SENSe:DLOG:FUNCtion:CURRent 0|OFF|1|ON, (@<chanlist>) 76
SENSe:DLOG:FUNCtion:VOLTage 0|OFF|1|ON, (@<chanlist>) 77
SENSe:DLOG:FUNCtion:MINMax 0|OFF|1|ON 77
SENSe:DLOG:MARKer1:POINt <value> 77
SENSe:DLOG:MARKer2:POINt <value> 78
SENSe:DLOG:OFFSet <value> 78
SENSe:DLOG:PERiod <value> 78
SENSe:DLOG:TIME <value> 79
SENSe:DLOG:TINTerval <value> 79
SENSe:ELOG:FUNCtion:CURRent 0|OFF|1|ON, (@<chanlist>) 80

Copyright ©ITECH Electronic Co., Ltd.

SENSe:ELOG:FUNCtion:VOLTage 0|OFF|1|ON, (@<chanlist>) 80
SENSe:ELOG:FUNCtion:CURRent:MINMax 0|OFF|1|ON, (@<chanlist>) 80
SENSe:ELOG:FUNCtion:VOLTage:MINMax 0|OFF|1|ON, (@<chanlist>) 81
SENSe:ELOG:PERiod <value>, (@<chanlist>) 81
[SOURce:]LIST:COUNt <value>, (@<chanlist>) 82
[SOURce:]LIST:CURRent:STEP[:LEVel] <value>,{<step>,(@<chanlist>) 82
[SOURce:]LIST:CURRent:SLEW <value>,<step>, (@<chanlist>) 83
[SOURce:]LIST:CURRent:POINts <value>,(@<chanlist>) 83
[SOURce:]LIST:VOLTage:STEP[:LEVel] <value>,<step>, (@<chanlist>) 83
[SOURce:]LIST:VOLTage:SLEW <value>,<step>, (@<chanlist>) 84
[SOURce:]LIST:VOLTage:POINts <value>,(@<chanlist>) 84
[SOURce:]LIST:POWer:STEP[:LEVel] <value>,<step>, (@<chanlist>) 85
[SOURce:]LIST:POWer:SLEW <value>,<step>, (@<chanlist>) 85
[SOURce:]LIST:POWer:POINts <value>,(@<chanlist>) 85
[SOURce:]LIST:RESistance:STEP[:LEVel] <value>,<step>, (@<chanlist>) 86
[SOURce:]LIST:RESistance:SLEW <value>,<step>, (@<chanlist>) 86
[SOURce:]LIST:RESistance:POINts <value>,(@<chanlist>) 87
[SOURce:]LIST:TERMinate:MODE <NORMal|LAST|OFF>,(@<chanlist>) 87
[SOURce:]LIST:DWELl <value>,<step>, (@<chanlist>) 87
[SOURce:]LIST:TOUTput:BOSTep:POINts? (@<chanlist>) 88
[SOURce:]LIST:TOUTput:EOSTep:POINts? (@<chanlist>) 88
[SOURce:]LIST:STEP ONCE|AUTO,<step>(@<chanlist>) 88
[SOURce:]LIST:TOUTput:BOSTep[:DATA] 0|OFF|1|ON,<step>,(@<chanlist>) 89
[SOURce:]LIST:TOUTput:EOSTep[:DATA] 0|OFF|1|ON,<step>,(@<chanlist>) 89
[SOURce:]LIST:COUNt:RUN? (@<chanlist>) 89
[SOURce:]LIST:STEP:RUN? (@<chanlist>) 90
[SOURce]LIST:FILE:OPEN <string>,LOCal|USB,(@<chanlist>) 90
[SOURce]LIST:FILE:SAVE <string>,LOCal|USB,(@<chanlist>) 90
[SOURce:]LIST:FUNCtion:TYPE <type>,(@<chanlist>) 90
TRIGger:ACQuire[:IMMediate] (@<chanlist>) 91
TRIGger:ACQuire:SOURce <source>, (@<chanlist>) 91
TRIGger:ACQuire:VOLTage[:LEVel] <value>, (@<chanlist>) 91
TRIGger:ACQuire:VOLTage:SLOPe POSitive|NEGative, (@<chanlist>) 92
TRIGger:ACQuire:CURRent[:LEVel] <value>, (@<chanlist>) 92
TRIGger:ACQuire:CURRent:SLOPe POSitive|NEGative, (@<chanlist>) 92
TRIGger:ELOG[:IMMediate] (@<chanlist>) 93
TRIGger:ELOG:SOURce <source>, (@<chanlist>) 93
TRIGger:DLOG[:IMMediate] (@<chanlist>) 93
TRIGger:DLOG:SOURce <source> 94
TRIGger:DLOG:VOLTage[:LEVel] <value>, (@<chanlist>) 94
TRIGger:DLOG:VOLTage:SLOPe POSitive|NEGative, (@<chanlist>) 95
TRIGger:DLOG:CURRent[:LEVel] <value>, (@<chanlist>) 95
TRIGger:DLOG:CURRent:SLOPe POSitive|NEGative, (@<chanlist>) 95
TRIGger:TRANsient[:IMMediate] (@<chanlist>) 96
TRIGger:MAP:DC[:STATe] 0|OFF|1|ON, (@chanlist) 96
TRIGger:MAP:DC:PIN PIN<1-7>, (@chanlist) 96
TRIGger:MAP:ASToutput[:STATe] 0|1|OFF|ON, (@chanlist) 97
TRIGger:MAP:ASToutput:PIN PIN<1-7>, (@chanlist) 97
[SOURce:]DIGital:INPut:DATA? 98
[SOURce:]DIGital:OUTPut:DATA <value> 98
[SOURce:]DIGital:PIN<1-7>:FUNCtion <function> 98
[SOURce:]DIGital:PIN<1-7>:POLarity POSitive|NEGative 99
[SOURce:]DIGital:ONOFf:STATe:CHANnel <1-8> 99
[SOURce:]DIGital:PLUSe:WIDTh <value> 100
BATTery:CHARge:MODE <CURRent|VOLTage>,(@<chanlist>) 100
BATTery:CHARge:VOLTage <value>,(@<chanlist>) 101
BATTery:CHARge:CURRent <value>,(@<chanlist>) 101
BATTery:CHARge:STOP:VOLTage <value>,(@<chanlist>) 101
BATTery:CHARge:STOP:VOLTage:STATe <0|1|OFF|ON>,(@<chanlist>) 102

Copyright ©ITECH Electronic Co., Ltd.

BATTery:CHARge:STOP:CURRent <value>,(@<chanlist>) 102
BATTery:CHARge:STOP:CURRent:STATe <0|1|OFF|ON>,(@<chanlist>) 103
BATTery:CHARge:STOP:AHOur <value>,(@<chanlist>) 103
BATTery:CHARge:STOP:AHOur:STATe <0|1|OFF|ON>,(@<chanlist>) 103
BATTery:CHARge:STOP:WHOur <value>,(@<chanlist>) 104
BATTery:CHARge:STOP:WHOur:STATe <0|1|OFF|ON>,(@<chanlist>) 104
BATTery:CHARge:STOP:TIME <value>,(@<chanlist>) 105
BATTery:CHARge:STOP:TIME:STATe <0|1|OFF|ON>,(@<chanlist>) 105
BATTERY:CHARge:SAVe <"filename">,(@<chanlist>) 105
BATTery:CHARge:RECall <"filename">,(@<chanlist>) 106
BATTery:CHARge:DELete <"filename">,(@<chanlist>) 106
BATTery:CHARge:TIME? (@<chanlist>) 106
BATTery:CHARge:RSTate? (@<chanlist>) 107
BATTery:DISCharge:MODE <CURRent|VOLTage>,(@<chanlist>) 107
BATTery:DISCharge:VOLTage <value>,(@<chanlist>) 107
BATTery:DISCharge:CURRent <value>,(@<chanlist>) 108
BATTery:DISCharge:POWer <value>,(@<chanlist>) 108
BATTery:DISCharge:RESistance <value>,(@<chanlist>) 108
BATTery:DISCharge:STOP:VOLTage <value>,(@<chanlist>) 109
BATTery:DISCharge:STOP:VOLTage:STATe <0|1|OFF|ON>,(@<chanlist>) 109
BATTery:DISCharge:STOP:CURRent <value>,(@<chanlist>) 110
BATTery:DISCharge:STOP:CURRent:STATe <0|1|OFF|ON>,(@<chanlist>) 110
BATTery:DISCharge:STOP:AHOur <value>,(@<chanlist>) 110
BATTery:DISCharge:STOP:AHOur:STATe <0|1|OFF|ON>,(@<chanlist>) 111
BATTery:DISCharge:STOP:WHOur <value>,(@<chanlist>) 111
BATTery:DISCharge:STOP:WHOur:STATe <0|1|OFF|ON>,(@<chanlist>) 112
BATTery:DISCharge:STOP:TIME <value>,(@<chanlist>) 112
BATTery:DISCharge:STOP:TIME:STATe <0|1|OFF|ON>,(@<chanlist>) 112
BATTERY:DISCharge:SAVe <"filename">,(@<chanlist>) 113
BATTery:DISCharge:RECall <"filename">,(@<chanlist>) 113
BATTery:DISCharge:DELete <"filename">,(@<chanlist>) 113
BATTery:DISCharge:TIME? (@<chanlist>) 114
BATTery:DISCharge:RSTate? (@<chanlist>) 114

Chapter6 DC Load Command 115

SYSTem:VERSion? 115
SYSTem:ERRor[:NEXT]? 115
SYSTem:CHANnel[:COUNt]? 116
SYSTem:CHANnel:MODel? (@<chanlist>) 116
SYSTem:CHANnel:OPTion? (@<chanlist>) 116
SYSTem:CHANnel:DESCription? (@<chanlist>) 116
SYSTem:CHANnel:SERial? (@<chanlist>) 116
SYSTem:CHANnel:STATus? (@<chanlist>) 117
SYSTem:CHANnel:STATus:SUMMary? 117
SYSTem:CHANnel:VERSion? (@<chanlist>) 117
SYSTem:CHANnel:CATegory? (@<chanlist>) 117
SYSTem:CHANnel:TYPE? (@<chanlist>) 118
SYSTem:COMMunicate:RLSTate LOCal|REMote|RWLock 118
SYSTem:REMote 118
SYSTem:LOCal 118
SYSTem:RWLock 119
SYSTem:KEY <0-255> 119
SYSTem:PRESet 119
SYSTem:POSetup <CPD> 119
SYSTem:CLEar 120
SYSTem:BEEPer:IMMediate 120
SYSTem:BEEPer:CLICk <0|1|OFF|ON> 120
SYSTem:BEEPer:ALARm <0|1|OFF|ON> 121
SYSTem:COMMunicate:USB:TYPE <CPD> 121

Copyright ©ITECH Electronic Co., Ltd.

SYSTem:COMMunicate:SELect <TMC|VCP> 122
SYSTem:COMMunicate:LAN:IP[:CONFiguration] <ip> 122
SYSTem:COMMunicate:LAN:IP[:CONFiguration]:MODE <CPD> 123
SYSTem:COMMunicate:LAN:SMASk <SPD> 123
SYSTem:COMMunicate:LAN:DGATeway <SPD> 124
SYSTem:COMMunicate:LAN:RAWSocketport <port> 124
SYSTem:BRIGhtness:LEVel <NR1> 125
SYSTem:TOUCh[:STATe] <CPD> 125
SYSTem:LANGuage <CPD> 126
SYSTem:SOFT:KEYBoard[:STATe] <CPD> 126
SYSTem:KNOB:IMMediate:EFFective <CPD> 126
SYSTem:RUN:TIME? 127
SYSTem:GROup:DEFine (@<chanlist>) 127
SYSTem:GROup:DELete (@<chanlist>) 127
SYSTem:GROup:DELete:ALL 128
SYSTem:GROup:APPLy 128
SYSTem:REBoot 128
SYSTem:SECurity:IMMediate 128
SYSTem:LOAD:CURRent:SYMBol <POSitive|NEGative> 128
6.2 ABORt Subsystem 129
ABORt:ACQuire (@<chanlist>) 129
ABORt[:ALL] 129
ABORt:DLOG 129
ABORt:ELOG (@<chanlist>) 130
ABORt:TRANsient (@<chanlist>) 130
6.3 INITiate Subsystem 130
INITiate[:IMMediate]:ACQuire (@<chanlist>) 130
INITiate[:IMMediate]:DLOG <"filename"> 131
INITiate[:IMMediate]:ELOG (@<chanlist>) 131
INITiate[:IMMediate]:TRANsient (@<chanlist>) 131
INITiate:CONTinuous[:ACQuire] 0|OFF|1|ON,(@<chanlist>) 131
INITiate:CONTinuous:TRANsient 0|OFF|1|ON,(@<chanlist>) 132
6.4 SOURce Subsystem 132
[SOURce:]FUNCtion CURRent|VOLTage, (@<chanlist>) 132
[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>) 133
[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>) 133
[SOURce:]VOLTage:PROTection[:LOCal][:POSitive][:LEVel] <value>, (@<chanlist>) 134
[SOURce:]VOLTage:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>) 134
[SOURce:]VOLTage:PROTection:DELay[:TIME] <value>, (@<chanlist>) 135
[SOURce:]VOLTage:UNDer:PROTection[:POSitive][:LEVel] <value>, (@<chanlist>) 135
[SOURce:]VOLTage:UNDer:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>) 136
[SOURce:]VOLTage:UNDer:PROTection:DELay[:TIME] <value>, (@<chanlist>) 136
[SOURce:]VOLTage:UNDer:PROTection:WARM[:TIME] <value>, (@<chanlist>) 137
[SOURce:]VOLTage:RESistance[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>) 137
[SOURce:]VOLTage:RESistance:STATe <0|OFF|1|ON>, (@<chanlist>) 138
[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>) 138
[SOURce:]VOLTage:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>) 139
[SOURce:]VOLTage:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>) 139
[SOURce:]VOLTage:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>) 139
[SOURce:]VOLTage:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>) 140
[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>) 140
[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>) 141
[SOURce:]CURRent:LIMit[:POSitive][:IMMediate][:AMPLitude] <value>, (@<chanlist>) 141
[SOURce:]CURRent:LIMit:NEGative[:IMMediate][:AMPLitude] <value>, (@<chanlist>) 142
[SOURce:]CURRent:LIMit:COUPle 0|OFF|1|ON, (@<chanlist>) 142
[SOURce:]CURRent:PROTection[:LEVel][:POSitive] <level>,(@<chanlist>) 143
[SOURce:]CURRent:PROTection[:LEVel]:NEGative <level>,(@<chanlist>) 143
[SOURce:]CURRent:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>) 144
[SOURce:]CURRent:PROTection:DELay[:TIME] <value>, (@<chanlist>) 144

Copyright ©ITECH Electronic Co., Ltd.

[SOURce:]CURRent:UNDer:PROTection[:LEVel][:POSitive] <level>,(@<chanlist>) 144
[SOURce:]CURRent:UNDer:PROTection[:LEVel]:NEGative <level>,(@<chanlist>) 145
[SOURce:]CURRent:UNDer:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>) 145
[SOURce:]CURRent:UNDer:PROTection:DELay[:TIME] <value>, (@<chanlist>) 145
[SOURce:]CURRent:UNDer:PROTection:WARM[:TIME] <value>, (@<chanlist>) 146
[SOURce:]CURRent:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>) 146
[SOURce:]CURRent:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>) 146
[SOURce:]CURRent:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>) 147
[SOURce:]CURRent:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>) 147
[SOURce:]CURRent:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>) 147
[SOURce:]POWer:LIMit[:POSitive][:IMMediate][:AMPLitude] <value>, (<@chanlist>) 148
[SOURce:]POWer:LIMit:NEGative[:IMMediate][:AMPLitude]<value>, (<@chanlist>) 148
[SOURce:]POWer:LIMit:COUPle 0|OFF|1|ON, (@<chanlist>) 149
[SOURce:]POWer:PROTection:DELay[:TIME] <value>, (@<chanlist>) 149
[SOURce:]POWer:PROTection:STATe 0|OFF|1|ON, (@<chanlist>) 150
[SOURce:]POWer:PROTection[:LEVel][:POSitive] <value>,(@<chanlist>) 150
[SOURce:]POWer:PROTection[:LEVel]:NEGative <value>,(@<chanlist>) 150
[SOURce:]POWer:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>) 151
[SOURce:]POWer:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>) 151
[SOURce:]POWer:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>) 151
[SOURce:]POWer:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>) 152
[SOURce:]POWer:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>) 152
[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>) 153
[SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>) 153
[SOURce:]RESistance:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>) 154
[SOURce:]RESistance:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>) 154
[SOURce:]RESistance:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>) 154
[SOURce:]RESistance:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>) 155
[SOURce:]RESistance:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>) 155
6.5 OUTPut Subsystem 156
INPut|OUTPut[:STATe] 0|OFF|1|ON , (@<chanlist>) 156
INPut|OUTPut[:STATe]:COUPle:CHANnel ALL|NONE|[<channel> {,<channel>}] 156
INPut|OUTPut[:STATe]:COUPle:DOFFset <value> 156
INPut|OUTPut[:STATe]:COUPle:DOFFset:MODE AUTO|MANual 157
INPut|OUTPut[:STATe]:COUPle:MAX:DOFFset? 157
INPut|OUTPut[:STATe]:DELay:FALL <value>, (@<chanlist>) 157
INPut|OUTPut[:STATe]:DELay:RISE <value>, (@<chanlist>) 158
INPut|OUTPut:INHibit:COUPle:CHANnel ALL|NONE|[<channel> {,<channel>}] 158
INPut|OUTPut:INHibit:MODE LATChing|LIVE|OFF 159
INPut|OUTPut:PON:STATe RST|LAST|LOFF 159
INPut|OUTPut:PROTection:CLEar (@<chanlist>) 160
INPut|OUTPut:PROTection:COUPle 0|OFF|1|ON 160
INPut|OUTPut:PROTection:COUPle:CHANnel ALL|NONE|[<channel>{,<channel>}] 160
INPut|OUTPut:PROTection:WDOG[:STATe] 0|OFF|1|ON 161
INPut|OUTPut:PROTection:WDOG:DELay <value> 161
INPut|OUTPut:PROTection:FOLDback[:MODE] <OFF|CC|CV>,(@<chanlist>) 161
INPut|OUTPut:PROTection:FOLDback:DELay <value>,(@<chanlist>) 162
INPut|OUTPut:RELay:LOCK[:STATe] 0|1|OFF|ON,(@<chanlist>) 162
INPut|OUTPut:REGulation:SPEed <LOW|HIGH1|HIGH2|HIGH3> (@<chanlist>) 163
INPut|OUTPut:FUNCtion:MODE <FIXed|ARB|BCHarge|BEMulation|TRANsient|OCP|OPP|SEQuence|CDARB|SSWeep|LIST|BDISCharge>,(@<chanlist>) 163
INPut|OUTPut:OFF:VOLTage:RTZero <0||OFF|1|ON>,(@<chanlist>) 164
INPut|OUTPut:SHORt[:STATe] 0|OFF|1|ON, (@<chanlist>) 164
INPut|OUTPut:REGulation:SPEed <LOW|HIGH1|HIGH2|HIGH3> (@<chanlist>) 165
INPut|OUTPut:FUNCtion:MODE <FIXed|ARB|BCHarge|BEMulation|TRANsient|OCP|OPP|SEQuence|CDARB|SSWeep|LIST|BDISCharge>,(@<chanlist>) 165
6.6 FETCh & MEASureSubsystem 166

FETCh[:SCALar]:CURRent[:DC]? (@<chanlist>) 166
MEASure[:SCALar]:CURRent[:DC]? (@<chanlist>) 166
FETCh[:SCALar]:CURRent:ACDC? (@<chanlist>) 166
MEASure[:SCALar]:CURRent:ACDC? (@<chanlist>) 166
FETCh[:SCALar]:CURRent:MAXimum? (@<chanlist>) 167
MEASure[:SCALar]:CURRent:MAXimum? (@<chanlist>) 167
FETCh[:SCALar]:CURRent:MINimum? (@<chanlist>) 167
MEASure[:SCALar]:CURRent:MINimum? (@<chanlist>) 167
FETCh[:SCALar]:POWer[:DC]? (@<chanlist>) 168
MEASure[:SCALar]:POWer[:DC]? (@<chanlist>) 168
FETCh[:SCALar]:POWer:MAXimum? (@<chanlist>) 168
MEASure[:SCALar]:POWer:MAXimum? (@<chanlist>) 168
FETCh[:SCALar]:POWer:MINimum? (@<chanlist>) 169
MEASure[:SCALar]:POWer:MINimum? (@<chanlist>) 169
FETCh[:SCALar]:VOLTage[:DC]? (@<chanlist>) 169
MEASure[:SCALar]:VOLTage[:DC]? (@<chanlist>) 169
FETCh[:SCALar]:VOLTage:ACDC? (@<chanlist>) 170
MEASure[:SCALar]:VOLTage:ACDC? (@<chanlist>) 170
FETCh[:SCALar]:VOLTage:MINimum? (@<chanlist>) 170
MEASure[:SCALar]:VOLTage:MINimum? (@<chanlist>) 170
FETCh[:SCALar]:VOLTage:MAXimum? (@<chanlist>) 171
MEASure[:SCALar]:VOLTage:MAXimum? (@<chanlist>) 171
FETCh[:SCALar]:AHOur? (@<chanlist>) 171
MEASure[:SCALar]:AHOur? (@<chanlist>) 171
FETCh[:SCALar]:WHOur? (@<chanlist>) 172
MEASure[:SCALar]:WHOur? (@<chanlist>) 172
FETCh[:SCALar]:TIME? (@<chanlist>) 172
MEASure[:SCALar]:TIME? (@<chanlist>) 172
FTECh[:SCALar][:ALL]? (@<chanlist>) 172
MEASure[:SCALar][:ALL]? (@<chanlist>) 172
FETCh:ARRay:CURRent[:DC]? (@<chanlist>) 173
MEASure:ARRay:CURRent[:DC]? (@<chanlist>) 173
FETCh:ARRay:VOLTage[:DC]? (@<chanlist>) 173
MEASure:ARRay:VOLTage[:DC]? (@<chanlist>) 173
FETCh:ARRay:POWer[:DC]? (@<chanlist>) 174
MEASure:ARRay:POWer[:DC]? (@<chanlist>) 174
SENSe[:REMote][:STATe] 0|1|OFF|ON, (@<channlist>) 175
SENSe:SWEep:OFFSet:POINts <value>, (@<chanlist>) 175
SENSe:SWEep:POINts <value>, (@<chanlist>) 176
SENSe:SWEep:TINTerval <value>, (@<chanlist>) 176
SENSe:NPLCycles? [MINimum|MAXimum,](@<chanlist>) 177
SENSe:APERture <time|MINimum|MAXimum>,(@<chanlist>) 177
SENSe:MEASure:TIME:MODE <NPLC|APERture>,(@<chanlist>) 178
SENSe:AHOur:CLEar (@<chanlist>) 178
SENSe:WHOur:CLEar (@<chanlist>) 178
SENSe:AHOur:CLEar:AUTO <0|1|OFF|ON>,(@<chanlist>) 179
SENSe:WHOur:CLEar:AUTO <0|1|OFF|ON>,(@<chanlist>) 179
SENSe:DLOG:FUNCtion:CURRent 0|OFF|1|ON, (@<chanlist>) 180
SENSe:DLOG:FUNCtion:VOLTage 0|OFF|1|ON, (@<chanlist>) 180
SENSe:DLOG:FUNCtion:MINMax 0|OFF|1|ON 180
SENSe:DLOG:MARKer1:POINt <value> 181
SENSe:DLOG:MARKer2:POINt <value> 181
SENSe:DLOG:OFFSet <value> 182
SENSe:DLOG:PERiod <value> 182
SENSe:DLOG:TIME <value> 182
SENSe:DLOG:TINTerval <value> 183
SENSe:ELOG:FUNCtion:CURRent 0|OFF|1|ON, (@<chanlist>) 183
SENSe:ELOG:FUNCtion:VOLTage 0|OFF|1|ON, (@<chanlist>) 184
SENSe:ELOG:FUNCtion:CURRent:MINMax 0|OFF|1|ON, (@<chanlist>) 184

Copyright ©ITECH Electronic Co., Ltd.

SENSe:ELOG:FUNCtion:VOLTage:MINMax 0|OFF|1|ON, (@<chanlist>) *184*
SENSe:ELOG:PERiod <value>, (@<chanlist>) *185*
[SOURce:]LIST:COUNt <value>, (@<chanlist>) *185*
[SOURce:]LIST:CURRent:STEP[:LEVel] <value>,{<step>,(@<chanlist>) *186*
[SOURce:]LIST:CURRent:SLEW <value>,<step>, (@<chanlist>) *186*
[SOURce:]LIST:CURRent:POINts <value>,(@<chanlist>) *187*
[SOURce:]LIST:VOLTage:STEP[:LEVel] <value>,<step>, (@<chanlist>) *187*
[SOURce:]LIST:VOLTage:SLEW <value>,<step>, (@<chanlist>) *187*
[SOURce:]LIST:VOLTage:POINts <value>,(@<chanlist>) *188*
[SOURce:]LIST:POWer:STEP[:LEVel] <value>,<step>, (@<chanlist>) *188*
[SOURce:]LIST:POWer:SLEW <value>,<step>, (@<chanlist>) *188*
[SOURce:]LIST:POWer:POINts <value>,(@<chanlist>) *189*
[SOURce:]LIST:RESistance:STEP[:LEVel] <value>,<step>, (@<chanlist>) *189*
[SOURce:]LIST:RESistance:SLEW <value>,<step>, (@<chanlist>) *190*
[SOURce:]LIST:RESistance:POINts <value>,(@<chanlist>) *190*
[SOURce:]LIST:TERMinate:MODE <NORMal|LAST|OFF>,(@<chanlist>) *190*
[SOURce:]LIST:DWELl <value>,<step>, (@<chanlist>) *191*
[SOURce:]LIST:TOUTput:BOSTep:POINts? (@<chanlist>) *191*
[SOURce:]LIST:TOUTput:EOSTep:POINts? (@<chanlist>) *192*
[SOURce:]LIST:STEP ONCE|AUTO,<step>(@<chanlist>) *192*
[SOURce:]LIST:TOUTput:BOSTep[:DATA] 0|OFF|1|ON,<step>,(@<chanlist>) *192*
[SOURce:]LIST:TOUTput:EOSTep[:DATA] 0|OFF|1|ON,<step>,(@<chanlist>) *192*
[SOURce:]LIST:COUNt:RUN? (@<chanlist>) *193*
[SOURce:]LIST:STEP:RUN? (@<chanlist>) *193*
[SOURce]LIST:FILE:OPEN <string>,LOCal|USB,(@<chanlist>) *193*
[SOURce]LIST:FILE:SAVE <string>,LOCal|USB,(@<chanlist>) *194*
[SOURce:]LIST:FUNCtion:TYPE <type>,(@<chanlist>) *194*
TRIGger:ACQuire[:IMMediate] (@<chanlist>) *194*
TRIGger:ACQuire:SOURce <source>, (@<chanlist>) *194*
TRIGger:ACQuire:VOLTage[:LEVel] <value>, (@<chanlist>) *195*
TRIGger:ACQuire:VOLTage:SLOPe POSitive|NEGative, (@<chanlist>) *195*
TRIGger:ACQuire:CURRent[:LEVel] <value>, (@<chanlist>) *195*
TRIGger:ACQuire:CURRent:SLOPe POSitive|NEGative, (@<chanlist>) *196*
TRIGger:ELOG[:IMMediate] (@<chanlist>) *196*
TRIGger:ELOG:SOURce <source>, (@<chanlist>) *196*
TRIGger:DLOG[:IMMediate] (@<chanlist>) *197*
TRIGger:DLOG:SOURce <source> *197*
TRIGger:DLOG:VOLTage[:LEVel] <value>, (@<chanlist>) *198*
TRIGger:DLOG:VOLTage:SLOPe POSitive|NEGative, (@<chanlist>) *198*
TRIGger:DLOG:CURRent[:LEVel] <value>, (@<chanlist>) *198*
TRIGger:DLOG:CURRent:SLOPe POSitive|NEGative, (@<chanlist>) *199*
TRIGger:TRANsient[:IMMediate] (@<chanlist>) *199*
TRIGger:MAP:DC[:STATe] 0|OFF|1|ON, (@chanlist) *199*
TRIGger:MAP:DC:PIN PIN<1-7>, (@chanlist) *200*
TRIGger:MAP:ASToutput[:STATe] 0|1|OFF|ON, (@chanlist) *200*
TRIGger:MAP:ASToutput:PIN PIN<1-7>, (@chanlist) *201*
[SOURce:]DIGital:INPut:DATA? *201*
[SOURce:]DIGital:OUTPut:DATA <value> *201*
[SOURce:]DIGital:PIN<1-7>:FUNCtion <function> *202*
[SOURce:]DIGital:PIN<1-7>:POLarity POSitive|NEGative *202*
[SOURce:]DIGital:ONOFf:STATe:CHANnel <1-8> *203*
[SOURce:]DIGital:PLUSe:WIDTh <value> *203*
[SOURce:]TRANsient:COUNt <value>|MINimum|MAXimum, (@<chanlist>) *204*
[SOURce:]TRANsient:COUNt:RUN? (@<chanlist>) *204*
[SOURce:]TRANsient:DCYCle <value>|MINimum|MAXimum, (@<chanlist>) *204*
[SOURce:]TRANsient:FREQuency <value>|MINimum|MAXimum,(@<chanlist>) *205*
[SOURce:]TRANsient:PERiod <value>|MINimum|MAXimum,(@<chanlist>) *205*
[SOURce:]TRANsient:MODE CONTinuous | PULSe | TOGGle,(@<chanlist>) *206*
[SOURce:]TRANsient:TWIDth <value> | MINimum | MAXimum, (@<chanlist>) *206*

Copyright ©ITECH Electronic Co., Ltd.

[SOURce:]TRANsient:CURRent:TLEVel <transient level>|MINimum|MAXimum,(@<chanlist>) *206*
[SOURce:]TRANsient:VOLTage:TLEVel <transient level>|MINimum|MAXimum,(@<chanlist>) *207*
[SOURce:]TRANsient:POWer:TLEVel <transient level>|MINimum|MAXimum,(@<chanlist>) *207*
[SOURce:]TRANsient:RESistance:TLEVel <transient level>|MINimum|MAXimum,(@<chanlist>) *208*
BATTery:DISCharge:MODE <CURRent|VOLTage>,(@<chanlist>) *208*
BATTery:DISCharge:VOLTage <value>,(@<chanlist>) *208*
BATTery:DISCharge:CURRent <value>,(@<chanlist>) *209*
BATTery:DISCharge:POWer <value>,(@<chanlist>) *209*
BATTery:DISCharge:RESistance <value>,(@<chanlist>) *210*
BATTery:DISCharge:STOP:VOLTage <value>,(@<chanlist>) *210*
BATTery:DISCharge:STOP:VOLTage:STATe <0|1|OFF|ON>,(@<chanlist>) *210*
BATTery:DISCharge:STOP:CURRent <value>,(@<chanlist>) *211*
BATTery:DISCharge:STOP:CURRent:STATe <0|1|OFF|ON>,(@<chanlist>) *211*
BATTery:DISCharge:STOP:AHOur <value>,(@<chanlist>) *212*
BATTery:DISCharge:STOP:AHOur:STATe <0|1|OFF|ON>,(@<chanlist>) *212*
BATTery:DISCharge:STOP:WHOur <value>,(@<chanlist>) *212*
BATTery:DISCharge:STOP:WHOur:STATe <0|1|OFF|ON>,(@<chanlist>) *213*
BATTery:DISCharge:STOP:TIME <value>,(@<chanlist>) *213*
BATTery:DISCharge:STOP:TIME:STATe <0|1|OFF|ON>,(@<chanlist>) *214*
BATTERY:DISCharge:SAVe <"filename">,(@<chanlist>) *214*
BATTery:DISCharge:RECall <"filename">,(@<chanlist>) *214*
BATTery:DISCharge:DELete <"filename">,(@<chanlist>) *215*
BATTery:DISCharge:TIME? (@<chanlist>) *215*
BATTery:DISCharge:RSTate? (@<chanlist>) *215*

Chapter7 IEEE-488 Common Commands 216

*CLS 216
*ESE 216
*ESE? 217
*ESR? 217
*IDN? 217
*OPC 218
*RST 218
*SRE <NR1> 218
*STB? 219
*PSC 219
*SAV 220
*RCL 220

Copyright ©ITECH Electronic Co., Ltd.

Chapter1SCPI Introduction

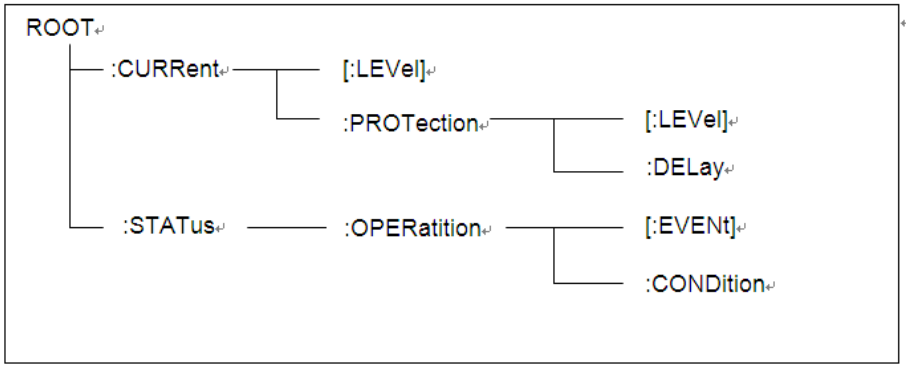
1.1 Overview

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. A part of OUTPut subsystem is listed below:

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 commoncommands 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",

Copyright ©ITECH Electronic Co., Ltd.

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:STATe ON

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.

Case Sensitivity

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

***RST = *rst**

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:ERRor? long-form

:SYST:ERR? short form

:SYSTem:ERR? 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:Erro?** is illegal and will generate an error. The command will not be executed.

Query

Observe the following precautions with queries:

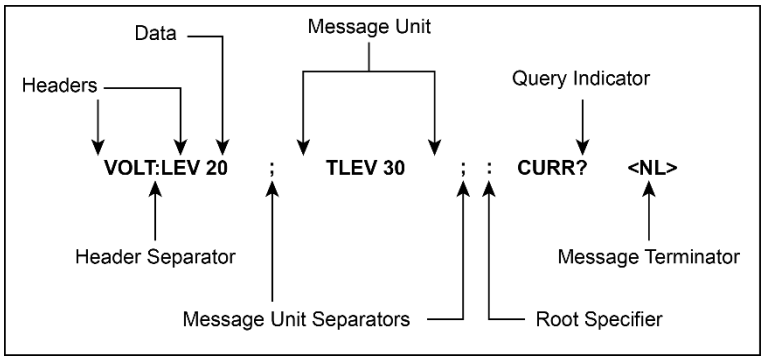
- Set up the proper number of variables for the returned data. For Example, if you are reading back a measurement array, you must dimension the array according to the number of measurements that you have placed in the measurement buffer.
- Read back all the results of a query before sending another command 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").

Copyright ©ITECH Electronic Co., Ltd.

- Some syntax elements (such as nodes and Parameter) are included in square brackets ([ ]). It indicates that these elements can be selected and omitted. Angle brackets are not sent along with command strings. If no value is assigned to the optional Parameter, the instrument will select a default value. In the above Examples, "SOURce[1|2]" indicates that you may refer to source channel 1 by "SOURce" or "SOURce1" or "SOUR1" or "SOUR". In addition, since the whole SOURce node is optional (in the square bracket), you can refer to the channel 1 by omitting the whole SOURce node. It is because the channel 1 is the default channel for SOURce language node. On the other hand, if you want to refer to channel 2, "SOURce2" or "SOUR2" must be used in the program line.

Colon (:)

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

APPL:SIN 455E3,1.15,0.0

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

Semicolon (;)

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

TRIG:SOUR EXT; COUNT 10

has the same effect as sending the following two commands:

TRIG:SOUR EXT
TRIG:COUNT 10

Question mark (?)

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

TRIG:COUN 10

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

TRIG:COUN?

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

TRIG:COUN?MIN
TRIG:COUN?MAX

Comma (,)

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

Space

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

Common commands (*)

The IEEE-488.2 standard defines a set of common commands that perform

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

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

Command terminator

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

NOTE

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

1.6 Data Type

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

- Numerical parameter

  Commands requiring numerical parameter support the notations of all common decimal notations, including optional signs, decimal points, scientific notation, etc. Special values of numerical parameter are also acceptable, such as MIN, MAX and DEF. In addition, suffixes for engineering units can also be sent together with numerical parameter (including M, k, m or u). If the command accepts only some specific values, the instrument will automatically round the input parameter to acceptable values. The following commands require numerical parameter of frequency value:
  [SOURce[1|2]:]FREQuency:CENTer {<Frequency>|MINimum|MAXimum}
  – **<NR1>**: represents an integer value, such as 273;
  – **<NR2>**: represents a real number in floating-point format, such as .273;
  – **<NR3>**: represents a real number in scientific notation, such as 2.73E+2;
  – **<Nrf>**: The extensible form includes <NR1>, <NR2> and <NR3>;
  – **<Nrf+>**: The extensible decimal form includes <Nrf>, MIN, MAX and DEF. MIN and MAX are the minimum and maximum finite number. Within the range of the parameter definition, DEF is the default of the parameter.

- Discrete parameter

  Discrete parameter are used for settings with limited number of programming values (such as IMMediate, EXTernal or BUS). They can use short and long format like key words of commands. They may be expressed in both upper and lower case. The query response always returns uppercase Parameter in short format. The following commands require discrete parameter in voltage unit:
  [SOURce[1|2]:]VOLTage:UNIT {VPP|VRMS|DBM}

- Boolean parameter

  Boolean parameter refer to true or false binary conditions. In case of false

conditions, the instrument will accept "OFF" or "0". In case of true conditions, the instrument will accept "ON" or "1". In query of Boolean settings, the instrument will always return "0" or "1". Boolean parameter are required by the following commands:
DISPlay {OFF|0|ON|1}

- ASCII string parameter

  String parameter may actually include all ASCII character sets. Character strings must start and end with paired quotation marks; and single quotation marks or double quotation marks are both allowed. Quotation mark separators may also act as one part of a string, they can be typed twice without any character added between them. String parameter is used in the following command:
  DISPlay:TEXT <quoted string>
  For Example, the following commands display message of "WAITING..." (without quotation marks) on the front panel of the instrument.
  DISP:TEXT "WAITING..."
  Single quotation marks may also be used to display the same message.
  DISP:TEXT 'WAITING...'
  – **<SPD>**: string program data. String parameters enclosed in single or double quotes.
  – **<CPD>**: character program data.

1.7 Remote Interface Connection

IT8200 series electronic load are equipped with three communication interfaces as standard: USB, LAN and CAN, and two optional interfaces: RS232 and GPIB. Users can choose any one to realize communication with the computer. For detailed introduction of remote interface connection, please refer to the content in the user manual.

Chapter2Example of Common Commands

This chapter displays the programming examples to remotely control IT2700 power supply using SCPI commands.

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.

Example 1:Identify the power supply

You can verify that you are communicating with the correct IT2700 power supply.

To query the identification of the power supply, enter the following command:

-> *IDN?

To check the error queue of the power supply, enter the following command:

-> SYST:ERR?

Example 2:Common output commands

The IT2700 series instruments can be connected to a PC via a communication interface to realize remote operation of the instrument by sending SCPI commands.

The following common commands to help you quickly implement common operations. For more command information, refer to the corresponding section. You can omit the small write parts in the following commands.

SYST:REM \\ Set the power supply to remote operation mode.

SYST:CHAN:STAT? (@1:8) \\ Read the online channels

CV Priority:

FUNC VOLT ,(@3,5) \\ Setting CH3 and CH5 voltage priority mode

VOLT 10,(@3,5) \\ Set CH3 and CH5 voltage to 10V.

CURR:LIM:COUP OFF,(@3,5) \\ Setting the tracking state of CH3 and CH5 current positive and negative limit values to not coupling.

CURR:LIM 2,(@3,5) \\ Setting CH3 and CH5 I+ to 2A

CURR:LIM:NEG -2.5,(@3,5) \\ Setting CH3 和 CH5 I- to -2.5A ;

CC Priority:

FUNC CURR ,(@3,5) \\ Setting CH3 and CH5 current priority mode

CURRent 5,(@3,5) \\ Setting CH3 and CH5 current to 5A

CURRent -5,(@3,5) \\ Setting CH3 and CH5 current to -5A

VOLT:LIM:HIGH 10,(@3,5) \\ Setting CH3 and CH5 VH to 10V

VOLT:LIM:LOW 0,(@3,5) \\ Setting CH3 and CH5 VL to 0V

Open the output

OUTP ON ,(@3,5) \\ Setting CH3 and CH5 Instrument output state on

MEAS:VOLT? (@3,5) \\ Query the CH3 and CH5 output

Copyright ©ITECH Electronic Co., Ltd.

```
MEAS:CURR? (@3,5)          \\ Query the CH3 and CH5 output current
```

Chapter3SCPI Status Register

IT2700 series instrument records variable instrument status via three kinds of status register, the three kinds of registers are standard event register, query status register and status bytes register. Status byte register records the information of other status registers.

Event register is read only register, used to store the implementation status of the power, the data used in the event register latches in the form, once the data is stored, subsequent data will be completely ignored. Even cannot be changed by re-setting order (* RST) or equipment restart.

but if query the data in event register or send clear order *CLS (clear status), the event register will be automatically cleared. The main content which the standard event register records is:

Whether power output is turned on, order syntax errors, order execution errors, self-test or calibration errors, query errors and so on.

Operate Event			
Bit	Bit Name	Decimal Value	Description
0	ACQ-WTG	1	Measuring system waiting to be triggered
1	TRAN-WTG	2	Transient System Waiting Trigger
2	DLOG-WTG	4	DLOG system waiting to be triggered
3	ACQ-Active	8	Measurement system in operation
4	TRAN-Active	16	Transient system in operation
5	DLOG- Active	32	DLOG system is operational
6	OFF	64	Output off state
7	CC	128	Output operates in constant current mode
8	CV	256	Output operates in constant voltage mode
9	CW	512	Output operates in constant power mode
10	CR	1024	Output operates in constant resistance mode
11	CC-	2048	Output operates in constant negative current mode
12	CP-	4096	Output operates in constant negative power mode
NU	13-31	0	reserved bit

Copyright ©ITECH Electronic Co., Ltd.

Questionable Status			
Bit	Bit Name	Decimal Value	Description
0	OV	1	Output is shut down by overvoltage protection
1	OC	2	Output shut down by overcurrent protection
2	OP	4	Output shut down by over power protection
3	UV	8	Output shut down by undervoltage protection
4	UC	16	Output is shut down by undercurrent protection
5	Rvs	32	Output is shut down by reverse connection protection
6	Errsense	64	Outputs are turned off by sense protection
7	OT	128	Output is shut down by over-temperature protection
8	PCP	256	Output is turned off by output coupling protection
9	FOLDBACK	512	Output is turned off by foldback protection
10	WDOG	1024	Output protected by watchdog
11	LIVING	2048	The output is disabled by Living protection, which does not affect the onoff state, but stops the output.
12	LATCH	4096	Outputs shut down by Latch protection
13	PF	8192	Output is shut down by APFC power module protection
14	MF	16384	Output shut down by power module protection
15-31	NULL		reserved bit

Standard Envent			
0	Operation Complete	1	All instructions, including *OPC, have been executed.
1	not used	2	Returns a value of 0
2	Query Error	4	The instrument attempted to read the output buffer, but the buffer is empty, or a new command line was received before the last query was read, or both the input and output buffers are full.

Copyright ©ITECH Electronic Co., Ltd.

3	Device-Specific Error	8	A device-specific error has occurred, including a self-test error, calibration error, or other device-specific error.
4	Execution Error	16	Generate an execution error
5	Command	32	Generate a command syntax error
6	not used	64	Returns a value of 0
7	Power On	128	Power has been turned off and on since the last time the event register was read or cleared

Status Byte Register			
Bit	Bit Name	Decimal Value	Description
0	not used	1	Return 0
1	not used	2	Return 0
2	Error Queue	4	There are one or more errors in the error queue, use SYSTem:ERRor? to read and delete the errors.
3	Questionable Status Summary	8	One or more bits are set and queried in the Problem Data Register and must be enabled, see STATus:QUEStionable:ENABle
4	Message Available	16	Data available in the instrument output buffer
5	Event Status Summary	32	One or more bits are set and queried in the standard event register.
6	Master Status Summary	64	One or more bits in the Status Byte Register are set and queried and may generate service requests. The bits must be enabled, details build *SRE.
7	Operation Status Summary	128	One or more bits are set and queried in the Operation Status Register. The bit bits must be enabled as detailed in the STATus:OPERation:ENABle

Copyright ©ITECH Electronic Co., Ltd.

Chapter4STATus Commands

STATus:QUEStionable[:EVENt]? (<@chanlist>)

This command can be used to read the value in query event register.The power supply will return a decimal number corresponding to the binary weighted sum of each unit digit of register. These digits will be latched. After executing this command, the query event register will be cleared.

Syntax

STATus:QUEStionable[:EVENt]? (<@chanlist>)

Arguments

Null

Returns

<NR1>

Example

-> STAT:QUES? (@1)
<- 4140

Related commands

STATus:QUEStionable:ENABle

STATus:QUEStionable:CONDition? (<@chanlist>)

This command is used to read the value of query condition register.

This is a read-only register, which directly reflects those bits in the instrument's problem state that are set to 1 or 0, and will not be latched. After the command is executed, the value of the query condition register will not be cleared.

Syntax

STATus:QUEStionable:CONDition?

Arguments

Null

Returns

<NR1> This is a decimal value with a binary weighted sum.

STATus:QUEStionable:ENABle <value>,(<@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.

Copyright ©ITECH Electronic Co., Ltd.

Syntax

STATus:QUEStionable:ENABle <value>,(<@chanlist>)

Arguments

0-0x7fffffff

Default value

0|OFF

Example

STAT:QUES:ENAB 128,(@1)

Query syntax

STATus:QUEStionable:ENABle? (<@chanlist>)

Returns

<NR1>

STATus:QUEStionable:NTRansition <value>, (<@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.

Syntax

STATus:QUEStionable:NTRansition <value>, (<@chanlist>)

Arguments

0-0x7fffffff

Default value

0

Example

STAT:QUES:NTR 128,(@1)

Query syntax

STATus:QUEStionable:NTRansition? (<@chanlist>)

Returns

<value>

STATus:QUEStionable:PTRansition <value>, (<@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.

Syntax

STATus:QUEStionable:PTRansition <value>,(<@chanlist>)

Arguments

0-0x7fffffff

Default value

0

Example

STAT:QUES:PTR 128,(@1)

Query syntax

STATus:QUEStionable:PTRansition? (<@chanlist>)

Returns

<NR1>

STATus:OPERation[:EVENt]? (<@chanlist>)

This command can read the parameter from operation event register. After executing this order, operation event register is reset.

Syntax

STATus:OPERation[:EVENt]? (<@chanlist>)

Arguments

无

Returns

<NR1>

Related commands

STATus:OPERation:ENABle

STATus:OPERation:CONDition? (<@chanlist>)

This command can read the parameter from the operation condition register.

This is a read-only register, which directly reflects those bits in the operating state

of the instrument are set to 1 or 0, and will not be latched. After the command is executed, the value of the query condition register will not be cleared.

Syntax

STATus:OPERation:CONDition? (<@chanlist>)

Arguments

无

Example

STATus:OPERation:CONDition?

Returns

<NR1>

STATus:OPERation:ENABle <value>,(<@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.

Syntax

STATus:OPERation:ENABle <value>, (<@chanlist>)

Arguments

0-0x7fffffff

Default value

0|OFF

Example

STATus:OPERation:ENABle 128,(@1)

Query syntax

STATus:OPERation:ENABle?

Returns

<NR1>

STATus:OPERation:NTRansition <value>,(<@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.

Syntax

STATus:OPERation:NTRansition <value>, (<@chanlist>)

Arguments

0-0x7fffffff

Default value

0

Example

STATus:OPERation:NTRansition 128,(@1)

Query syntax

STATus:OPERation:NTRansition?

Returns

<NR1>

STATus:OPERation:PTRansition <value>,(<@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.

Syntax

STATus:OPERation:PTRansition <value>, (<@chanlist>)

Arguments

0-0x7fffffff

Default value

0

Example

STATus:OPERation:PTRansition 128,(@1)

Query syntax

STATus:OPERation:PTRansition?

Returns

<NR1>

STATus:PRESet

This command is used to preset the value of the status register.

Syntax

STATus:PRESet

Example

STAT:PRES

Chapter5DC power Commands

IT2700 series multi-channel source and load module system can realize different functions by installing different modules, the module categories are bidirectional DC power supply module, unipolar power module and load module, this chapter introduces the commands related to bidirectional DC power supplies module and unipolar power supplies module.

5.1 SYSTem Subsystem

SYSTem:VERSion?

This command queries the SCPI version of the instrument.

Syntax

SYSTem:VERSion?

Arguments

None

Reset value

Not applicable

Example

- > SYST:VERS?
< - "1993.1"

Returns

SRD

SYSTem:ERRor[:NEXT]?

This order is used to query the error information of the power supply. When the error LED lights on the front panel, it tells us that there is one or more hardware or order syntax error happened. There are at most 20 groups of error message. Send the order once will read one error message from the error queue.

- Error messages follow the FIFO (first-in-first-out) principle. The first error will be returned first. when you read all the error tips information in the error queue. The ERROR led will turn off. When error occur the buzzer of the power supply will beep once.
- If more than 20 errors occurred, the last one stored in the queue will be replaced by “-350”, Meaning "too many mistakes.". If the error messages in the error message queue are not read out, the error following will not be saved to the queue. If there is no error message, when read error information, it will return “+0”, meaning “no error”.
- If turn off the power supply or send *CLS (clear status) order, the error messages in the error queue will be cleared. *RST order will not clear the error messages in the error queue.

SYSTem:CHANnel[:COUNt]?

Returns the number of output channels in a mainframe.

Syntax

SYSTem:CHANnel[:COUNt]?

Example

SYST:CHAN?

SYSTem:CHANnel:MODel? (@<chanlist>)

Returns the model numbers of the selected channels. Multiple responses are separated by commas.

Syntax

SYSTem:CHANnel:MODel? (@<chanlist>)

Example

SYST:CHAN:MOD? (@1)

SYSTem:CHANnel:OPTion? (@<chanlist>)

Returns the options installed in the selected channels. The query returns an ASCII string enclosed in double quotes. If no option is installed, an empty quoted string ("") is returned. Multiple responses are separated by commas.

Syntax

SYSTem:CHANnel:OPTion? (@<chanlist>)

Example

SYSTem:CHANnel:OPTion? (@1)

SYSTem:CHANnel:DESCription? (@<chanlist>)

Returns the description of the selected channels.Multiple responses are separated by commas.

Syntax

SYSTem:CHANnel:DESCription? (@<chanlist>)

Example

SYST:CHAN:DESC? (@1)

SYSTem:CHANnel:SERial? (@<chanlist>)

Returns the serial numbers of the selected channels. Multiple responses are separated by semicolons.

Syntax

SYSTem:CHANnel:SERial? (@<chanlist>)

Example

SYST:CHAN:SER? (@1)

SYSTem:CHANnel:STATus? (@<chanlist>)

This command is used to read the status of the channel.

Syntax

SYSTem:CHANnel:STATus? (@<chanlist>)

Example

SYST:CHAN:STAT? (@1>)

SYSTem:CHANnel:STATus:SUMMary?

This command is used to read a summary of all states.

Syntax

SYSTem:CHANnel:STATus:SUMMary?

Example

SYST:CHAN:STAT:SUMM?

SYSTem:CHANnel:VERSion? (@<chanlist>)

This command is used to read the version of the channel.

Syntax

SYSTem:CHANnel:VERSion? (@<chanlist>)

Example

SYST:CHAN:VERS? (@1)

SYSTem:CHANnel:CATegory? (@<chanlist>)

This command is used to read the model category of the channel.

Syntax

SYSTem:CHANnel:CATegory? (@<chanlist>)

Arguments

0:DC Source

1:DC Load

2:AC Source

3:AC Load

Example

SYST:CHAN:CAT? (@1)

SYSTem:COMMunicate:RLSTate LOCal|REMote|RWLock

This command is used to set the remote or local state. local: the instrument operation mode is switched to local mode, Remote: the instrument is switched to remote control mode. rwlock: the device goes to the remote state and locks the local LOCAL key.

This command has the same function as SYSTem:REMote/ SYSTem:LOCal/ SYSTem:RWLock.

Syntax

SYSTem:COMMunicate:RLSTate LOCal|REMote|RWLock

Example

SYST:COMM:RLST REM

Query syntax

SYSTem:COMMunicate:RLSTate?

SYSTem:REMote

This command takes the instrument out of front-panel control mode and switches it to remote control mode.

Syntax

SYSTem:REMote

Example

SYST:REM

SYSTem:LOCal

This command is used to switch the power supply into the control from the front panel.

Syntax

SYSTem:LOCal

Example

SYST:LOC

SYSTem:RWLock

This command locks the power supply in remote control mode. When this command is executed, pressing the LOCAL button does not switch the instrument to local control mode.

Syntax

SYSTem:RWLock

Example

SYST:RWL

Copyright ©ITECH Electronic Co., Ltd.

SYSTem:KEY <0-255>

This command is used to simulate a key.

Syntax

SYSTem:KEY <0-255>

Arguments

0-255

Example

SYST:KEY

Query syntax

SYSTem:KEY?

SYSTem:PRESet

This command is used to make the instrument in a state suitable for panel operation(reset).

Syntax

SYSTem:PRESet

Example

SYST:PRES

SYSTem:POSetup <CPD>

This command is used to set and query some parameters or working status 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: 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.

Syntax

SYSTem:POSetup <CPD>

Arguments

RST|LAST|LAST_OFF

*RST

RST

Example

SYST:POS RST

Copyright ©ITECH Electronic Co., Ltd.

Query syntax

SYSTem:POSetup?

SYSTem:CLEar

This command is used to clear the error queue.

Syntax

SYSTem:CLEar

Example

SYST:CLE

SYSTem:BEEPer:IMMediate

This command tests the beeper function of the power supply. If it passes the test, a beep is issued.

Syntax

SYSTem:BEEPer:IMMediate

Example

SYST:BEEP:IMM

SYSTem:BEEPer:CLICk <0|1|OFF|ON>

This command is used to set the button click buzzer.

Syntax

SYSTem:BEEPer:CLICk <0|1|OFF|ON>

Arguments

0|1|OFF|ON

*RST

ON

Return Arguments

0/1

Example

SYSTem:BEEPer:CLICk ON

Query syntax

SYSTem:BEEPer:CLICk?

SYSTem:BEEPer:ALARm <0|1|OFF|ON>

This command is used to set the buzzer sound state in the alarm state.

Syntax

SYSTem:BEEPer:ALARm <0|1|OFF|ON>

Arguments

0|1|OFF|ON

*RST

ON

Return Arguments

0/1

Example

SYSTem:BEEPer:ALARm ON

Query syntax

SYSTem:BEEPer:ALARm?

SYSTem:COMMunicate:USB:TYPE <DEVice|HOST>

This command is used to set and query the USB interface type. IT2700 series power supply USB can be set as the host or device type, used to set the USB interface for communication or as a storage disk. When set as host type, it is used as a storage disk, and when set as device type, it is used as a communication interface.

Syntax

SYSTem:COMMunicate:USB:TYPE <CPD>

Arguments

DEVice|HOST

Default value

HOST

Return Arguments

DEVice|HOST

Example

SYST:COMM:USB:TYPE HOST

Query syntax

SYSTem:COMMunicate:USB:TYPE?

SYSTem:COMMunicate:SELect <TMC|VCP>

This command is used to set and query the USB interface type.

Syntax

SYSTem:COMMunicate:SELect <TMC|VCP>

Arguments

TMC|VCP

*RST

VCP

Return Arguments

TMC|VCP

Example

SYST:COMM:SEL VCP

Query syntax

SYSTem:COMMunicate:SELect?

SYSTem:COMMunicate:GPIB:ADDRess <NR1>

This command is used to set and query the GPIB address.

Syntax

SYSTem:COMMunicate:GPIB:ADDRess <NR1>

Arguments

<NR1>

Range: 1~30

Default value

1

Returns

<NR1>

Example

SYST:COMM:GPIB:ADDR 2

Query syntax

SYSTem:COMMunicate:GPIB:ADDRess?

SYSTem:COMMunicate:SERial:BAUDrate <CPD>

This command is used to set and query the baud rate of the serial port.

Syntax

SYSTem:COMMunicate:SERial:BAUDrate <CPD>

Arguments

<CPD>

115200|57600|38400|19200|9600|4800

Default value

9600

Returns

<CPD>

Example

SYST:COMM:SER:BAUD 4800

Query syntax

SYSTem:COMMunicate:SERial:BAUDrate?

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

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

Syntax

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

Arguments

<SPD>

默认 Arguments

"192.168.0.11"

Return Arguments

<SPD>

Example

SYST:COMM:LAN:IP "192.168.0.11"

Query syntax

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

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

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

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

Syntax

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

Arguments

<CPD>

AUTO|MANual

*RST

MANual

Return Arguments

AUTO|MANual

Example

SYST:COMM:LAN:IP:MODE AUTO

Query syntax

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

SYSTem:COMMunicate:LAN:SMASk <SPD>

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

Syntax

SYSTem:COMMunicate:LAN:SMASk <SPD>

Arguments

<SPD>

*RST

"255.255.255.0"

Return Arguments

<SPD>

Example

SYST:COMM:LAN:SMAS "255.255.255.1"

Query syntax

SYSTem:COMMunicate:LAN:SMASk?

SYSTem:COMMunicate:LAN:DGATeway <SPD>

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

Syntax

SYSTem:COMMunicate:LAN:DGATeway <SPD>

Arguments

<SPD>

*RST

"192.168.0.1"

Example

SYST:COMM:LAN:DGAT "192.168.0.1"

Query syntax

SYST:COMM:LAN:DGAT?

SYSTem:COMMunicate:LAN:RAWSocketport <port>

This command is used to set and query the socket port of the LAN communication.

Syntax

SYSTem:COMMunicate:LAN:RAWSocketport <port>

Arguments

<SPD>

*RST

"30000"

Example

SYSTem:COMMunicate:LAN:RAWSocketport 30001

Query syntax

SYSTem:COMMunicate:LAN:RAWSocketport?

SYSTem:OPERation:CATalog?

This command is used to query the operating modes supported by the device.

Syntax

SYSTem:OPERation:CATalog?

Returns

VOLTage|LOAD|CURRent

SYSTem:OPERation:MODE <CPD>

This command is used to set the operating mode.

Syntax

SYSTem:OPERation:MODE <CPD>

Arguments

VOLTage|LOAD|CURRent

Example

SYSTem:OPERation:MODE VOLT

Query syntax

SYSTem:OPERation:MODE?

Returns

VOLTage

SYSTem:BRIGhtness:LEVel <NR1>

This command is used to set and query the screen brightness of the present power supply, the setting range is 1-10.

Syntax

SYSTem:BRIGhtness:LEVel <NR1>

Arguments

1-10

Return Arguments

<NR1>

*RST

8

Example

SYST:BRIG:LEVel 10

Query syntax

SYSTem:BRIGhtness:LEVel?

SYSTem:TOUCh[:STATe] <CPD>

This command is used to set and query the touch screen status of the screen, 0|OFF means the touch screen is turned off, 1|ON means the touch screen is turned on.

Syntax

SYSTem:TOUCh[:STATe] <CPD>

Arguments

<CPD>

0|OFF|1|ON

Return Arguments

0|1

Example

SYST:TOUC 0

Query syntax

SYST:TOUC?

SYSTem:LANGuage <CPD>

This command is used to set and query the language display of the power supply, Chinese or English can be selected.

Syntax

SYSTem:LANGuage <CPD>

Arguments

ENGLish|CHINese

Example

SYST:LANG CHIN

Query syntax

SYSTem:LANGuage?

SYSTem:SOFT:KEYBoard[:STATe] <CPD>

This command is used to set and query the UI soft keyboard switch of the present power supply.

Syntax

SYSTem:SOFT:KEYBoard[:STATe] <CPD>

Arguments

0|OFF|1|ON

Example

SYST:SOFT:KEYB 1

Query syntax

SYSTem:SOFT:KEYBoard[:STATe]?

SYSTem:KNOB:IMMediate:EFFective <CPD>

This command is used to set and query whether the parameters adjusted by the knob take effect immediately.

Syntax

SYSTem:KNOB:IMMediate:EFFective <CPD>

Copyright ©ITECH Electronic Co., Ltd.

Arguments

0|OFF|1|ON

*RST

0

Return Arguments

0 |1

Example

SYST:KNOB:IMM:EFF 1

Query syntax

SYSTem:KNOB:IMMediate:EFFective?

SYSTem:RUN:TIME?

Query machine running time

More than one day returns: xxd hh:mm:ss, xxd means days, days and minutes, minutes and seconds followed by two spaces between, Within one day: hh:mm:ss

Syntax

SYSTem:RUN:TIME?

Return Arguments

xxd hh:mm:ss

SYSTem:GROup:DEFine (@<chanlist>)

Define many channels as a single group.

Syntax

SYSTem:GROup:DEFine (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4

Example

SYST:GRO:DEF (@1,3,5)

SYSTem:GROup:DELete (@<chanlist>)

Deleting specific channels in a group.

Copyright ©ITECH Electronic Co., Ltd.

Syntax

SYSTem:GROup:DELete (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Example

SYST:GRO:DEL (@3)

SYSTem:GROup:DELete:ALL

Deleting specific channels in a group.

Syntax

SYSTem:GROup:DELete:ALL

Example

SYST:GRO:DEL:ALL

SYSTem:GROup:APPLy

Apply the group setting.

Syntax

SYSTem:GROup:APPLy

Example

SYST:GRO:APPL

SYSTem:REBoot

Reboot the instrument.

Syntax

SYSTem:REBoot

Example

SYST:REB

SYSTem:SECurity:IMMediate

Syntax

SYSTem:SECurity:IMMediate

Example

SYST:SEC:IMM

Copyright ©ITECH Electronic Co., Ltd.

SYSTem:SOURce:SLEW:MODE <RATE|TIME>

This command is used to set and query the slope mode of the source.

Syntax

SYSTem:SOURce:SLEW:MODE <RATE|TIME>

Arguments

<RATE|TIME>

Example

SYST:SOUR:SLEW:MODE RATE

Query syntax

SYSTem:SOURce:SLEW:MODE?

Return Arguments

RATE|TIME

SYSTem:LOAD:CURRent:SYMBol <POSitive|NEGative>

This command is used to set the sign of the load current.

Syntax

SYSTem:LOAD:CURRent:SYMBol <POSitive|NEGative>

Arguments

POSitive|NEGative

Example

SYSTem:LOAD:CURRent:SYMBol POSitive

Query syntax

SYSTem:LOAD:CURRent:SYMBol?

Return Arguments

POSitive|NEGative

5.2 ABORt Subsystem

ABORt:ACQuire (@<chanlist>)

This command is used to stop the meter function.

Syntax

ABORt:ACQuire

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4.

Example

ABOR:ACQ (@1)

ABORt[:ALL]

This command is used to stop all of function.

Syntax

ABORt[:ALL]

Example

ABOR[:ALL]

ABORt:DLOG

This command is used to stop the data logger.

Syntax

ABORt:DLOG

Example

ABOR:DLOG

ABORt:ELOG (@<chanlist>)

This command is used to stop the external logger.

Syntax

ABORt:ELOG (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4.

Example

ABORt:ELOG (@1)

ABORt:TRANsient (@<chanlist>)

This command is used to stop the transient system.

Syntax

ABORt:TRANsient (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) –channels 1 and 4
(@1:4) – channels 1 through 4.

Example

ABORt:TRANsient (@1)

5.3 INITiate Subsystem

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

This command is used to initiates the measurement trigger system.

Syntax

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

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Example

INIT:ACQ (@1)

INITiate[:IMMediate]:DLOG <"filename">

This command is used to initiates internal data logging.

Syntax

INITiate[:IMMediate]:DLOG <"filename">

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4.

Example

INIT:LIST

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

This command is used to initiates external data logging

Syntax

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

Example

INIT:ELOG (@1)

INITiate[:IMMediate]:TRANsient (@<chanlist>)

This command is used to initiates the transient trigger system.

Syntax

INITiate[:IMMediate]:TRANsient (@<chanlist>)

Example

INIT:TRAN (@1)

INITiate:CONTinuous[:ACQuire] 0|OFF|1|ON,(@<chanlist>)

This command is used to set the continuous startup state of the acquisition trigger system.

Syntax

INITiate:CONTinuous[:ACQuire] 0|OFF|1|ON,(@<chanlist>)

Arguments

0|1|OFF|ON

Query syntax

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

Example

INITiate:CONTinuous 0,(@1)

INITiate:CONTinuous:TRANsient 0|OFF|1|ON,(@<chanlist>)

This command sets the continuous startup state of the transient trigger system.

Syntax

INITiate:CONTinuous:TRANsient 0|OFF|1|ON,(@<chanlist>)

Arguments

0|1|OFF|ON

Query syntax

INITiate:CONTinuous:TRANsient? (@<chanlist>)

Example

INITiate:CONTinuous:TRANsient 0,(@1)

5.4 SOURce Subsystem

[SOURce:]FUNCtion CURRent|VOLTage, (@<chanlist>)

This command is used to set and query the priority mode of the function.

Syntax

[SOURce:]FUNCtion CURRent|VOLTage|POWer|RESistance|CC+CV|CR+CV|CP+CV|CC+CR|CC+CV+CP+CR|BSIM, (@<chanlist>)

Arguments

Source mode: CURRent|VOLTage|POWer|RESistance

Load mode:

CURRent|VOLTage|POWer|RESistance|CC+CV|CR+CV|CP+CV|CC+CR|CC+CV+CP+CR|BSIM

Query syntax

[SOURce:]FUNCtion? (@<chanlist>)

*RST

Load mode: CURRent

Source mode: VOLTage

Return Arguments

CURR,VOLT,POW or RES

Example

FUNC CURR ,(@1)

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

This command sets the immediate voltage level of the specified output channel. Units are in volts. The immediate level is the output voltage setting.

Syntax

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

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? [MIN|MAX,] (@<chanlist>)

*RST

0

Return Arguments

0-maximum|MINimum|MAXimum

Example

VOLTage 100, (@1)

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>)

This command sets the triggered voltage level of the specified output channel. Units are in volts.

Syntax

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]? [MIN|MAX,] (@<chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

VOLTage:TRIGgered 50, (@1)

[SOURce:]VOLTage:LIMit:HIGH[:IMMediate][:AMPLitude] <value>,(@<chanlist>)

This command sets the voltage upper limit under CC priority mode.

For bidirectional programmable DC power supply,

if [SOURce:]VOLTage:LIMit:COUPle is enabled, this command also sets the value of the voltage lower limit.

Syntax:

VOLTage:LIMit:HIGH[:IMMediate][:AMPLitude] <value>,(@<chanlist>)

Arguments:

<value>：0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax:

[SOURce:]VOLTage:LIMit:HIGH[:IMMediate][:AMPLitude]?
[MINimum|MAXimum,](@<chanlist>)

Example:

VOLTage:LIMit:HIGH 50,(@1)

[SOURce:]VOLTage:LIMit:LOW[:IMMediate][:AMPLitude] <value>,(@<chanlist>)

This command sets the voltage lower limit under CC priority mode, this command is only required for bidirectional programmable DC power supply.

if [SOURce:]VOLTage:LIMit:COUPle is enabled, this command also sets the value of the voltage lower limit.

Syntax:

[SOURce:]VOLTage:LIMit:LOW[:IMMediate][:AMPLitude] <value>,(@<chanlist>)

Arguments:

<value>：0-maximum|MINimum|MAXimum

<chanlist>: Channel number

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax:

[SOURce:]VOLTage:LIMit:LOW[:IMMediate][:AMPLitude]?
[MINimum|MAXimum,](@<chanlist>)

Example:

VOLTage:LIMit:LOW 10,(@1)

[SOURce:]VOLTage:LIMit:COUPle 0|OFF|1|ON,(@<chanlist>)

This command sets the voltage limit tracking state. The enabled state is On (1); the disabled state is Off (0). If limit tracking is enabled, The lower voltage limit tracks the upper voltage limit setting.

Syntax:

[SOURce:]VOLTage:LIMit:COUPle 0|OFF|1|ON,(@<chanlist>)

Arguments:

0|OFF|1|ON

Copyright ©ITECH Electronic Co., Ltd.

Query syntax:

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

Example

VOLT:LIM:COUP ON,(@1)

[SOURce:]VOLTage:PROTection[:LOCal][:POSitive][:LEVel] <value>,(@<chanlist>)

This command sets the over-voltage protection (OVP) level of the output channel. The values are programmed in volts. If the output voltage exceeds the OVP level, the output is disabled.

Syntax

[SOURce:]VOLTage:PROTection[:LOCal][:POSitive][:LEVel] <value>, (@<chanlist>)

Query syntax

[SOURce:]VOLTage:PROTection[:LOCal][:POSitive][:LEVel]? [MIN|MAX,] (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

*RST

MAX

Example

VOLTage:PROTection 300, (@1)

[SOURce:]VOLTage:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>)

This command enables or disables the over-current protection (OVP) function. The enabled state is On (1); the disabled state is Off (0). If the over-voltage protection function is enabled and the output is disabled.

Syntax

[SOURce:]VOLTage:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>)

Arguments

<0|OFF|1|ON>0-OFF|1-ON

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

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

Example

VOLTage:PROTection:STATe ON, (@1)

[SOURce:]VOLTage:PROTection:DELay[:TIME] <value>, (@<chanlist>)

This command sets the over-voltage protection delay. The over-voltage protection function will not be triggered on the specified output channel during the delay time. After the delay time has expired, the over-voltage protection function will be active.

Syntax

[SOURce:]VOLTage:PROTection:DELay[:TIME] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]VOLTage:PROTection:DELay[:TIME]? [MIN|MAX,] (@<chanlist>)

Example

VOLTage:PROTection:DELay 30, (@1)

[SOURce:]VOLTage:UNDer:PROTection[:POSitive][:LEVel] <value>,(@<chanlist>)

This command set and query the positive voltage protection value for undervoltage protection.

Syntax

[SOURce:]VOLTage:UNDer:PROTection[:POSitive][:LEVel] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]VOLTage:UNDer:PROTection[:LEVel]? [MIN|MAX,] (@<chanlist>)

Copyright ©ITECH Electronic Co., Ltd.

Example

VOLTage:UNDer:PROTection 10, (@1)

[SOURce:]VOLTage:UNDer:PROTection:STATe <0|OFF|1|ON>,(@<chanlist>)

This command enables or disables the under-voltage protection (UVP) function. The enabled state is On (1); the disabled state is Off (0). If the under-voltage protection function is enabled and the output is disabled.

Syntax

[SOURce:]VOLTage:UNDer:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>)

Arguments

<0|OFF|1|ON>0-OFF|1-ON

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 2
(@1:4) – channels 1 through 4.

Query syntax

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

Example

VOLTage:UNDer:PROTection:STATe ON, (@1)

[SOURce:]VOLTage:UNDer:PROTection:DELay[:TIME] <value>,(@<chanlist>)

This command sets the under-voltage protection delay. The under-voltage protection function will not be triggered on the specified output channel during the delay time. After the delay time has expired, the under-voltage protection function will be active.

Syntax

[SOURce:]VOLTage:UNDer:PROTection:DELay[:TIME] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]VOLTage:UNDer:PROTection:DELay[:TIME]? [MIN|MAX,] (@<chanlist>)

Example

VOLTage:UNDer:PROTection:DELay 10, (@1)

[SOURce:]VOLTage:UNDer:PROTection:WARM[:TIME] <value>,(@<chanlist>)

This command sets the under-voltage protection warm time. The under-voltage protection function will not be triggered on the specified output channel during the warm time. even if the voltage reaches the under-voltage protection point and meets the delay time.

Syntax

[SOURce:]VOLTage:UNDer:PROTection:WARM[:TIME] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]VOLTage:UNDer:PROTection:WARM[:TIME]? [MIN|MAX,] (@<chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

VOLTage:UNDer:PROTection:WARM 10, (@1)

[SOURce:]VOLTage:RESistance[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>)

This command sets and queries the internal resistance value of the power supply in voltage priority mode. Units are in Ω.

Syntax

[SOURce:]VOLTage:RESistance[:LEVel][:IMMediate][:AMPLitude] <value>,(@<chanlist>)

Arguments

<value>MAXimum|MINimum|DEFault

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

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

Return Arguments

MAXimum|MINimum|DEFault

Example

VOLTage:RESistance 0.1,(@1)

[SOURce:]VOLTage:RESistance:STATe <0|OFF|1|ON>, (@<chanlist>)

This command sets and queries the status of the internal resistance function of the power supply in voltage priority mode.

Syntax

[SOURce:]VOLTage:RESistance:STATe <0|OFF|1|ON>, (@<chanlist>)

Arguments

<0|OFF|1|ON>0|OFF|1|ON

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]VOLTage:RESistance:STATe? (@<chanlist>)

Return Arguments

0|OFF|1|ON

Example

VOLTage:RESistance:STATe? (@1,2)

[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

This command sets and queries the voltage rise slope.

If the setting state of the [SOURce:]VOLTage:SLEW:COUPle is ON, this command sets the voltage rise slope while simultaneously setting the voltage fall slope.

Syntax

[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Copyright ©ITECH Electronic Co., Ltd.

Query syntax

[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate]? [MIN|MAX,] (@<chanlist>)

Example

VOLTage:SLEW 10, (@1)

[SOURce:]VOLTage:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>)

This command sets and queries the voltage fall slope.

If the setting state of the [SOURce:]VOLTage:SLEW:COUPle is ON, this command sets the voltage fall slope while simultaneously setting the voltage rise slope.

Syntax

[SOURce:]VOLTage:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]VOLTage:SLEW:NEGative[:IMMediate]? [MIN|MAX,] (@<chanlist>)

Example

VOLTage:SLEW:NEGative 10, (@1)

[SOURce:]VOLTage:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

This command sets and queries the tracking status of the voltage slope.

Syntax

[SOURce:]VOLTage:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]VOLTage:SLEW:COUPle? (@<chanlist>)

Example

VOLTage:SLEW:COUPle ON, (@1)

[SOURce:]VOLTage:SLEW[:POSitive]:MAXimum 0|OFF|1|ON,(@<chanlist>)

Enable or disable voltage rise slope maximum.

Syntax

[SOURce:]VOLTage:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

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

Example

VOLTage:SLEW:MAXimum ON, (@1)

[SOURce:]VOLTage:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

Enable or disable voltage drop slope maximum.

Syntax

[SOURce:]VOLTage:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

Query syntax

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

Example

VOLTage:SLEW:NEGative:MAXimum ON, (@1)

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>)

This command sets the immediate current level of the specified output channel. Units are in amperes. The immediate level is the output current setting.

Syntax

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]? [MIN|MAX,] (@<chanlist>)

Example

CURRent 30, (@1)

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>)

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

Syntax

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]? [MIN|MAX,] (@<chanlist>)

Example

CURRent:TRIGgered 20, (@1)

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

This command sets the positive current limit of the specified output channel. Units are in amperes. If [SOURce:]CURRent:LIMit:COUPle is enabled, this command also sets the value of the negative current limit.

Syntax

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

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]CURRent:LIMit[:POSitive][:IMMediate][:AMPLitude]? [MIN|MAX,] (@<chanlist>)

Example

CURRent:LIMit 50, (@1)

[SOURce:]CURRent:LIMit:NEGative[:IMMediate][:AMPLitude] <value>, (@<chanlist>)

This command sets the negative current limit of the specified output channel. Units are in amperes. If [SOURce:]CURRent:LIMit:COUPle is enabled, this command also sets the value of the positive current limit.

Syntax

[SOURce:]CURRent:LIMit:NEGative[:IMMediate][:AMPLitude] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]CURRent:LIMit:NEGative[:IMMediate][:AMPLitude]? [MINimum|MAXimum,] (@<chanlist>)

Example

CURR:LIM:NEG -2,(@1)

[SOURce:]CURRent:LIMit:COUPle 0|OFF|1|ON, (@<chanlist>)

This command sets the current limit tracking state. The enabled state is On (1); the disabled state is Off (0). If limit tracking is enabled, the negative current limit tracks the positive current limit setting. The [SOURce:]CURRent:LIMit command sets the value of the positive and negative current limits.

This command only applies to bidirectional DC power supply models.

Syntax

[SOURce:]CURRent:LIMit:COUPle 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

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

Example

CURR:LIM:COUP ON,(@1)

[SOURce:]CURRent:PROTection[:LEVel][:POSitive] <level>,(@<chanlist>)

This command sets the positive current protection value for over-current protection.

Syntax

[SOURce:]CURRent:PROTection[:LEVel][:POSitive] <level>,(@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]CURRent:PROTection[:LEVel][:POSitive]?
[MINimum|MAXimum,](@<chanlist>)

*RST

MAX

Example

CURRent:PROTection 300, (@1)

[SOURce:]CURRent:PROTection[:LEVel]:NEGative <level>,(@<chanlist>)

This command sets the negative current protection value for over-current protection, This command only applies to bidirectional DC power supply models.

Syntax

[SOURce:]CURRent:PROTection[:LEVel]:NEGative <level>,(@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]CURRent:PROTection[:LEVel]:NEGative?
[MINimum|MAXimum,](@<chanlist>)

*RST

MAX

Example

CURRent:PROTection:NEGative -2,(@1:2)

[SOURce:]CURRent:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>)

This command enables or disables the over-current protection (OCP) function. The enabled state is On (1); the disabled state is Off (0). If the over-current protection function is enabled and the output is disabled.

Syntax

[SOURce:]CURRent:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>)

Query syntax

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

Example

CURRent:PROTection:STATe ON, (@1)

[SOURce:]CURRent:PROTection:DELay[:TIME] <value>, (@<chanlist>)

This command sets the over-current protection delay. The over-current protection function will not be triggered on the specified output channel during the delay time. After the delay time has expired, the over-current protection function will be active.

Syntax

[SOURce:]CURRent:PROTection:DELay[:TIME] <value>, (@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

Query syntax

[SOURce:]CURRent:PROTection:DELay[:TIME]? [MIN|MAX,] (@<chanlist>)

Example

CURRent:PROTection:DELay 30, (@1)

[SOURce:]CURRent:UNDer:PROTection[:LEVel][:POSitive] <level>,(@<chanlist>)

This command sets and queries the positive current protection value for under-current protection.

Syntax

[SOURce:]CURRent:UNDer:PROTection[:LEVel][:POSitive] <level>,(@<chanlist>)

Query syntax

[SOURce:]CURRent:UNDer:PROTection[:LEVel][:POSitive] [MINimum|MAXimum,](@<chanlist>)

Example

VOLTage:UNDer:PROTection 10, (@1)

[SOURce:]CURRent:UNDer:PROTection[:LEVel]:NEGative <level>,(@<chanlist>)

This command sets and queries the negative current protection value for under-current protection.

Syntax

[SOURce:]CURRent:UNDer:PROTection[:LEVel]:NEGative <level>,(@<chanlist>)

Query syntax

[SOURce:]CURRent:UNDer:PROTection[:LEVel]:NEGative? [MINimum|MAXimum,](@<chanlist>)

Example

CURRent:UNDer:PROTection:NEGative 10, (@1)

[SOURce:]CURRent:UNDer:PROTection:STATe <0|OFF|1|ON>,(@<chanlist>)

This command enables or disables the under-current protection (UCP) function. The enabled state is On (1); the disabled state is Off (0). If the under-current protection function is enabled and the output is disabled.

Syntax

[SOURce:]CURRent:UNDer:PROTection:STATe 0|OFF|1|ON,(@<chanlist>)

Query syntax

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

Example

CURRent:UNDer:PROTection:STATe ON, (@1)

[SOURce:]CURRent:UNDer:PROTection:DELay[:TIME] <value>,(@<chanlist>)

This command sets the under-current protection delay. The under-current protection function will not be triggered on the specified output channel during the delay time. After the delay time has expired, the under-current protection function will be active.

Syntax

[SOURce:]CURRent:UNDer:PROTection:DELay[:TIME] <level>,(@<chanlist>)

Query syntax

[SOURce:]CURRent:UNDer:PROTection:DELay[:TIME]? [MIN|MAX,] (@<chanlist>)

Example

CURRent:UNDer:PROTection:DELay 10, (@1)

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

This command sets the under-current protection warm time. The under-voltage protection function will not be triggered on the specified output channel during the warm time. even if the voltage reaches the under-voltage protection point and meets the delay time.

Syntax

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

Arguments

MAXimum|MINimum|DEFault

Query syntax

[SOURce:]CURRent:UNDer:PROTection:WARM[:TIME]? [MIN|MAX,] (@<chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

CURRent:UNDer:PROTection:WARM 10, (@1)

[SOURce:]CURRent:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

This command sets the current rise slew rate.

If the setting state of the [SOURce:]CURRent:SLEW:COUPle is ON, this command sets the current rise slew rate while simultaneously setting the current fall slew rate.

Syntax

[SOURce:]CURRent:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

Arguments

0.001-30|MINimum|MAXimum (unit: A/ms)

Query syntax

[SOURce:]CURRent:SLEW[:POSitive][:IMMediate]? [MIN|MAX,] (@<chanlist>)

Example

CURRent:SLEW 10, (@1)

[SOURce:]CURRent:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>)

This command sets the current fall slew rate.

If the setting state of the [SOURce:]CURRent:SLEW:COUPle is ON, this command sets the current fall slew rate while simultaneously setting the current rise slew rate.

Syntax

[SOURce:]CURRent:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>)

Query syntax

[SOURce:]CURRent:SLEW:NEGative[:IMMediate]? [MIN|MAX,] (@<chanlist>)

Example

CURRent:SLEW:NEGative 10, (@1)

[SOURce:]CURRent:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

This command sets and queries the tracking status of the current slope.

Syntax

[SOURce:]CURRent:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]CURRent:SLEW:COUPle? (@<chanlist>)

Example

CURRent:SLEW:COUPle ON, (@1)

[SOURce:]CURRent:SLEW[:POSitive]:MAXimum 0|OFF|1|ON,(@<chanlist>)

Enable or disable current rise slope maximum.

Syntax

[SOURce:]CURRent:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

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

Example

CURRent:SLEW:MAXimum ON, (@1)

[SOURce:]CURRent:SLEW:NEGative:MAXimum 0|OFF|1|ON,(@<chanlist>)

Enable or disable current drop slope maximum.

Syntax

[SOURce:]CURRent:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

Copyright ©ITECH Electronic Co., Ltd.

Arguments

MAXimum|MINimum|DEFault

Query syntax

[SOURce:]CURRent:SLEW:NEGative:MAXimum? (@<chanlist>)

Example

CURRent:SLEW:NEGative:MAXimum ON, (@1)

[SOURce:]POWer:LIMit[:POSitive][:IMMediate][:AMPLitude] <value>, (<@chanlist>)

This command sets the positive power limit of the specified output channel. If [SOURce:]POWer:LIMit:COUPle is enabled, this command also sets the value of the negative power limit.

Syntax

[SOURce:]POWer:LIMit[:POSitive][:IMMediate][:AMPLitude] <value>, (<@chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]POWer:LIMit[:POSitive][:IMMediate][:AMPLitude]? [MINimum|MAXimum,] (<@chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

POW:LIM 75,(@1,2)

[SOURce:]POWer:LIMit:NEGative[:IMMediate][:AMPLitude]<value>, (<@chanlist>)

This command sets the negative power limit of the specified output channel. If [SOURce:]POWer:LIMit:COUPle is enabled, this command also sets the value of the positive power limit.

Syntax

[SOURce:]POWer:LIMit:NEGative[:IMMediate][:AMPLitude]<value>, (<@chanlist>)

Arguments

MAXimum|MINimum|DEFault

Query syntax

[SOURce:]POWer:LIMit:NEGative[:IMMediate][:AMPLitude]? [MINimum|MAXimum,] (<@chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

POWer:LIMit:NEGative 20, (@1)

[SOURce:]POWer:LIMit:COUPle 0|OFF|1|ON, (@<chanlist>)

This command sets and queries the tracking status of the power slope.

Syntax

[SOURce:]POWer:LIMit:COUPle 0|OFF|1|ON, (@<chanlist>)

Query syntax

[SOURce:]POWer:LIMit:COUPle? (@<chanlist>)

Arguments

0|OFF|1|ON

*RST

OFF

Example

POWer:LIMit:COUPle ON, (@1)

[SOURce:]POWer:PROTection:DELay[:TIME] <value>, (@<chanlist>)

The commands set and query the delay time of overpower protection.

Syntax

[SOURce:]POWer:PROTection:DELay[:TIME] <value>, (@<chanlist>)

Arguments

0-maximum|MINimum|MAXimum

Query syntax

[SOURce:]POWer:PROTection:DELay[:TIME]? [MINimum|MAXimum,] (@<chanlist>)

Example

POWer:PROTection:DELay 20, (@1)

[SOURce:]POWer:PROTection:STATe 0|OFF|1|ON, (@<chanlist>)

This command enables or disables the over-power protection (OPP) function. The enabled state is On (1); the disabled state is Off (0). If the over-power protection function is enabled and the output is disabled.

Syntax

[SOURce:]POWer:PROTection:STATe 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

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

Example

VOLTage:PROTection:DELay 30, (@1)

[SOURce:]POWer:PROTection[:LEVel][:POSitive] <value>,(@<chanlist>)

This command sets the positive power protection value for over-power protection.

Syntax

[SOURce:]POWer:PROTection[:LEVel][:POSitive] <value>,(@<chanlist>)

Query syntax

[SOURce:]POWer:PROTection[:LEVel][:POSitive]?
[MINimum|MAXimum,](@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

Example

POWer:PROTection 100, (@1)

[SOURce:]POWer:PROTection[:LEVel]:NEGative <value>,(@<chanlist>)

This command sets the negative power protection value for over-power protection, This command only applies to bidirectional DC power supply models.

Syntax

[SOURce:]POWer:PROTection[:LEVel]:NEGative <value>,(@<chanlist>)

Query syntax

[SOURce:]POWer:PORTection[:LEVel]:NEGative?
[MINimum|MAXimum,](@<chanlist>)

Copyright ©ITECH Electronic Co., Ltd.

Arguments

MAXimum|MINimum|DEFault

Example

POWer:PORTection:NEGative 10, (@1)

[SOURce:]POWer:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

This command sets the power rise slew rate.

If the setting state of the [SOURce:]POWer:SLEW:COUPle is ON, this command sets the power rise slew rate while simultaneously setting the power fall slew rate.

Syntax

[SOURce:]POWer:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]POWer:SLEW[:POSitive][:IMMediate]? [MIN|MAX,] (@<chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

POWer:SLEW 2 (@1)

[SOURce:]POWer:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>)

This command sets the power fall slew rate.

If the setting state of the [SOURce:]POWer:SLEW:COUPle is ON, this command sets the power fall slew rate while simultaneously setting the power rise slew rate.

Syntax

[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

Query syntax

[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate]? [MIN|MAX,] (@<chanlist>)

Example

VOLTage:SLEW 10, (@1)

[SOURce:]POWer:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

This command sets and queries the tracking status of the power slope.

Syntax

[SOURce:]POWer:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]POWer:SLEW:COUPle? (@<chanlist>)

Example

POWer:SLEW:COUPle ON, (@1)

[SOURce:]POWer:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>)

This command set and query the status of the maximum slope of the power rise slope.

Syntax

[SOURce:]POWer:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]POWer:SLEW[:POSitive]:MAXimum? (@<chanlist>)

Example

POWer:SLEW:MAXimum ON, (@1)

[SOURce:]POWer:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

This command sets and queries the status of the maximum slope of the power down slope.

Syntax

[SOURce:]POWer:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]POWer:SLEW:NEGative:MAXimum? (@<chanlist>)

Example

POWer:SLEW:NEGative:MAXimum ON, (@1)

[SOURce:]SINK:RESistance[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>)

This command sets and queries the level of the SINK resistor in CC mode.

Syntax

[SOURce:]SINK:RESistance[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

Query syntax

[SOURce:]SINK:RESistance[:LEVel][:IMMediate][:AMPLitude]? [MIN|MAX,] (@<chanlist>)

Example

SINK:RESistance 50, (@1)

[SOURce:]SINK:RESistance:STATe 0|OFF|1|ON,(@<chanlist>)

This command sets and queries the status of the SINK resistor in CC mode.

Syntax

[SOURce:]SINK:RESistance:STATe 0|OFF|1|ON,(@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]SINK:RESistance:STATe? (@<chanlist>)

Example

SINK:RESistance:STATe ON, (@1)

5.5 OUTPut Subsystem

INPut|OUTPut[:STATe] 0|OFF|1|ON , (@<chanlist>)

This command is used to control the output state of the instrument.

Syntax

INPut|OUTPut[:STATe] 0|OFF|1|ON , (@<chanlist>)

Arguments

0|OFF|1|ON

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut[:STATe]? (@<chanlist>)

Return Arguments

0|1

Example

OUTPut ON , (@1)

INPut|OUTPut[:STATe]:COUPle:CHANnel ALL|NONE|[<channel>{,<channel>}]

Sets and queries the channels for output coupling.

Syntax

INPut|OUTPut[:STATe]:COUPle:CHANnel ALL|NONE|[<channel> {,<channel>}]

Arguments

ALL|NONE|<1,2,3,4,5,6,7,8>

Query syntax

INPut|OUTPut[:STATe]:COUPle:CHANnel?

Example

OUTP:COUP:CHAN 1,2,4

INPut|OUTPut[:STATe]:COUPle:DOFFset <value>

This instruction is used to set and query the delay offset.

Syntax

INPut|OUTPut[:STATe]:COUPle:DOFFset <value>

Arguments

MIN|MAX|

Query syntax

INPut|OUTPut[:STATe]:COUPle:DOFFset? [MIN|MAX]

Return Arguments

MIN|MAX

Example

OUTPut:COUPle:DOFFset 10

INPut|OUTPut[:STATe]:COUPle:DOFFset:MODE AUTO|MANual

Set software watchdog timeout, unit is s.

Syntax

INPut|OUTPut[:STATe]:COUPle:DOFFset:MODE AUTO|MANual

Arguments

AUTO|MANual

Query syntax

INPut|OUTPut[:STATe]:COUPle:DOFFset:MODE?

Return Arguments

AUTO|MANual

Example

OUTPut:COUPle:DOFF:MODE AUTO

INPut|OUTPut[:STATe]:COUPle:MAX:DOFFset?

Sets the maximum delay offset for querying the machine.

Syntax

INPut|OUTPut[:STATe]:COUPle:MAX:DOFFset?

Return Arguments

NRf

Example

OUTPut:COUPle:MAX:DOFFset?

INPut|OUTPut[:STATe]:DELay:FALL <value>, (@<chanlist>)

Sets the delay time for output shutdown.

Syntax

INPut|OUTPut[:STATe]:DELay:FALL <value>, (@<chanlist>)

Arguments

<value>MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut[:STATe]:DELay:FALL? [MIN|MAX,] (@<chanlist>)

Return Arguments

MINimum|MAXimum

Example

OUTPut:DELay:FALL 10, (@1)

INPut|OUTPut[:STATe]:DELay:RISE <value>, (@<chanlist>)

Sets the delay time for the output to turn on.

Syntax

INPut|OUTPut[:STATe]:DELay:RISE <value>, (@<chanlist>)

Arguments

<value>MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut[:STATe]:DELay:RISE? [MIN|MAX,] (@<chanlist>)

Return Arguments

MINimum|MAXimum

Example

OUTPut:DELay:RISE 10, (@1)

INPut|OUTPut:INHibit:COUPle:CHANnel ALL|NONE|[<channel>{,<channel>}]

Sets and queries the channels for output suppression coupling.

Syntax

INPut|OUTPut:INHibit:COUPle:CHANnel ALL|NONE|[<channel> {,<channel>}]

Arguments

ALL|NONE|[<channel> {,<channel>}]

Query syntax

INPut|OUTPut:INHibit:COUPle:CHANnel?

Example

OUTPut:INHibit:COUPle:CHANnel ALL

INPut|OUTPut:INHibit:MODE LATChing|LIVE|OFF

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

Syntax

INPut|OUTPut:INHibit:MODE LATChing|LIVE|OFF

Arguments

LATChing|LIVE|OFF

Query syntax

INPut|OUTPut:INHibit:MODE?

Return Arguments

LATChing|LIVE|OFF

Example

OUTPut:INHibit:MODE LATChing

INPut|OUTPut:PON:STATe RST|LAST|LOFF

Setting and querying the status of the power-up. RST means restoring the factory value, LAST means the status before the last power-off, and LOFF means LAST+OFF.

Syntax

INPut|OUTPut:PON:STATe RST|LAST|LOFF

Arguments

RST|LAST|LOFF

Query syntax

INPut|OUTPut:PON:STATe?

Example

OUTPut:PON:STATe RST

INPut|OUTPut:PROTection:CLEar (@<chanlist>)

This command is used to clear the protection status.

Syntax

INPut|OUTPut:PROTection:CLEar (@<chanlist>)

Example

OUTPut:PROT:CLE (@1)

INPut|OUTPut:PROTection:COUPle 0|OFF|1|ON

This command is used to set enable or disable protection coupling.

Syntax

INPut|OUTPut:PROTection:COUPle 0|OFF|1|ON

Arguments

0|OFF|1|ON

Query syntax

INPut|OUTPut:PROTection:COUPle?

Return Arguments

0|OFF|1|ON

Example

OUTP:PROT:COUPle ON

INPut|OUTPut:PROTection:COUPle:CHANnel ALL|NONE|[<channel>{,<channel>}]

The commands set and query the channel for protection coupling.

Syntax

INPut|OUTPut:PROTection:COUPle:CHANnel ALL|NONE|[<channel>{,<channel>}]

Arguments

ALL|NONE|[<channel> {,<channel>}]

Query syntax

INPut|OUTPut:PROTection:COUPle:CHANnel?

Return Arguments

ALL|NONE|[<channel> {,<channel>}]

Example

OUTP:PROT:COUP:CHAN 1,2,3

INPut|OUTPut:PROTection:WDOG[:STATe] 0|OFF|1|ON

The commands set and query the enable state of the software watchdog.

Syntax

INPut|OUTPut:PROTection:WDOG[:STATe] 0|OFF|1|ON

Arguments

0|OFF|1|ON

Query syntax

INPut|OUTPut:PROTection:WDOG[:STATe]?

Return Arguments

0|1

Example

OUTP:PROT:WDOG ON

INPut|OUTPut:PROTection:WDOG:DELay <value>

This command sets the delay time of the software watchdog.

Syntax

INPut|OUTPut:PROTection:WDOG:DELay <value>

Arguments

[MIN|MAX]

Query syntax

INPut|OUTPut:PROTection:WDOG:DELay? [MIN|MAX]

Return Arguments

[MIN|MAX]

Example

INPut|OUTPut:PROTection:WDOG:DELay 10

INPut|OUTPut:PROTection:FOLDback[:MODE] <OFF|CC|CV>,(@<chanlist>)

The commands set and query the mode of foldback protection.

Syntax

INPut|OUTPut:PROTection:FOLDback[:MODE] <OFF|CC|CV>,(@<chanlist>)

Arguments

OFF|CC|CV

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut:PROTection:FOLDback[:MODE]? (@<chanlist>)

Return Arguments

OFF|CC|CV

Example

OUTPut:PROTection:FOLDback CC,(@1)

INPut|OUTPut:PROTection:FOLDback:DELay <value>,(@<chanlist>)

The commands set and query the foldback protection delay time.

Syntax

INPut|OUTPut:PROTection:FOLDback:DELay <value>,(@<chanlist>)

Arguments

<value>MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut:PROTection:FOLDback:DELay? (@<chanlist>)

Return Arguments

[MIN|MAX]

Example

OUTPut:PROTection:FOLDback:DELay 10,(@1)

INPut|OUTPut:RELay:LOCK[:STATe] 0|1|OFF|ON,(@<chanlist>)

This command enables or disables the latched-off state of the output relay.

Syntax

INPut|OUTPut:RELay:LOCK[:STATe] 0|1|OFF|ON,(@<chanlist>)

Arguments

0|1|OFF|ON

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Copyright ©ITECH Electronic Co., Ltd.

Query syntax

INPut|OUTPut:RELay:LOCK[:STATe]? (@<chanlist>)

Return Arguments

0|1|OFF|ON

Example

OUTPut:RELay:LOCK ON,(@1)

INPut|OUTPut:REGulation:SPEed <LOW|HIGH1|HIGH2|HIGH3> (@<chanlist>)

The commands set and query the regulation speed.

Syntax

INPut|OUTPut:REGulation:SPEed <LOW|HIGH1|HIGH2|HIGH3> (@<chanlist>)

Arguments

LOW|HIGH1|HIGH2|HIGH3

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut:REGulation:SPEed? <LOW|HIGH1|HIGH2|HIGH3> (@<chanlist>)

Return Arguments

LOW|HIGH1|HIGH2|HIGH3

Example

OUTPut:REGulation:SPEed LOW (@1)

INPut|OUTPut:FUNCtion:MODE <FIXed|ARB|BCHarge|BEMulation|TRANsient|OCP|OPP|SEQuence|CDARB|SSWeep|LIST|BDIScharge>,(@<chanlist>)

This command sets the function mode of the inputs and outputs.

Syntax

INPut|OUTPut:FUNCtion:MODE <FIXed|ARB|BCHarge|BEMulation|TRANsient|OCP|OPP|SEQuence|CDARB|SSWeep|LIST|BDIScharge>,(@<chanlist>)

Arguments

<FIXed|ARB|BCHarge|BEMulation|TRANsient|OCP|OPP|SEQuence|CDARB|SSWeep|LIST|BDIScharge>

<chanlist>: Channel number.

Copyright ©ITECH Electronic Co., Ltd.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut:FUNCtion:MODE? (@<chanlist>)

Example

OUTPut:FUNCtion:MODE FIXed,(@1)

INPut|OUTPut:OFF:VOLTage:RTZero <0||OFF|1|ON>,(@<chanlist>)

The commands set the output and query the state of zeroing the voltage of OFF port.

Syntax

INPut|OUTPut:OFF:VOLTage:RTZero <0||OFF|1|ON>,(@<chanlist>)

Arguments

<0||OFF|1|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut:OFF:VOLTage:RTZero? (@<chanlist>)

Example

OUTPut:OFF:VOLTage:RTZero ON,(@1)

5.6 FETCh & MEASureSubsystem

FETCh[:SCALar]:CURRent[:DC]? (@<chanlist>)

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

This command is used to read the average value of current.

Syntax

FETCh[:SCALar]:CURRent? (@<chanlist>)

MEASure[:SCALar]:CURRent[:AC]? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETC:CURR? (@1)

MEAS:CURR? (@1)

FETCh[:SCALar]:CURRent:ACDC? (@<chanlist>)

MEASure[:SCALar]:CURRent:ACDC? (@<chanlist>)

This command is used to read the current RMS value.

Syntax

FETCh[:SCALar]:CURRent:ACDC? (@<chanlist>)

MEASure[:SCALar]:CURRent:ACDC? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>…

Example

FETC:CURR:ACDC? (@1)

MEAS:CURR:ACDC? (@1)

FETCh[:SCALar]:CURRent:MAXimum? (@<chanlist>)

MEASure[:SCALar]:CURRent:MAXimum? (@<chanlist>)

This command is used to read the current maximum value.

Syntax

FETCh[:SCALar]:CURRent:MAXimum? (@<chanlist>)

MEASure[:SCALar]:CURRent:MAXimum? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:CURRent:MAX? (@1)

MEASure:CURRent:MAX? (@1)

FETCh[:SCALar]:CURRent:MINimum? (@<chanlist>)

MEASure[:SCALar]:CURRent:MINimum? (@<chanlist>)

This command is used to read the current min.

Syntax

FETCh[:SCALar]:CURRent:MINimum? (@<chanlist>)

MEASure[:SCALar]:CURRent:MINimum? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:CURR:MIN? (@1)

MEAS:CURRent:MIN? (@1)

FETCh[:SCALar]:POWer[:DC]? (@<chanlist>)

MEASure[:SCALar]:POWer[:DC]? (@<chanlist>)

This command is used to read the power average.

Syntax

FETCh[:SCALar]:POWer[:DC]? (@<chanlist>)

MEASure[:SCALar]:POWer[:DC]? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:POWer? (@1)

MEASure:POWer? (@1)

FETCh[:SCALar]:POWer:MAXimum? (@<chanlist>)

MEASure[:SCALar]:POWer:MAXimum? (@<chanlist>)

This command is used to read the maximum value of the power value.

Syntax

FETCh[:SCALar]:POWer:MAXimum? (@<chanlist>)

MEASure[:SCALar]:POWer:MAXimum? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>…

Example

FETCh:POWer:MAX? (@1)

MEASure:POWer:MAX? (@1)

FETCh[:SCALar]:POWer:MINimum? (@<chanlist>)

MEASure[:SCALar]:POWer:MINimum? (@<chanlist>)

This command is used to read the power min.

Syntax

FETCh[:SCALar]:POWer:MINimum? (@<chanlist>)

MEASure[:SCALar]:POWer:MINimum? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:POWer:MIN? (@1)

MEAS:POWer:MIN? (@1)

FETCh[:SCALar]:VOLTage[:DC]? (@<chanlist>)

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

This command is used to read the average value of voltage.

Syntax

FETCh[:SCALar]:VOLTage[:DC]? (@<chanlist>)

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

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>…

Example

FETC:VOLT? (@1)

MEAS:VOLT? (@1)

FETCh[:SCALar]:VOLTage:ACDC? (@<chanlist>)

MEASure[:SCALar]:VOLTage:ACDC? (@<chanlist>)

This command is used to read the voltage RMS value.

Syntax

FETCh[:SCALar]:VOLTage:ACDC? (@<chanlist>)

MEASure[:SCALar]:VOLTage:ACDC? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETC:VOLT:ACDC? (@1)

MEAS:VOLT:ACDC? (@1)

FETCh[:SCALar]:VOLTage:MINimum? (@<chanlist>)

MEASure[:SCALar]:VOLTage:MINimum? (@<chanlist>)

This command is used to read the voltage min.

Syntax

FETCh[:SCALar]:VOLTage:MINimum? (@<chanlist>)

MEASure[:SCALar]:VOLTage:MINimum? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETC:VOLT:MIN? (@1)

MEAS:VOLT:MIN? (@1)

FETCh[:SCALar]:VOLTage:MAXimum? (@<chanlist>)

MEASure[:SCALar]:VOLTage:MAXimum? (@<chanlist>)

This command is used to read the maximum value of voltage.

Syntax

FETCh[:SCALar]:VOLTage:MAXimum? (@<chanlist>)

MEASure[:SCALar]:VOLTage:MAXimum? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Arguments

<NRf>

Example

FETC:VOLTage:MAX? (@1)

MEAS:VOLTage:MAX? (@1)

FETCh[:SCALar]:AHOur? (@<chanlist>)

MEASure[:SCALar]:AHOur? (@<chanlist>)

This command is used to read the channel's corresponding amp-hours data.

Syntax

FETCh[:SCALar]:AHOur? (@<chanlist>)

MEASure[:SCALar]:AHOur? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:AHOur? (@1)

FETCh[:SCALar]:WHOur? (@<chanlist>)

MEASure[:SCALar]:WHOur? (@<chanlist>)

This command is used to read the watt-hour data of the channel.

Syntax

FETCh[:SCALar]:WHOur? (@<chanlist>)

MEASure[:SCALar]:WHOur? (@<chanlist>)

Return Arguments

<NRf>

Example

FETC:WHOur? (@1)

MEAS:WHOur? (@1)

FETCh[:SCALar]:TIME? (@<chanlist>)

MEASure[:SCALar]:TIME? (@<chanlist>)

This command is used to read the load time.

Syntax

FETCh[:SCALar]:TIME? (@<chanlist>)

MEASure[:SCALar]:TIME? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETC:TIME? (@1)

MEAS:TIME? (@1)

FTECh[:SCALar][:ALL]? (@<chanlist>)

MEASure[:SCALar][:ALL]? (@<chanlist>)

This command is used to read all measurement data at once.

(CURR_DC,VOLT_DC,POW_DC,CURR_ACDC,VOLT_ACDC,CURR_MAX,VOLT_MAX,POW_MAX,CURR_MIN,VOLT_MIN,POW_MIN,AH,WH,TIME)

Syntax

FTECh[:SCALar][:ALL]? (@<chanlist>)

MEASure[:SCALar][:ALL]? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FTEC? (@1)

MEAS? (@1)

Copyright ©ITECH Electronic Co., Ltd.

FETCh:ARRay:CURRent[:DC]? (@<chanlist>)

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

This query returns an array containing the digitized output current in amperes. The data returned by the FETCh command is the result of the last measurement command or triggered acquisition.

Syntax

FETCh:ARRay:CURRent[:DC]? (@<chanlist>)

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

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Return Arguments

<NRf>

Example

FETCh:ARRay:CURRent? (@1)

MEASure:ARRay:CURRent? (@1)

FETCh:ARRay:VOLTage[:DC]? (@<chanlist>)

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

This query returns an array containing the digitized output voltage in volts. The data returned by the FETCh command is the result of the last measurement command or triggered acquisition.

Syntax

FETCh:ARRay:VOLTage[:DC]? (@<chanlist>)

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

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:ARRay:VOLTage? (@1)

MEASure:ARRay:VOLTage? (@1)

FETCh:ARRay:POWer[:DC]? (@<chanlist>)

MEASure:ARRay:POWer[:DC]? (@<chanlist>)

This query returns an array containing the digitized output power in watts. The data returned by the FETCh command is the result of the last measurement command or triggered acquisition.

Syntax

FETCh:ARRay:POWer[:DC]? (@<chanlist>)

MEASure:ARRay:POWer[:DC]? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:ARRay:POWer? (@1)

MEASure:ARRay:POWer? (@1)

5.7 SENSe Subsystem

SENSe[:REMote][:STATe] 0|1|OFF|ON, (@<channlist>)

Set and query the status of remote sensing.

Syntax

FETCh:ARRay:POWer[:DC]? (@<chanlist>)

MEASure:ARRay:POWer[:DC]? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Query syntax

SENSe[:REMote][:STATe]? (@<channlist>)

Copyright ©ITECH Electronic Co., Ltd.

Example

SENS ON,(@1)

SENSe:SWEep:OFFSet:POINts <value>, (@<chanlist>)

This command defines the offset in a data sweep when the measurement is triggered. Programmed values can range from –(SENS:SWE:POIN setting –1) through 2,000,000,000 (2E9). Positive values represent the delay after the trigger occurs but before the samples are acquired. Negative values represent data samples taken prior to the trigger. The maximum negative value that can be programmed is one less than the maximum allowable SENSe:SWEep:POINts value.

Syntax

SENSe:SWEep:OFFSet:POINts <value>, (@<chanlist>)

Arguments

<Value>[MINimum|MAXimum]

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Query syntax

SENSe:SWEep:OFFSet:POINts? [MINimum|MAXimum,] (@<chanlist>)

Example

SENS:SWE:OFFS:POIN -1024,(@1)

SENSe:SWEep:POINts <value>, (@<chanlist>)

This command defines the number of points in a measurement. The number of samples (points) that can be specified depends on the number of measurement parameters selected. You can measure up to eight parameters (voltage and current X 4 outputs).

Syntax

SENSe:SWEep:POINts <value>, (@<chanlist>)

Arguments

<Value>1 to 16384|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

[MINimum|MAXimum]

Query syntax

SENSe:SWEep:POINts? [MINimum|MAXimum,] (@<chanlist>)

Example

SENS:SWE:POIN 1024,(@1)

SENSe:SWEep:TINTerval <value>, (@<chanlist>)

This command defines the time period between samples in seconds. Programmed values can range from 5 us to 2100 seconds. Note that the shortest time interval (fastest speed) that can be specified depends on the number of parameters that are being measured and the model that is doing the measuring.

Syntax

SENSe:SWEep:TINTerval <value>, (@<chanlist>)

Arguments

<Value>0.000005-2100.0|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

[MINimum|MAXimum]

Query syntax

SENSe:SWEep:TINTerval? [MINimum|MAXimum,] (@<chanlist>)

Example

SENS:SWE:TINT 0.002,(@1)

SENSe:NPLCycles? [MINimum|MAXimum,](@<chanlist>)

This command queries the number of measured power frequency cycles.

Syntax

SENSe:NPLCycles? [MINimum|MAXimum,](@<chanlist>)

Arguments

<Value>0.000005-2100.0|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Copyright ©ITECH Electronic Co., Ltd.

Return Arguments

10 (50HZ)

12 (60HZ)

Example

SENS:NPLC? (@1)

SENSe:APERture <time|MINimum|MAXimum>,(@<chanlist>)

The commands set and query the measured aperture time.

Syntax

SENSe:APERture <time|MINimum|MAXimum>,(@<chanlist>)

Arguments

<Value>0.02-34406400|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

[MINimum|MAXimum]

Query syntax

SENSe:APERture? [MINimum|MAXimum,] (@<chanlist>)

Example

SENS:APER 0.1,(@1)

SENSe:MEASure:TIME:MODE <NPLC|APERture>,(@<chanlist>)

Set and query the time mode of the measurement.

Syntax

SENSe:MEASure:TIME:MODE <NPLC|APERture>,(@<chanlist>)

Arguments

<NPLC|APERture>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NPLC|APERture>

Query syntax

SENSe:MEASure:TIME:MODE?(@<chanlist>)

Example

SENS:MEAS:TIME:MODE NPLC, (@1)

SENSe:AHOur:CLEar (@<chanlist>)

This instruction is used to clear the amp-hours data.

Syntax

SENSe:AHOur:CLEar (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Example

SENS:AHO:CLE (@1)

SENSe:WHOur:CLEar (@<chanlist>)

This instruction is used to clear the watt-hour data.

Syntax

SENSe:AHOur:CLEar (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Example

SENS:AHO:CLE (@1)

SENSe:AHOur:CLEar:AUTO <0|1|OFF|ON>,(@<chanlist>)

This instruction is used to set up the automatic clearing of the amp-hours data.

Syntax

SENSe:AHOur:CLEar (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1

(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:AHOur:CLEar:AUTO? (@<chanlist>)

Example

SENS:AHO:CLE:AUTO 1,(@1)

SENSe:WHOur:CLEar:AUTO <0|1|OFF|ON>,(@<chanlist>)

This instruction is used to set up the automatic clearing of the watt-hour data.

Syntax

SENSe:WHOur:CLEar:AUTO <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:WHOur:CLEar:AUTO? (@<chanlist>)

Example

SENS:WHO:CLE:AUTO ON,(@1)

SENSe:DLOG:FUNCtion:CURRent 0|OFF|1|ON, (@<chanlist>)

This command is used to set and query the current measurement status of the internal recorder.

Syntax

SENSe:DLOG:FUNCtion:CURRent 0|OFF|1|ON, (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:DLOG:FUNCtion:CURRent? (@<chanlist>)

Example

SENSe:DLOG:FUNCtion:CURRent ON,(@1)

SENSe:DLOG:FUNCtion:VOLTage 0|OFF|1|ON, (@<chanlist>)

This command is used to set and query the voltage measurement status of the internal recorder.

Syntax

SENSe:DLOG:FUNCtion:VOLTage 0|OFF|1|ON, (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:DLOG:FUNCtion:VOLTage? (@<chanlist>)

Example

SENSe:DLOG:FUNCtion:CURRent ON,(@1)

SENSe:DLOG:FUNCtion:MINMax 0|OFF|1|ON

This command is used to set and query the maximum and minimum value measurement function of the internal recorder.

Syntax

SENSe:DLOG:FUNCtion:MINMax 0|OFF|1|ON

Arguments

<0|1|OFF|ON>

Query syntax

SENSe:DLOG:FUNCtion:MINMax?

Example

SENSe:DLOG:FUNCtion:MINMax ON

SENSe:DLOG:MARKer1:POINt <value>

This command is used to set and query the start position and end position of data analysis.

Syntax

SENSe:DLOG:MARKer1:POINt <value>

Arguments

<value|MINimum|MAXimum> (unit is s)

MINimum0s

MAXimum: setting value

Query syntax

SENSe:DLOG:MARKer1:POINt?

Example

SENSe:DLOG:MARKer1:POINt 20

SENSe:DLOG:MARKer2:POINt <value>

This command is used to set and query the start position and end position of data analysis.

Syntax

SENSe:DLOG:MARKer2:POINt <value>

Arguments

<value|MINimum|MAXimum> (unit is s)

MINimum: 0s

MAXimum: setting value

Query syntax

SENSe:DLOG:MARKer2:POINt?

Example

SENSe:DLOG:MARKer2:POINt 20

SENSe:DLOG:OFFSet <value>

This command is used to set and query the trigger offset time percentage of the data logger.

Syntax

SENSe:DLOG:OFFSet <value>

Arguments

<0.00-100.00|MINimum|MAXimum> (unit is %)

MINimum0.00

MAXimum100.00

Query syntax

SENSe:DLOG:OFFSet?

Example

SENSe:DLOG:OFFSet 20

SENSe:DLOG:PERiod <value>

This command is used to set and query the integration time of the data logger.

Syntax

SENSe:DLOG:PERiod <value>

Arguments

<value|MINimum|MAXimum> (unit is s)

MINimum0.00002s

MAXimum60s

Query syntax

SENSe:DLOG:PERiod? [MINimum|MAXimum]

Example

SENSe:DLOG:PERiod 5

SENSe:DLOG:TIME <value>

This command is used to set and query the time of the data logger.

Syntax

SENSe:DLOG:TIME <value>

Arguments

<value|MINimum|MAXimum> (unit is s)

MINimum0s

MAXimum86400s

Query syntax

SENSe:DLOG:TIME? [MINimum|MAXimum]

Example

SENSe:DLOG:TIME 50

SENSe:DLOG:TINTerval <value>

This command is used to set and query the time interval of the data logger.

Syntax

SENSe:DLOG:TINTerval <value>

Arguments

<value|MINimum|MAXimum> (unit is s)

MINimum0.00002s

MAXimum60s

Query syntax

SENSe:DLOG:TINTerval? [MINimum|MAXimum]

Copyright ©ITECH Electronic Co., Ltd.

Example

SENSe:DLOG:TINTerval 10

SENSe:ELOG:FUNCtion:CURRent 0|OFF|1|ON, (@<chanlist>)

This command is used to set and query the current measurement status of the internal recorder.

Syntax

SENSe:ELOG:FUNCtion:CURRent 0|OFF|1|ON, (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:ELOG:FUNCtion:CURRent? (@<chanlist>)

Example

SENSe:ELOG:FUNCtion:CURRent ON,(@1)

SENSe:ELOG:FUNCtion:VOLTage 0|OFF|1|ON, (@<chanlist>)

This command is used to set and query the voltage measurement status of the internal recorder.

Syntax

SENSe:ELOG:FUNCtion:VOLTage 0|OFF|1|ON, (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:ELOG:FUNCtion:VOLTage? (@<chanlist>)

Example

SENSe:ELOG:FUNCtion:CURRent ON,(@1)

SENSe:ELOG:FUNCtion:CURRent:MINMax 0|OFF|1|ON, (@<chanlist>)

This command is used to set and query the current maximum and minimum measurement function of the internal recorder.

Syntax

SENSe:ELOG:FUNCtion:CURRent:MINMax 0|OFF|1|ON, (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:ELOG:FUNCtion:CURRent:MINMax? (@<chanlist>)

Example

SENSe:ELOG:CURRent:FUNCtion:MINMax ON,(@1)

SENSe:ELOG:FUNCtion:VOLTage:MINMax 0|OFF|1|ON, (@<chanlist>)

This command is used to set and query the current maximum and minimum measurement function of the internal recorder.

Syntax

SENSe:ELOG:FUNCtion:VOLTage:MINMax 0|OFF|1|ON, (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:ELOG:FUNCtion:VOLTage:MINMax? (@<chanlist>)

Example

SENSe:ELOG:CURRent:FUNCtion:MINMax ON,(@1)

SENSe:ELOG:PERiod <value>, (@<chanlist>)

This command is used to set and query the integration time of the data logger.

Syntax

SENSe:ELOG:PERiod <value>, (@<chanlist>)

Arguments

<value|MINimum|MAXimum>(unit is s)

MINimum0.0001s

MAXimum1s

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:ELOG:PERiod? [MINimum|MAXimum,] (@<chanlist>)

Example

SENSe:ELOG:PERiod 0.005,(@1)

5.8 List Subsystem

[SOURce:]LIST:COUNt <value>, (@<chanlist>)

This command is used to set and query the number of cycles that the list runs.

Syntax

[SOURce:]LIST:COUNt <value>, (@<chanlist>)

Arguments

0-65535|MINimum|MAXimum

Query syntax

[SOURce:]LIST:COUNt? (@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:COUN 10,(@1)

[SOURce:]LIST:CURRent:STEP[:LEVel] <value>,{<step>,(@<chanlist>)

The commands set and query the current value of the specified step of LIST. (Unit is A)

Syntax

[SOURce:]LIST:CURRent:STEP[:LEVel] <value>,{<step>,(@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:CURRent:STEP[:LEVel]?
[MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:CURRent 1.5,1,(@1)

[SOURce:]LIST:CURRent:SLEW <value>,<step>, (@<chanlist>)

The commands set and query the current slope of the specified step of LIST. (Unit is A)

Syntax

[SOURce:]LIST:CURRent:SLEW <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:CURRent:SLEW? [MINimum|MAXimum,]<step>(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:CURRent:SLEW 2,1,(@1)

[SOURce:]LIST:CURRent:POINts <value>,(@<chanlist>)

The commands set and query the number of current list points.

Syntax

[SOURce:]LIST:CURRent:POINts <value>,(@<chanlist>)

Arguments

1-2000|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:CURRent:POINts? (@<chanlist>)

Example

LIST:CURRent:POINts 1,(@1)

[SOURce:]LIST:VOLTage:STEP[:LEVel] <value>,<step>, (@<chanlist>)

The commands set and query the voltage value of the specified step of LIST. (Unit is V)

Syntax

[SOURce:]LIST:VOLTage:STEP[:LEVel] <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:VOLTage:STEP[:LEVel]?
[MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:VOLTage 10,1,(@1)

[SOURce:]LIST:VOLTage:SLEW <value>,<step>, (@<chanlist>)

The commands set and query the voltage slope of the specified step of LIST. (Unit is V)

Syntax

[SOURce:]LIST:VOLTage:SLEW <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:VOLTage:SLEW? [MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:VOLTage:SLEW? 1,(@1)

[SOURce:]LIST:VOLTage:POINts <value>,(@<chanlist>)

The commands set and query the number of voltage list points.

Syntax

[SOURce:]LIST:VOLTage:POINts <value>,(@<chanlist>)

Arguments

1-2000|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:VOLTage:POINts? (@<chanlist>)

Example

LIST:VOLTage:POINts 1,(@1)

[SOURce:]LIST:POWer:STEP[:LEVel] <value>,<step>, (@<chanlist>)

The commands set and query the power value of the specified step of LIST. (Unit is W)

Syntax

[SOURce:]LIST:POWer:STEP[:LEVel] <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:POWer:STEP[:LEVel]?
[MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:POWer 100,1,(@1)

[SOURce:]LIST:POWer:SLEW <value>,<step>, (@<chanlist>)

The commands set and query the power slope of the specified step of LIST. (Unit is W)

Syntax

[SOURce:]LIST:POWer:SLEW <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:POWer:SLEW? [MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:POWer:SLEW 100,1,(@1)

[SOURce:]LIST:POWer:POINts <value>,(@<chanlist>)

The commands set and query the number of current list points.

Syntax

[SOURce:]LIST:POWer:POINts <value>,(@<chanlist>)

Arguments

1-2000|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:POWer:POINts? (@<chanlist>)

Example

LIST:POWer:POINts 1,(@1)

[SOURce:]LIST:RESistance:STEP[:LEVel] <value>,<step>, (@<chanlist>)

The commands set and query the resistance value of the specified step of LIST. (Unit is Ω)

Syntax

[SOURce:]LIST:RESistance:STEP[:LEVel] <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:RESistance:STEP[:LEVel]?
[MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:RESistance 10,1,(@1)

[SOURce:]LIST:RESistance:SLEW <value>,<step>, (@<chanlist>)

The commands set and query the resistance value of the specified step of LIST. (Unit is Ω)

Syntax

[SOURce:]LIST:RESistance:SLEW <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:RESistance:SLEW? [MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:RESistance:SLEW? 1,(@1)

[SOURce:]LIST:RESistance:POINts <value>,(@<chanlist>)

The commands set and query the number of current list points.

Syntax

[SOURce:]LIST:RESistance:POINts <value>,(@<chanlist>)

Arguments

1-2000|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:RESistance:POINts? (@<chanlist>)

Example

LIST:RESistance:POINts 1,(@1)

[SOURce:]LIST:TERMinate:MODE <NORMal|LAST|OFF>,(@<chanlist>)

This command is used to set and query the way to end the list operation. There are three ways to end:

NORMal: after the list runs, it will automatically return to normal.

LAST: keep the output of the last step after the list runs.

OFF: turn off the output after the list operation is over and still in the list mode.

Syntax

[SOURce:]LIST:TERMinate:MODE <NORMal|LAST|OFF>,(@<chanlist>)

Arguments

NORMal|LAST|OFF

Query Syntax

[SOURce:]LIST:TERMinate:MODE? (@<chanlist>)

Returns

NORMal|LAST|OFF

*RST

OFF

Example

LIST:TERM NORM,(@1)

[SOURce:]LIST:DWELl <value>,<step>, (@<chanlist>)

This command is used to set and query the dwell time of LIST for the specified number of steps.

Syntax

[SOURce:]LIST:DWELl <value>,<step>, (@<chanlist>)

Arguments

0.001-10737|MINimum|MAXimum

Query syntax

[SOURce:]LIST:DWELl? [MINimum|MAXimum,],<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:DWELl 2,1,(@1)

[SOURce:]LIST:TOUTput:BOSTep:POINts? (@<chanlist>)

This query returns the number of points (steps) in the beginning of the step trigger list (BOSTep), not the point values.

Syntax

[SOURce:]LIST:TOUTput:BOSTep:POINts? (@<chanlist>)

Example

LIST:TOUTput:BOSTep:POINts? (@1)

[SOURce:]LIST:TOUTput:EOSTep:POINts? (@<chanlist>)

This query returns the number of points (steps) in the end of the step trigger list (EOSTep), not the point values.

Syntax

[SOURce:]LIST:TOUTput:EOSTep:POINts? (@<chanlist>)

Example

LIST:TOUTput:EOSTep:POINts? (@1)

[SOURce:]LIST:STEP ONCE|AUTO,<step>(@<chanlist>)

This command is used to Set and query step jump method, ONCE need to wait for the trigger, AUTO automatic jump.

Syntax

[SOURce:]LIST:STEP ONCE|AUTO,<step>(@<chanlist>)

Arguments

ONCE|AUTO

Query Syntax

[SOURce:]LIST:STEP? <step>,(@<chanlist>)

Example

LIST:STEP AUTO,1,(@1)

[SOURce:]LIST:TOUTput:BOSTep[:DATA] 0|OFF|1|ON,<step>,(@<chanlist>)

This command specifies which list steps generate a trigger signal at the beginning of the step (BOSTep).

A trigger is only generated when the state is set to ON. The trigger signal can be used as a trigger source for measurements and transients of other channels, and for digital port pins configured as trigger outputs.

Syntax

[SOURce:]LIST:TOUTput:BOSTep[:DATA] 0|OFF|1|ON,<step>,(@<chanlist>)

Query Syntax

[SOURce:]LIST:TOUTput:BOSTep[:DATA]? <step>,(@<chanlist>)

Example

LIST:TOUTput:BOSTep ON,1,(@1)

[SOURce:]LIST:TOUTput:EOSTep[:DATA] 0|OFF|1|ON,<step>,(@<chanlist>)

This command specifies which list steps generate a trigger signal at the end of the step (EOSTep).

A trigger is only generated when the state is set to ON. The trigger signal can be used as a trigger source for measurements and transients of other channels, and for digital port pins configured as trigger outputs.

Syntax

[SOURce:]LIST:TOUTput:EOSTep[:DATA] 0|OFF|1|ON,<step>,(@<chanlist>)

Query Syntax

[SOURce:]LIST:TOUTput:EOSTep[:DATA]? <step>,(@<chanlist>)

Example

LIST:TOUTput:EOSTep ON,1,(@1)

[SOURce:]LIST:COUNt:RUN? (@<chanlist>)

This command queries the number of repetitions of the current LIST run.

Syntax

[SOURce:]LIST:COUNt:RUN? (@<chanlist>)

Returns

<NRf+>

Example

LIST:COUN:RUN? (@1)

[SOURce:]LIST:STEP:RUN? (@<chanlist>)

This command is used to query the number of steps of the LIST run.

Syntax

[SOURce:]LIST:STEP:RUN? (@<chanlist>)

Returns：

<NRf+>

Example

LIST:STEP:RUN? (@1)

[SOURce]LIST:FILE:OPEN <string>,LOCal|USB,(@<chanlist>)

This command is used to open a list file in the specified storage path.

<string>: file name

<local_type>:LOCcal/USB

Syntax

[SOURce]LIST:FILE:OPEN <string>,LOCal|USB,(@<chanlist>)

Example

LIST:FILE:OPEN “test.csv”,USB,(@1)

[SOURce]LIST:FILE:SAVE <string>,LOCal|USB,(@<chanlist>)

This command is used to save the list file in the specified storage path.

<string>: file name

<local_type>LOCal/USB

Syntax

[SOURce]LIST:FILE:SAVE <string>,LOCal|USB,(@<chanlist>)

Example

LIST:FILE:SAVE “test.csv”,USB,(@1)

[SOURce:]LIST:FUNCtion:TYPE <type>,(@<chanlist>)

This command is used to set list mode.

Syntax

[SOURce:]LIST:FUNCtion:TYPE <type>,(@<chanlist>)

Arguments

CURRent|VOLTage|POWer|RESistance

Example

LIST:FUNCtion:TYPE VOLTage,(@1)

Copyright ©ITECH Electronic Co., Ltd.

5.9 Trigger Subsystem

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

This command is used to generate a sampling system immediate trigger signal.

Syntax

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

Example

TRIG:ACQ (@1)

TRIGger:ACQuire:SOURce <source>, (@<chanlist>)

The commands set and query the trigger source of the acquisition system.

Syntax

TRIGger:ACQuire:SOURce <source>, (@<chanlist>)

Arguments

IMMediate|BUS|PIN<1-7>|CURRent<1-8>|VOLTage<1-8>

Query Syntax

TRIGger:ACQuire:SOURce? (@<chanlist>)

Returns

IMM,BUS,,PIN<1-7>,CURR<1-8>,VOLT<1-8>

Example

TRIG:ACQ:SOUR IMM,(@1)

TRIG:ACQ:SOUR VOLTage8

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

The commands set and query the trigger level of the acquisition voltage. (Unit is V)

Syntax

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

Arguments

0-MAXimum|MINimum|MAXimum

Query Syntax

TRIGger:ACQuire:VOLTage[:LEVel]? [MINimum|MAXimum,] (@<chanlist>)

Returns

[MINimum|MAXimum,]

Example

TRIG:ACQ:VOLT? (@1)

TRIGger:ACQuire:VOLTage:SLOPe POSitive|NEGative, (@<chanlist>)

The commands set and query the trigger edge selection of the acquisition voltage.

Syntax

TRIGger:ACQuire:VOLTage:SLOPe POSitive|NEGative, (@<chanlist>)

Arguments

POSitive|NEGative

Query Syntax

TRIGger:ACQuire:VOLTage:SLOPe? (@<chanlist>)

Returns

[MINimum|MAXimum,]

Example

TRIG:ACQ:VOLT:SLOP NEG,(@1)

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

The commands set and query the trigger level of the acquisition current. (Unit is A)

Syntax

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

Arguments

0-MAXimum|MINimum|MAXimum

Query Syntax

TRIGger:ACQuire:CURRent[:LEVel]? [MINimum|MAXimum,] (@<chanlist>)

Returns

<voltage trig level>

Example

TRIG:ACQ:CURR? (@1)

TRIGger:ACQuire:CURRent:SLOPe POSitive|NEGative, (@<chanlist>)

The commands set and query the trigger edge selection of the acquisition current.

Syntax

TRIGger:ACQuire:CURRent:SLOPe POSitive|NEGative, (@<chanlist>)

Copyright ©ITECH Electronic Co., Ltd.

Arguments

POSitive|NEGative

Query Syntax

TRIGger:ACQuire:CURRent:SLOPe? (@<chanlist>)

Returns

POS or NEG

Example

TRIG:ACQ:CURR:SLOP NEG,(@1)

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

This command is used to generate an external data logger immediate trigger signal.

Syntax

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

Example

TRIG:ACQ (@1)

TRIGger:ELOG:SOURce <source>, (@<chanlist>)

The commands set and query the trigger source of external data records.

Syntax

TRIGger:ELOG:SOURce? (@<chanlist>)

Arguments

IMMediate|BUS|PIN<1-7>

Query Syntax

TRIGger:ELOG:SOURce? (@<chanlist>)

Returns

IMMediate|BUS|PIN<1-7>

Example

TRIG:ELOG:SOUR BUS,(@1)

TRIG:ELOG:SOUR? (@1)

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

This command is used to generate an internal data logger immediate trigger signal.

Syntax

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

Example

TRIG:DLOG (@1)

TRIGger:DLOG:SOURce <source>

The commands set and query the trigger source of internal data logger.

Syntax

TRIGger:DLOG:SOURce <source>

Arguments

IMMediate|BUS|MANual|PIN<1-7>|CURRent<1-8>|VOLTage<1-8>|TRSK|OOOK

TRSK: The transient system function switch is the trigger source;

OOOK: The on/off switch is the trigger source;

Query Syntax

TRIGger:DLOG:SOURce?

Returns

IMM,BUS,MAN,PIN<1-7>,CURR<1-8>,VOLT<1-8>,TRSK,OOOK

Example

TRIG:DLOG:SOUR BUS

TRIG:DLOG:SOUR VOLT8

If CH8 does not exist or exists as a slave, VOLTage8 may not be set as the trigger source for CH1

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

The commands set and query the trigger level for the internal data logger voltage. (Unit is V)

Syntax

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

Arguments

0-MAXimum|MINimum|MAXimum

Query Syntax

TRIGger:DLOG:VOLTage[:LEVel]? [MINimum|MAXimum,] (@<chanlist>)

Returns

[MINimum|MAXimum,]

Example

TRIG:DLOG:VOLT 1.0,(@1)

TRIGger:DLOG:VOLTage:SLOPe POSitive|NEGative, (@<chanlist>)

The commands set and query the trigger edge selection of the internal data logger voltage.

Syntax

TRIGger:DLOG:VOLTage:SLOPe POSitive|NEGative, (@<chanlist>)

Arguments

POSitive|NEGative

Query Syntax

TRIGger:DLOG:VOLTage:SLOPe? (@<chanlist>)

Returns

[MINimum|MAXimum,]

Example

TRIG:ACQ:VOLT:SLOP NEG,(@1)

TRIGger:DLOG:CURRent[:LEVel] <value>, (@<chanlist>)

The commands set and query the trigger level of the internal data logger current. (Unit is A)

Syntax

TRIGger:DLOG:CURRent[:LEVel] <value>, (@<chanlist>)

Arguments

0-MAXimum|MINimum|MAXimum

Query Syntax

TRIGger:DLOG:CURRent[:LEVel]? [MINimum|MAXimum,] (@<chanlist>)

Returns

[MINimum|MAXimum,]

Example

TRIG:ACQ:CURR? (@1)

TRIGger:DLOG:CURRent:SLOPe POSitive|NEGative, (@<chanlist>)

The commands set and query the trigger edge selection of the internal data logger current.

Syntax

TRIGger:DLOG:CURRent:SLOPe POSitive|NEGative, (@<chanlist>)

Arguments

POSitive|NEGative

Copyright ©ITECH Electronic Co., Ltd.

Query Syntax

TRIGger:DLOG:CURRent:SLOPe? (@<chanlist>)

Returns

POS or NEG

Example

TRIG:DLOG:CURR:SLOP NEG,(@1)

TRIGger:TRANsient[:IMMediate] (@<chanlist>)

This command is used to generate a transient system immediate trigger signal.

Syntax

TRIGger:TRANsient[:IMMediate] (@<chanlist>)

Example

TRIG:TRAN (@1)

TRIGger:MAP:DC[:STATe] 0|OFF|1|ON, (@chanlist)

This command is used to set and query the status of the external output of a trigger signal when the DC value is changed.

Syntax

TRIGger:MAP:DC[:STATe] 0|OFF|1|ON, (@chanlist)

Arguments

OFF|ON

Query Syntax

TRIGger:MAP:DC[:STATe]? (@chanlist)

Returns

0 or 1

Example

TRIG:MAP:DC? (@1)

TRIGger:MAP:DC:PIN PIN<1-7>, (@chanlist)

Set and query the pin that outputs a trigger signal externally when the DC value is changed.

Syntax

TRIGger:MAP:DC:PIN PIN<1-7>, (@chanlist)

Arguments

PIN<1-7>

Query Syntax

TRIGger:MAP:DC:PIN? (@chanlist)

Returns

PIN<1-7>

Example

TRIG:MAP:DC:PIN PIN1,(@1)

TRIGger:MAP:ASToutput[:STATe] 0|1|OFF|ON, (@chanlist)

The commands set and query the status of outputting one trigger signal when ARB step changes.

Syntax

TRIGger:MAP:ASToutput[:STATe] 0|1|OFF|ON, (@chanlist)

Arguments

OFF|ON

Query Syntax

TRIGger:MAP:ASToutput[:STATe]? (@chanlist)

Returns

0 or 1

Example

TRIG:MAP:AST ON,(@1)

TRIGger:MAP:ASToutput:PIN PIN<1-7>, (@chanlist)

This commands set and query the pin that outputs a trigger signal when the Arb step changes.

Syntax

TRIGger:MAP:ASToutput:PIN PIN<1-7>, (@chanlist)

Arguments

PIN<1-7>

Query Syntax

TRIGger:MAP:ASToutput:PIN? (@chanlist)

Returns

PIN<1-7>

Example

TRIG:MAP:AST:PIN PIN1,(@1)

5.10 Digital Subsystem

[SOURce:]DIGital:INPut:DATA?

This command is used to read the status of the digital IO inputs.

Syntax

[SOURce:]DIGital:INPut:DATA?

Returns

<bit value>

Example

DIG:INP:DATA?

[SOURce:]DIGital:OUTPut:DATA <value>

The commands set and query the status of digital IO output.

Syntax

[SOURce:]DIGital:OUTPut:DATA <value>

Arguments

<0-127>

Query Syntax

[SOURce:]DIGital:OUTPut:DATA?

Returns

<bit value>

Example

DIG:OUTP:DATA 21

21 corresponds to bit value 0001 0101

Indicates that the output state of IO-1 IO-3 IO-5 is on.

[SOURce:]DIGital:PIN<1-7>:FUNCtion <function>

The commands set and query the functions of digital IOs. DIO|DINPut|TINput|TOUTput are generic functions, and the all of the IO1-IO7 support the generic functions.

INHibit: IO1 support

FAULt: IO2 support

PSCLear: IO3 support

ONOFfstate: IO4 support

SYNCin,SYNCout: IO5 support

ONCouple: IO6 support

OFFCouple: IO7 support

Syntax

[SOURce:]DIGital:PIN<1-7>:FUNCtion <function>

Arguments

<DIO|DINPut|FAULt|INHibit|ONCouple|OFFCouple|TINPut|TOUTput|ONOFfstate|SIN|SOUT|PSCLear>

Query Syntax

[SOURce:]DIGital:PIN<1-7>:FUNCtion?

Returns

DIO or DINP or FAUL or INH or ONC or OFFC or TINP or TOUT or ONOF or SIN or SOUT or PSCL

Example

DIGital:PIN1:FUNCtion DIO

[SOURce:]DIGital:PIN<1-7>:POLarity POSitive|NEGative

The commands set and query the polarity of the digital IO pin.

Syntax

[SOURce:]DIGital:PIN<1-7>:POLarity POSitive|NEGative

Arguments

POSitive|NEGative

Query Syntax

[SOURce:]DIGital:PIN<1-7>:POLarity?

Returns

POS or NEG

Example

DIGital:PIN1:POLarity POSitive

[SOURce:]DIGital:ONOFf:STATe:CHANnel <1-8>

This command is used to set and query the corresponding channel number of IO4 indicating onoff state.

Syntax

[SOURce:]DIGital:ONOFf:STATe:CHANnel <1-8>

Arguments

<1-8>

Query Syntax

[SOURce:]DIGital:ONOFf:STATe:CHANnel?

Returns

<1-8>

Example

DIGital:ONOFf:STATe:CHANnel 2

[SOURce:]DIGital:PLUSe:WIDTh <value>

The commands set and query the width of digital IO pulse.

Syntax

[SOURce:]DIGital:PLUSe:WIDTh <value>

Arguments

<value>: pulse width

MAXimum: 500us

MINimum: 5us

DEFault: 20us

Query Syntax

[SOURce:]DIGital:PLUSe:WIDTh? [MAXimum|MINimum|DEFault]

Returns

<value>

Example

DIGital:PLUSe:WIDTh 30

5.11 BATTery CHARge Subsystem

BATTery:CHARge:MODE <CURRent|VOLTage>,(@<chanlist>)

This commands set the battery charging mode.

Syntax

BATTery:CHARge:MODE <CURRent|VOLTage>,(@<chanlist>)

Arguments

CURRent|VOLTage

Example

BATT:CHAR:MODE CURR,(@1)

Query Syntax

BATTery:CHARge:MODE? (@<chanlist>)

Returns

CURRent|VOLTage

BATTery:CHARge:VOLTage <value>,(@<chanlist>)

Set the voltage for battery charging.

Syntax

BATTery:CHARge:VOLTage <value>,(@<chanlist>)

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:CHAR:VOLT 5.0,(@1)

Query Syntax

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

Returns

NR3

BATTery:CHARge:CURRent <value>,(@<chanlist>)

Set the current limit value for battery charging.

Syntax

BATTery:CHARge:CURRent <value>,(@<chanlist>)

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:CHAR:CURR 3.0,(@1)

Query Syntax

BATTery:CHARge:CURRent? [MINimum|MAXimum],(@<chanlist>)

Returns

NR3

BATTery:CHARge:STOP:VOLTage <value>,(@<chanlist>)

This command sets the stop charging voltage level for battery charging test.

Syntax

BATTery:CHARge:STOP:VOLTage <value>,(@<chanlist>)

Arguments

0-MAXimum|MINimum|MAXimum (Unit is V)

Example

BATT:CHAR:STOP:VOLT 4.0,(@1)

Query Syntax

BATTery:CHARge:STOP:VOLTage? [MINimum|MAXimum,] (@<chanlist>)

Returns

NR3

BATTery:CHARge:STOP:VOLTage:STATe <0|1|OFF|ON>,(@<chanlist>)

This command sets the stop charging voltage status for battery charging test.

Syntax

BATTery:CHARge:STOP:VOLTage:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATTery:CHAR:STOP:VOLT:STAT 1,(@1)

Query Syntax

BATTery:CHARge:STOP:VOLTage? [MINimum|MAXimum,] (@<chanlist>)

Returns

0|1

BATTery:CHARge:STOP:CURRent <value>,(@<chanlist>)

This command sets and queries the stop charging current value for bettery charging test.

Syntax

BATTery:CHARge:STOP:CURRent <value>,(@<chanlist>)

Arguments

0-MAXimum|MINimum|MAXimum (Unit is A)

Example

BATT:CHAR:STOP:CURR 2.0, (@1)

Query Syntax

BATTery:CHARge:STOP:CURRent? <value>,(@<chanlist>)

Returns

NR3

BATTery:CHARge:STOP:CURRent:STATe <0|1|OFF|ON>,(@<chanlist>)

This command sets the stop charging current status for battery charging test.

Syntax

BATTery:CHARge:STOP:CURRent:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATT:CHAR:STOP:CURR:STAT 1,(@1)

Query Syntax

BATTery:CHARge:STOP:CURRent:STATe? (@<chanlist>)

Returns

0|1

BATTery:CHARge:STOP:AHOur <value>,(@<chanlist>)

This command sets and queries the stop Ahour for battery charging.

Syntax

BATTery:CHARge:STOP:AHOur <value>,(@<chanlist>)

Arguments

0.01-999.99|MINimum|MAXimum (unit is AH)

Example

BATT:CHAR:STOP:AHO 0.1,(@1)

Query Syntax

BATTery:CHARge:STOP:AHOur? [MINimum|MAXimum,] (@<chanlist>)

Returns

MINimum|MAXimum

BATTery:CHARge:STOP:AHOur:STATe <0|1|OFF|ON>,(@<chanlist>)

This command sets the stop charging Ahour status for battery charging test.

Syntax

BATTery:CHARge:STOP:AHOur:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATT:CHAR:STOP:AHO:STAT 1,(@1)

Query Syntax

BATTery:CHARge:STOP:AHOur:STATe? (@<chanlist>)

Returns

0|1

BATTery:CHARge:STOP:WHOur <value>,(@<chanlist>)

This command sets and queries the stop Whour for battery charging.

Syntax

BATTery:CHARge:STOP:AHOur <value>,(@<chanlist>)

Arguments

0.01-999.99|MINimum|MAXimum (Unit is WH)

Example

BATT:CHAR:STOP:WHO 0.1,(@1)

Query Syntax

BATTery:CHARge:STOP:WHOur? [MINimum|MAXimum,] (@<chanlist>)

Returns

MINimum|MAXimum

BATTery:CHARge:STOP:WHOur:STATe <0|1|OFF|ON>,(@<chanlist>)

This command sets the stop charging Ahour status for battery charging test.

Syntax

BATTery:CHARge:STOP:WHOur:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATT:CHAR:STOP:WHO:STAT 1,(@1)

Query Syntax

BATTery:CHARge:STOP:WHOur:STATe? (@<chanlist>)

Returns

0|1

BATTery:CHARge:STOP:TIME <value>,(@<chanlist>)

This command sets and queries the stop time for battery charging.

Syntax

BATTery:CHARge:STOP:TIME <value>,(@<chanlist>)

Arguments

0-86400|MINimum|MAXimum

Example

BATT:CHAR:STOP:TIME 12.0,(@1)

Query Syntax

BATTery:CHARge:STOP:TIME? [MINimum|MAXimum,] (@<chanlist>)

Returns

MINimum|MAXimum

BATTery:CHARge:STOP:TIME:STATe <0|1|OFF|ON>,(@<chanlist>)

设置和查询电池充电的时间停止的状态。

Syntax

BATTery:CHARge:STOP:TIME:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATT:CHAR:STOP:TIME:STAT 0,(@1)

Query Syntax

BATTery:CHARge:STOP:TIME:STATe? (@<chanlist>)

Returns

0|1

BATTERY:CHARge:SAVe <"filename">,(@<chanlist>)

This command saves the edit test item as a file, overwriting the current file if the file name already exists, otherwise, saving as a new file.

Syntax

BATTERY:CHARge:SAVe <"filename">,(@<chanlist>)

Arguments

<string>

Example

BATT:CHAR:SAVe (@1),"test.csv"

BATTery:CHARge:RECall <"filename">,(@<chanlist>)

This command recalls the battery charging test file stored in instrument.

Syntax

BATTery:CHARge:RECall <"filename">,(@<chanlist>)

Arguments

<string>

Example

BATT:CHAR:RECall "test.csv",(@1)

BATTery:CHARge:DELete <"filename">,(@<chanlist>)

This command deletes the battery charging test file stored in instrument.

Syntax

BATTery:CHARge:DELete <"filename">,(@<chanlist>)

Arguments

<string>

Example

BATT:CHAR:DEL "test.csv",(@1)

BATTery:CHARge:TIME? (@<chanlist>)

This command queries how long the battery charge has been running.

Syntax

BATTery:CHARge:TIME? (@<chanlist>)

Returns

<NRF>

Example

BATT:CHAR:TIME? (@1)

BATTery:CHARge:RSTate? (@<chanlist>)

This command queries the battery charge status.

Syntax

BATTery:CHARge:RSTate? (@<chanlist>)

Returns

"Idle","Charging","Voltage Up","Current Up","Energy Up","Time up","Abort"

Example

BATT:CHAR:RST? (@1)

5.12 BATTery DISCharge Subsystem

BATTery:DISCharge:MODE <CURRent|VOLTage>,(@<chanlist>)

This commands set the battery discharging mode.

Syntax

BATTery:DISCharge:MODE <CURRent|VOLTage>,(@<chanlist>)

Arguments

CURRent|VOLTage

Example

BATT:DISC:MODE CURR,(@1)

Query Syntax

BATTery:DISCharge:MODE? (@<chanlist>)

Returns

CURRent|VOLTage

BATTery:DISCharge:VOLTage <value>,(@<chanlist>)

Set the voltage for battery discharging.

Syntax

BATTery:DISCharge:VOLTage <value>,(@<chanlist>)

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:DISC:VOLT 5.0,(@1)

Query Syntax

BATTery:DISCharge:VOLTage? [MINimum|MAXimum],(@<chanlist>)

Copyright ©ITECH Electronic Co., Ltd.

Returns

NR3

BATTery:DISCharge:CURRent <value>,(@<chanlist>)

Set the current for battery discharging.

Syntax

BATTery:DISCharge:CURRent <value>,(@<chanlist>)

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:DISC:CURR 3.0,(@1)

Query Syntax

BATTery:DISCharge:CURRent? [MINimum|MAXimum],(@<chanlist>)

Returns

NR3

BATTery:DISCharge:POWer <value>,(@<chanlist>)

Set the power for battery discharging.

Syntax

BATTery:DISCharge:POWer <value>,(@<chanlist>)

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:DISC:POW 3.0,(@1)

Query Syntax

BATTery:DISCharge:POWer? [MINimum|MAXimum,] (@<chanlist>)

Returns

NR3

BATTery:DISCharge:RESistance <value>,(@<chanlist>)

Set and query the battery discharge resistance.

Syntax

BATTery:DISCharge:RESistance <value>,(@<chanlist>)

Arguments

0-MAXimum|MINimum|MAXimum

Example

BATT:DISC:RESistance 1000,(@1)

Query Syntax

BATTery:DISCharge:RESistance? [MINimum|MAXimum,] (@<chanlist>)

Returns

NR3

BATTery:DISCharge:STOP:VOLTage <value>,(@<chanlist>)

This command sets the stop discharging voltage level for battery discharging test.

Syntax

BATTery:DISCharge:STOP:VOLTage <value>,(@<chanlist>)

Arguments

0-MAXimum|MINimum|MAXimum (unit is V)

Example

BATT:DISCharge:STOP:VOLT 4.0,(@1)

Query Syntax

BATTery:DISCharge:STOP:VOLTage? [MINimum|MAXimum,] (@<chanlist>)

Returns

NR3

BATTery:DISCharge:STOP:VOLTage:STATe <0|1|OFF|ON>,(@<chanlist>)

This command sets the stop discharging voltage status for battery discharging test.

Syntax

BATTery:DISCharge:STOP:VOLTage:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATTery:DISC:STOP:VOLT:STAT 1,(@1)

Query Syntax

BATTery:DISCharge:STOP:VOLTage? [MINimum|MAXimum,] (@<chanlist>)

Returns

0|1

BATTery:DISCharge:STOP:CURRent <value>,(@<chanlist>)

This command sets the stop discharging current level for battery discharging test.

Syntax

BATTery:DISCharge:STOP:CURRent <value>,(@<chanlist>)

Arguments

0-MAXimum|MINimum|MAXimum (Unit is A)

Example

BATT:DISC:STOP:CURR 2.0, (@1)

Query Syntax

BATTery:DISC:STOP:CURRent? <value>,(@<chanlist>)

Returns

NR3

BATTery:DISCharge:STOP:CURRent:STATe <0|1|OFF|ON>,(@<chanlist>)

This command sets the stop discharging current status for battery discharging test.

Syntax

BATTery:DISCharge:STOP:CURRent:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATT:DISC:STOP:CURR:STAT 1,(@1)

Query Syntax

BATTery:DISCharge:STOP:CURRent:STATe? (@<chanlist>)

Returns

0|1

BATTery:DISCharge:STOP:AHOur <value>,(@<chanlist>)

This command sets the stop discharging AHour level for battery discharging test.

Syntax

BATTery:DISCharge:STOP:AHOur <value>,(@<chanlist>)

Copyright ©ITECH Electronic Co., Ltd.

Arguments

0.01-999.99|MINimum|MAXimum (Unit is AH)

Example

BATT:DISC:STOP:AHO 0.1,(@1)

Query Syntax

BATTery:DISCharge:STOP:AHOur? [MINimum|MAXimum,] (@<chanlist>)

Returns

MINimum|MAXimum

BATTery:DISCharge:STOP:AHOur:STATe <0|1|OFF|ON>,(@<chanlist>)

This command sets the stop discharging AHour status for battery discharging test.

Syntax

BATTery:DISCharge:STOP:AHOur:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATT:DISCharge:STOP:AHO:STAT 1,(@1)

Query Syntax

BATTery:DISCharge:STOP:AHOur:STATe? (@<chanlist>)

Returns

0|1

BATTery:DISCharge:STOP:WHOur <value>,(@<chanlist>)

This command sets the stop discharging WHour level for battery discharging test.

Syntax

BATTery:DISCharge:STOP:WHOur <value>,(@<chanlist>)

Arguments

0.01-999.99|MINimum|MAXimum (Unit is WH)

Example

BATT:DISC:STOP:WHO 0.1,(@1)

Query Syntax

BATTery:DISCharge:STOP:WHOur? [MINimum|MAXimum,] (@<chanlist>)

Copyright ©ITECH Electronic Co., Ltd.

Returns

MINimum|MAXimum

BATTery:DISCharge:STOP:WHOur:STATe <0|1|OFF|ON>,(@<chanlist>)

This command sets the stop discharging WHour status for battery discharging test.

Syntax

BATTery:DISCharge:STOP:WHOur:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATT:DISC:STOP:WHO:STAT 1,(@1)

Query Syntax

BATTery:DISCharge:STOP:WHOur:STATe? (@<chanlist>)

Returns

0|1

BATTery:DISCharge:STOP:TIME <value>,(@<chanlist>)

Set and query the stopping time of battery discharge.

Syntax

BATTery:DISCharge:STOP:TIME <value>,(@<chanlist>)

Arguments

0-86400|MINimum|MAXimum

Example

BATT:DISC:STOP:TIME 12.0,(@1)

Query Syntax

BATTery:DISCharge:STOP:TIME? [MINimum|MAXimum,] (@<chanlist>)

Returns

MINimum|MAXimum

BATTery:DISCharge:STOP:TIME:STATe <0|1|OFF|ON>,(@<chanlist>)

Setting and querying the status of the time stop of battery discharge.

Syntax

BATTery:DISCharge:STOP:TIME:STATe <0|1|OFF|ON>,(@<chanlist>)

Copyright ©ITECH Electronic Co., Ltd.

Arguments

<0|1|OFF|ON>

Example

BATT:DISC:STOP:TIME:STAT 0,(@1)

Query Syntax

BATTery:DISCharge:STOP:TIME:STATe? (@<chanlist>)

Returns

0|1

BATTERY:DISCharge:SAVe <"filename">,(@<chanlist>)

This command saves the edit test item as a file, overwriting the current file if the file name already exists, otherwise, saving as a new file.

Syntax

BATTERY:DISCharge:SAVe <"filename">,(@<chanlist>)

Arguments

<string>

Example

BATT:DISC:SAVe (@1),"test.csv"

BATTery:DISCharge:RECall <"filename">,(@<chanlist>)

This command recalls the battery discharging test file stored in instrument.

Syntax

BATTery:DISCharge:RECall <"filename">,(@<chanlist>)

Arguments

<string>

Example

BATT:DISC:RECall "test.csv",(@1)

BATTery:DISCharge:DELete <"filename">,(@<chanlist>)

This command deletes the battery discharging test file stored in instrument.

Syntax

BATTery:DISCharge:DELete <"filename">,(@<chanlist>)

Arguments

<string>

Example

BATT:DISC:DEL "test.csv",(@1)

BATTery:DISCharge:TIME? (@<chanlist>)

This command queries how long the battery discharge has been running.

Syntax

BATTery:DISC:TIME? (@<chanlist>)

Returns

<NRF>

Example

BATT:DISC:TIME? (@1)

BATTery:DISCharge:RSTate? (@<chanlist>)

This command queries the battery discharge status.

Syntax

BATTery:DISCharge:RSTate? (@<chanlist>)

Returns

"Idle","Discharging ","Voltage Up","Current Up","Energy Up","Time up","Abort"

Example

BATT:DISCharge:RST? (@1)

Chapter6 DC Load Command

IT2700 series multi-channel source and load module system can realize different functions by installing different modules, the module categories are bidirectional DC power supply module, unipolar power module and load module, this chapter introduces the commands related to DC load module.

6.1 SYSTem Subsystem

SYSTem:VERSion?

This command queries the SCPI version of the instrument.

Syntax

SYSTem:VERSion?

Arguments

None

Reset value

Not applicable

Example

- > SYST:VERS?
< - "1993.1"

Returns

SRD

SYSTem:ERRor[:NEXT]?

This order is used to query the error information of the power supply. When the error LED lights on the front panel, it tells us that there is one or more hardware or order syntax error happened. There are at most 20 groups of error message. Send the order once will read one error message from the error queue.

- Error messages follow the FIFO (first-in-first-out) principle. The first error will be returned first. when you read all the error tips information in the error queue. The ERROR led will turn off. When error occur the buzzer of the power supply will beep once.
- If more than 20 errors occurred, the last one stored in the queue will be replaced by “-350”, Meaning "too many mistakes.". If the error messages in the error message queue are not read out, the error following will not be saved to the queue. If there is no error message, when read error information, it will return “+0”, meaning “no error”.
- If turn off the power supply or send *CLS (clear status) order, the error messages in the error queue will be cleared. *RST order will not clear the error messages in the error queue.

SYSTem:CHANnel[:COUNt]?

Returns the number of output channels in a mainframe.

Syntax

SYSTem:CHANnel[:COUNt]?

Example

SYST:CHAN?

SYSTem:CHANnel:MODel? (@<chanlist>)

Returns the model numbers of the selected channels. Multiple responses are separated by commas.

Syntax

SYSTem:CHANnel:MODel? (@<chanlist>)

Example

SYST:CHAN:MOD? (@1)

SYSTem:CHANnel:OPTion? (@<chanlist>)

Returns the options installed in the selected channels. The query returns an ASCII string enclosed in double quotes. If no option is installed, an empty quoted string ("") is returned. Multiple responses are separated by commas.

Syntax

SYSTem:CHANnel:OPTion? (@<chanlist>)

Example

SYSTem:CHANnel:OPTion? (@1)

SYSTem:CHANnel:DESCription? (@<chanlist>)

Returns the description of the selected channels.Multiple responses are separated by commas.

Syntax

SYSTem:CHANnel:DESCription? (@<chanlist>)

Example

SYST:CHAN:DESC? (@1)

SYSTem:CHANnel:SERial? (@<chanlist>)

Returns the serial numbers of the selected channels. Multiple responses are separated by semicolons.

Syntax

SYSTem:CHANnel:SERial? (@<chanlist>)

Example

SYST:CHAN:SER? (@1)

SYSTem:CHANnel:STATus? (@<chanlist>)

This command is used to read the status of the channel.

Syntax

SYSTem:CHANnel:STATus? (@<chanlist>)

Example

SYST:CHAN:STAT? (@1>)

SYSTem:CHANnel:STATus:SUMMary?

This command is used to read a summary of all states.

Syntax

SYSTem:CHANnel:STATus:SUMMary?

Example

SYST:CHAN:STAT:SUMM?

SYSTem:CHANnel:VERSion? (@<chanlist>)

This command is used to read the version of the channel.

Syntax

SYSTem:CHANnel:VERSion? (@<chanlist>)

Example

SYST:CHAN:VERS? (@1)

SYSTem:CHANnel:CATegory? (@<chanlist>)

This command is used to read the model category of the channel.

Syntax

SYSTem:CHANnel:CATegory? (@<chanlist>)

Arguments

0:DC Source

1:DC Load

2:AC Source

3:AC Load

Example

SYST:CHAN:CAT? (@1)

SYSTem:CHANnel:TYPE? (@<chanlist>)

This command is used to read the model type of the channel.

Syntax

SYSTem:CHANnel:TYPE? (@<chanlist>)

Arguments

0: bidirectional source

1: DC source

2: DC load

3: High Precision SMU

4: AC source

5: AC load

Example

SYSTem:CHANnel:TYPE? (@2)

SYSTem:COMMunicate:RLSTate LOCal|REMote|RWLock

This command is used to set the remote or local state. local: the instrument operation mode is switched to local mode, Remote: the instrument is switched to remote control mode. rwlock: the device goes to the remote state and locks the local LOCAL key.

This command has the same function as SYSTem:REMote/ SYSTem:LOCal/ SYSTem:RWLock.

Syntax

SYSTem:COMMunicate:RLSTate LOCal|REMote|RWLock

Example

SYST:COMM:RLST REM

Query syntax

SYSTem:COMMunicate:RLSTate?

SYSTem:REMote

This command takes the instrument out of front-panel control mode and switches it to remote control mode.

Syntax

SYSTem:REMote

Example

SYST:REM

SYSTem:LOCal

This command is used to switch the power supply into the control from the front

panel.

Syntax

SYSTem:LOCal

Example

SYST:LOC

SYSTem:RWLock

This command locks the power supply in remote control mode. When this command is executed, pressing the LOCAL button does not switch the instrument to local control mode.

Syntax

SYSTem:RWLock

Example

SYST:RWL

SYSTem:KEY <0-255>

This command is used to simulate a key.

Syntax

SYSTem:KEY <0-255>

Arguments

0-255

Example

SYST:KEY

Query syntax

SYSTem:KEY?

SYSTem:PRESet

This command is used to make the instrument in a state suitable for panel operation(reset).

Syntax

SYSTem:PRESet

Example

SYST:PRES

SYSTem:POSetup <CPD>

This command is used to set and query some parameters or working status 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: 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.

Syntax

SYSTem:POSetup <CPD>

Arguments

RST|LAST|LAST_OFF

*RST

RST

Example

SYST:POS RST

Query syntax

SYSTem:POSetup?

SYSTem:CLEar

This command is used to clear the error queue.

Syntax

SYSTem:CLEar

Example

SYST:CLE

SYSTem:BEEPer:IMMediate

This command tests the beeper function of the power supply. If it passes the test, a beep is issued.

Syntax

SYSTem:BEEPer:IMMediate

Example

SYST:BEEP:IMM

SYSTem:BEEPer:CLICk <0|1|OFF|ON>

This command is used to set the button click buzzer.

Syntax

SYSTem:BEEPer:CLICk <0|1|OFF|ON>

Arguments

0|1|OFF|ON

*RST

ON

Return Arguments

0/1

Example

SYSTem:BEEPer:CLICk ON

Query syntax

SYSTem:BEEPer:CLICk?

SYSTem:BEEPer:ALARm <0|1|OFF|ON>

This command is used to set the buzzer sound state in the alarm state.

Syntax

SYSTem:BEEPer:ALARm <0|1|OFF|ON>

Arguments

0|1|OFF|ON

*RST

ON

Return Arguments

0/1

Example

SYSTem:BEEPer:ALARm ON

Query syntax

SYSTem:BEEPer:ALARm?

SYSTem:COMMunicate:USB:TYPE <CPD>

This command is used to set and query the USB interface type. IT2700 series power supply USB can be set as the host or device type, used to set the USB interface for communication or as a storage disk. When set as host type, it is used as a storage disk, and when set as device type, it is used as a communication interface.

Syntax

SYSTem:COMMunicate:USB:TYPE <CPD>

Arguments

DEVice|HOST

默认 Arguments

HOST

Return Arguments

DEVice|HOST

Example

SYST:COMM:USB:TYPE HOST

Query syntax

SYSTem:COMMunicate:USB:TYPE?

SYSTem:COMMunicate:SELect <TMC|VCP>

This command is used to set and query the USB interface type.

Syntax

SYSTem:COMMunicate:SELect <TMC|VCP>

Arguments

TMC|VCP

默认 Arguments

VCP

Return Arguments

TMC|VCP

Example

SYST:COMM:SEL VCP

Query syntax

SYSTem:COMMunicate:SELect?

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

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

Syntax

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

Arguments

<SPD>

默认 Arguments

"192.168.0.11"

Return Arguments

<SPD>

Example

SYST:COMM:LAN:IP "192.168.0.11"

Query syntax

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

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

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

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

Syntax

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

Arguments

<CPD>

AUTO|MANual

*RST

MANual

Return Arguments

AUTO|MANual

Example

SYST:COMM:LAN:IP:MODE AUTO

Query syntax

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

SYSTem:COMMunicate:LAN:SMASk <SPD>

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

Syntax

SYSTem:COMMunicate:LAN:SMASk <SPD>

Copyright ©ITECH Electronic Co., Ltd.

Arguments

<SPD>

*RST

"255.255.255.0"

Return Arguments

<SPD>

Example

SYST:COMM:LAN:SMAS "255.255.255.1"

Query syntax

SYSTem:COMMunicate:LAN:SMASk?

SYSTem:COMMunicate:LAN:DGATeway <SPD>

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

Syntax

SYSTem:COMMunicate:LAN:DGATeway <SPD>

Arguments

<SPD>

*RST

"192.168.0.1"

Example

SYST:COMM:LAN:DGAT "192.168.0.1"

Query syntax

SYST:COMM:LAN:DGAT?

SYSTem:COMMunicate:LAN:RAWSocketport <port>

This command is used to set and query the socket port of the LAN communication.

Syntax

SYSTem:COMMunicate:LAN:RAWSocketport <port>

Arguments

<SPD>

*RST

"30000"

Example

SYSTem:COMMunicate:LAN:RAWSocketport 30001

Query syntax

SYSTem:COMMunicate:LAN:RAWSocketport?

SYSTem:BRIGhtness:LEVel <NR1>

This command is used to set and query the screen brightness of the present power supply, the setting range is 1-10.

Syntax

SYSTem:BRIGhtness:LEVel <NR1>

Arguments

1-10

Return Arguments

<NR1>

*RST

8

Example

SYST:BRIG:LEVel 10

Query syntax

SYSTem:BRIGhtness:LEVel?

SYSTem:TOUCh[:STATe] <CPD>

This command is used to set and query the touch screen status of the screen, 0|OFF means the touch screen is turned off, 1|ON means the touch screen is turned on.

Syntax

SYSTem:TOUCh[:STATe] <CPD>

Arguments

<CPD>

0|OFF|1|ON

Return Arguments

0|1

Example

SYST:TOUC 0

Query syntax

SYST:TOUC?

SYSTem:LANGuage <CPD>

This command is used to set and query the language display of the power supply, Chinese or English can be selected.

Syntax

SYSTem:LANGuage <CPD>

Arguments

ENGLish|CHINese

Example

SYST:LANG CHIN

Query syntax

SYSTem:LANGuage?

SYSTem:SOFT:KEYBoard[:STATe] <CPD>

This command is used to set and query the UI soft keyboard switch of the present power supply.

Syntax

SYSTem:SOFT:KEYBoard[:STATe] <CPD>

Arguments

0|OFF|1|ON

Example

SYST:SOFT:KEYB 1

Query syntax

SYSTem:SOFT:KEYBoard[:STATe]?

SYSTem:KNOB:IMMediate:EFFective <CPD>

This command is used to set and query whether the parameters adjusted by the knob take effect immediately.

Syntax

SYSTem:KNOB:IMMediate:EFFective <CPD>

Arguments

0|OFF|1|ON

*RST

0

Return Arguments

0 |1

Example

SYST:KNOB:IMM:EFF 1

Query syntax

SYSTem:KNOB:IMMediate:EFFective?

SYSTem:RUN:TIME?

Query machine running time

More than one day returns: xxd hh:mm:ss, xxd means days, days and minutes, minutes and seconds followed by two spaces between, Within one day: hh:mm:ss

Syntax

SYSTem:RUN:TIME?

Return Arguments

xxd hh:mm:ss

SYSTem:GROup:DEFine (@<chanlist>)

Define many channels as a single group.

Syntax

SYSTem:GROup:DEFine (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4

Example

SYST:GRO:DEF (@1,3,5)

SYSTem:GROup:DELete (@<chanlist>)

Deleting specific channels in a group.

Syntax

SYSTem:GROup:DELete (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – channel 1
(@1,4) – channels 1 and 4

(@1:4) – channels 1 through 4

Example

SYST:GRO:DEL (@3)

SYSTem:GROup:DELete:ALL

Deleting specific channels in a group.

Syntax

SYSTem:GROup:DELete:ALL

Example

SYST:GRO:DEL:ALL

SYSTem:GROup:APPLy

Apply the group setting.

Syntax

SYSTem:GROup:APPLy

Example

SYST:GRO:APPL

SYSTem:REBoot

Reboot the instrument.

Syntax

SYSTem:REBoot

Example

SYST:REB

SYSTem:SECurity:IMMediate

Syntax

SYSTem:SECurity:IMMediate

Example

SYST:SEC:IMM

SYSTem:LOAD:CURRent:SYMBol <POSitive|NEGative>

This command is used to set the sign of the load current.

Syntax

SYSTem:LOAD:CURRent:SYMBol <POSitive|NEGative>

Arguments

POSitive|NEGative

Example

SYSTem:LOAD:CURRent:SYMBol POSitive

Query syntax

SYSTem:LOAD:CURRent:SYMBol?

Return Arguments

POSitive|NEGative

6.2 ABORt Subsystem

ABORt:ACQuire (@<chanlist>)

This command is used to stop the meter function.

Syntax

ABORt:ACQuire

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4.

Example

ABOR:ACQ (@1)

ABORt[:ALL]

This command is used to stop all of function.

Syntax

ABORt[:ALL]

Example

ABOR[:ALL]

ABORt:DLOG

This command is used to stop the data logger.

Syntax

ABORt:DLOG

Example

ABOR:DLOG

ABORt:ELOG (@<chanlist>)

This command is used to stop the external logger.

Syntax

ABORt:ELOG (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4.

Example

ABORt:ELOG (@1)

ABORt:TRANsient (@<chanlist>)

This command is used to stop the transient system.

Syntax

ABORt:TRANsient (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4.

Example

ABORt:TRANsient (@1)

6.3 INITiate Subsystem

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

This command is used to initiates the measurement trigger system.

Syntax

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

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Example

INIT:ACQ (@1)

INITiate[:IMMediate]:DLOG <"filename">

This command is used to initiates internal data logging.

Syntax

INITiate[:IMMediate]:DLOG <"filename">

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4.

Example

INIT:LIST

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

This command is used to initiates external data logging

Syntax

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

Example

INIT:ELOG (@1)

INITiate[:IMMediate]:TRANsient (@<chanlist>)

This command is used to initiates the transient trigger system.

Syntax

INITiate[:IMMediate]:TRANsient (@<chanlist>)

Example

INIT:TRAN (@1)

INITiate:CONTinuous[:ACQuire] 0|OFF|1|ON,(@<chanlist>)

This command is used to set the continuous startup state of the acquisition trigger system.

Syntax

INITiate:CONTinuous[:ACQuire] 0|OFF|1|ON,(@<chanlist>)

Arguments

0|1|OFF|ON

Query syntax

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

Example

INITiate:CONTinuous 0,(@1)

INITiate:CONTinuous:TRANsient 0|OFF|1|ON,(@<chanlist>)

This command sets the continuous startup state of the transient trigger system.

Syntax

INITiate:CONTinuous:TRANsient 0|OFF|1|ON,(@<chanlist>)

Arguments

0|1|OFF|ON

Query syntax

INITiate:CONTinuous:TRANsient? (@<chanlist>)

Example

INITiate:CONTinuous:TRANsient 0,(@1)

6.4 SOURce Subsystem

[SOURce:]FUNCtion CURRent|VOLTage, (@<chanlist>)

This command is used to set and query the priority mode of the function.

Syntax

[SOURce:]FUNCtion CURRent|VOLTage|POWer|RESistance|CC+CV|CR+CV|CP+CV|CC+CR|CC+CV+CP+CR|BSIM, (@<chanlist>)

Arguments

Source mode: CURRent|VOLTage|POWer|RESistance

Load mode:

CURRent|VOLTage|POWer|RESistance|CC+CV|CR+CV|CP+CV|CC+CR|CC+CV+CP+CR|BSIM

Query syntax

[SOURce:]FUNCtion? (@<chanlist>)

*RST

Load mode: CURRent

Source mode: VOLTage

Return Arguments

CURR,VOLT,POW or RES

Example

FUNC CURR ,(@1)

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

This command sets the immediate voltage level of the specified output channel. Units are in volts. The immediate level is the output voltage setting.

Syntax

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

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude]? [MIN|MAX,] (@<chanlist>)

*RST

0

Return Arguments

0-maximum|MINimum|MAXimum

Example

VOLTage 100, (@1)

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>)

This command sets the triggered voltage level of the specified output channel. Units are in volts.

Syntax

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]VOLTage[:LEVel]:TRIGgered[:AMPLitude]? [MIN|MAX,] (@<chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

VOLTage:TRIGgered 50, (@1)

[SOURce:]VOLTage:PROTection[:LOCal][:POSitive][:LEVel] <value>, (@<chanlist>)

This command sets the over-voltage protection (OVP) level of the output channel. The values are programmed in volts. If the output voltage exceeds the OVP level, the output is disabled.

Syntax

[SOURce:]VOLTage:PROTection[:LOCal][:POSitive][:LEVel] <value>, (@<chanlist>)

Query syntax

[SOURce:]VOLTage:PROTection[:LOCal][:POSitive][:LEVel]? [MIN|MAX,] (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4

*RST

MAX

Example

VOLTage:PROTection 300, (@1)

[SOURce:]VOLTage:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>)

This command enables or disables the over-current protection (OVP) function. The enabled state is On (1); the disabled state is Off (0). If the over-voltage protection function is enabled and the output is disabled.

Syntax

[SOURce:]VOLTage:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>)

Arguments

<0|OFF|1|ON>0-OFF|1-ON

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

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

Example

VOLTage:PROTection:STATe ON, (@1)

[SOURce:]VOLTage:PROTection:DELay[:TIME] <value>, (@<chanlist>)

This command sets the over-voltage protection delay. The over-voltage protection function will not be triggered on the specified output channel during the delay time. After the delay time has expired, the over-voltage protection function will be active.

Syntax

[SOURce:]VOLTage:PROTection:DELay[:TIME] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]VOLTage:PROTection:DELay[:TIME]? [MIN|MAX,] (@<chanlist>)

Example

VOLTage:PROTection:DELay 30, (@1)

[SOURce:]VOLTage:UNDer:PROTection[:POSitive][:LEVel] <value>, (@<chanlist>)

This command set and query the positive voltage protection value for undervoltage protection.

Syntax

[SOURce:]VOLTage:UNDer:PROTection[:POSitive][:LEVel] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]VOLTage:UNDer:PROTection[:LEVel]? [MIN|MAX,] (@<chanlist>)

Example

VOLTage:UNDer:PROTection 10, (@1)

[SOURce:]VOLTage:UNDer:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>)

This command enables or disables the under-voltage protection (UVP) function. The enabled state is On (1); the disabled state is Off (0). If the under-voltage protection function is enabled and the output is disabled.

Syntax

[SOURce:]VOLTage:UNDer:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>)

Arguments

<0|OFF|1|ON>0-OFF|1-ON

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – - channels 1 and 2
(@1:4) – channels 1 through 4.

Query syntax

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

Example

VOLTage:UNDer:PROTection:STATe ON, (@1)

[SOURce:]VOLTage:UNDer:PROTection:DELay[:TIME] <value>, (@<chanlist>)

This command sets the under-voltage protection delay. The under-voltage protection function will not be triggered on the specified output channel during the delay time. After the delay time has expired, the under-voltage protection function will be active.

Syntax

[SOURce:]VOLTage:UNDer:PROTection:DELay[:TIME] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]VOLTage:UNDer:PROTection:DELay[:TIME]? [MIN|MAX,] (@<chanlist>)

Example

VOLTage:UNDer:PROTection:DELay 10, (@1)

[SOURce:]VOLTage:UNDer:PROTection:WARM[:TIME] <value>, (@<chanlist>)

This command sets the under-voltage protection warm time. The under-voltage protection function will not be triggered on the specified output channel during the warm time. even if the voltage reaches the under-voltage protection point and meets the delay time.

Syntax

[SOURce:]VOLTage:UNDer:PROTection:WARM[:TIME] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]VOLTage:UNDer:PROTection:WARM[:TIME]? [MIN|MAX,] (@<chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

VOLTage:UNDer:PROTection:WARM 10, (@1)

[SOURce:]VOLTage:RESistance[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>)

This command sets and queries the internal resistance value of the power supply in voltage priority mode. Units are in Ω.

Syntax

[SOURce:]VOLTage:RESistance[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>)

Arguments

<value>MAXimum|MINimum|DEFault

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

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

Return Arguments

MAXimum|MINimum|DEFault

Example

VOLTage:RESistance 0.1,(@1)

[SOURce:]VOLTage:RESistance:STATe <0|OFF|1|ON>, (@<chanlist>)

This command sets and queries the status of the internal resistance function of the power supply in voltage priority mode.

Syntax

[SOURce:]VOLTage:RESistance:STATe <0|OFF|1|ON>, (@<chanlist>)

Arguments

<0|OFF|1|ON>0|OFF|1|ON

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]VOLTage:RESistance:STATe? (@<chanlist>)

Return Arguments

0|OFF|1|ON

Example

VOLTage:RESistance:STATe? (@1,2)

[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

This command sets and queries the voltage rise slope.

If the setting state of the [SOURce:]VOLTage:SLEW:COUPle is ON, this command sets the voltage rise slope while simultaneously setting the voltage fall slope.

Syntax

[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate]? [MIN|MAX,] (@<chanlist>)

Example

VOLTage:SLEW 10, (@1)

[SOURce:]VOLTage:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>)

This command sets and queries the voltage fall slope.

If the setting state of the [SOURce:]VOLTage:SLEW:COUPle is ON, this command sets the voltage fall slope while simultaneously setting the voltage rise slope.

Syntax

[SOURce:]VOLTage:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]VOLTage:SLEW:NEGative[:IMMediate]? [MIN|MAX,] (@<chanlist>)

Example

VOLTage:SLEW:NEGative 10, (@1)

[SOURce:]VOLTage:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

This command sets and queries the tracking status of the voltage slope.

Syntax

[SOURce:]VOLTage:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]VOLTage:SLEW:COUPle? (@<chanlist>)

Example

VOLTage:SLEW:COUPle ON, (@1)

[SOURce:]VOLTage:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>)

Enable or disable voltage rise slope maximum.

Syntax

[SOURce:]VOLTage:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

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

Example

VOLTage:SLEW:MAXimum ON, (@1)

[SOURce:]VOLTage:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

Enable or disable voltage drop slope maximum.

Syntax

[SOURce:]VOLTage:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

Query syntax

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

Example

VOLTage:SLEW:NEGative:MAXimum ON, (@1)

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>)

This command sets the immediate current level of the specified output channel. Units are in amperes. The immediate level is the output current setting.

Syntax

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]CURRent[:LEVel][:IMMediate][:AMPLitude]? [MIN|MAX,] (@<chanlist>)

Example

CURRent 30, (@1)

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>)

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

Syntax

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]CURRent[:LEVel]:TRIGgered[:AMPLitude]? [MIN|MAX,] (@<chanlist>)

Example

CURRent:TRIGgered 20, (@1)

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

This command sets the positive current limit of the specified output channel. Units are in amperes. If [SOURce:]CURRent:LIMit:COUPle is enabled, this command also sets the value of the negative current limit.

Syntax

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

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]CURRent:LIMit[:POSitive][:IMMediate][:AMPLitude]? [MIN|MAX,] (@<chanlist>)

Copyright ©ITECH Electronic Co., Ltd.

Example

CURRent:LIMit 50, (@1)

[SOURce:]CURRent:LIMit:NEGative[:IMMediate][:AMPLitude] <value>, (@<chanlist>)

This command sets the negative current limit of the specified output channel. Units are in amperes. If [SOURce:]CURRent:LIMit:COUPle is enabled, this command also sets the value of the positive current limit.

Syntax

[SOURce:]CURRent:LIMit:NEGative[:IMMediate][:AMPLitude] <value>, (@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Query syntax

[SOURce:]CURRent:LIMit:NEGative[:IMMediate][:AMPLitude]? [MINimum|MAXimum,] (@<chanlist>)

Example

CURR:LIM:NEG -2,(@1)

[SOURce:]CURRent:LIMit:COUPle 0|OFF|1|ON, (@<chanlist>)

This command sets the current limit tracking state. The enabled state is On (1); the disabled state is Off (0). If limit tracking is enabled, the negative current limit tracks the positive current limit setting. The [SOURce:]CURRent:LIMit command sets the value of the positive and negative current limits.

This command only applies to bidirectional DC power supply models.

Syntax

[SOURce:]CURRent:LIMit:COUPle 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

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

Example

CURR:LIM:COUP ON,(@1)

[SOURce:]CURRent:PROTection[:LEVel][:POSitive] <level>,(@<chanlist>)

This command sets the positive current protection value for over-current protection.

Syntax

[SOURce:]CURRent:PROTection[:LEVel][:POSitive] <level>,(@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]CURRent:PROTection[:LEVel][:POSitive]?
[MINimum|MAXimum,](@<chanlist>)

*RST

MAX

Example

CURRent:PROTection 300, (@1)

[SOURce:]CURRent:PROTection[:LEVel]:NEGative <level>,(@<chanlist>)

This command sets the negative current protection value for over-current protection, This command only applies to bidirectional DC power supply models.

Syntax

[SOURce:]CURRent:PROTection[:LEVel]:NEGative <level>,(@<chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]CURRent:PROTection[:LEVel]:NEGative?
[MINimum|MAXimum,](@<chanlist>)

*RST

MAX

Example

CURRent:PROTection:NEGative -2,(@1:2)

[SOURce:]CURRent:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>)

This command enables or disables the over-current protection (OCP) function. The enabled state is On (1); the disabled state is Off (0). If the over-current protection function is enabled and the output is disabled.

Syntax

[SOURce:]CURRent:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>)

Query syntax

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

Example

CURRent:PROTection:STATe ON, (@1)

[SOURce:]CURRent:PROTection:DELay[:TIME] <value>, (@<chanlist>)

This command sets the over-current protection delay. The over-current protection function will not be triggered on the specified output channel during the delay time. After the delay time has expired, the over-current protection function will be active.

Syntax

[SOURce:]CURRent:PROTection:DELay[:TIME] <value>, (@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

Query syntax

[SOURce:]CURRent:PROTection:DELay[:TIME]? [MIN|MAX,] (@<chanlist>)

Example

CURRent:PROTection:DELay 30, (@1)

[SOURce:]CURRent:UNDer:PROTection[:LEVel][:POSitive] <level>,(@<chanlist>)

This command sets and queries the positive current protection value for under-current protection.

Syntax

[SOURce:]CURRent:UNDer:PROTection[:LEVel][:POSitive] <level>,(@<chanlist>)

Query syntax

[SOURce:]CURRent:UNDer:PROTection[:LEVel][:POSitive] [MINimum|MAXimum,](@<chanlist>)

Example

VOLTage:UNDer:PROTection 10, (@1)

[SOURce:]CURRent:UNDer:PROTection[:LEVel]:NEGative <level>,(@<chanlist>)

This command sets and queries the negative current protection value for under-current protection.

Syntax

[SOURce:]CURRent:UNDer:PROTection[:LEVel]:NEGative <level>,(@<chanlist>)

Query syntax

[SOURce:]CURRent:UNDer:PROTection[:LEVel]:NEGative? [MINimum|MAXimum,](@<chanlist>)

Example

CURRent:UNDer:PROTection:NEGative 10, (@1)

[SOURce:]CURRent:UNDer:PROTection:STATe <0|OFF|1|ON>, (@<chanlist>)

This command enables or disables the under-current protection (UCP) function. The enabled state is On (1); the disabled state is Off (0). If the under-current protection function is enabled and the output is disabled.

Syntax

[SOURce:]CURRent:UNDer:PROTection:STATe 0|OFF|1|ON,(@<chanlist>)

Query syntax

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

Example

CURRent:UNDer:PROTection:STATe ON, (@1)

[SOURce:]CURRent:UNDer:PROTection:DELay[:TIME] <value>, (@<chanlist>)

This command sets the under-current protection delay. The under-current protection function will not be triggered on the specified output channel during the delay time. After the delay time has expired, the under-current protection function will be active.

Syntax

[SOURce:]CURRent:UNDer:PROTection:DELay[:TIME] <level>,(@<chanlist>)

Query syntax

[SOURce:]CURRent:UNDer:PROTection:DELay[:TIME]? [MIN|MAX,] (@<chanlist>)

Example

CURRent:UNDer:PROTection:DELay 10, (@1)

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

This command sets the undervoltage protection warm time. The under-voltage protection function will not be triggered on the specified output channel during the warm time. even if the voltage reaches the under-voltage protection point and meets the delay time.

Syntax

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

Arguments

MAXimum|MINimum|DEFault

Query syntax

[SOURce:]CURRent:UNDer:PROTection:WARM[:TIME]? [MIN|MAX,] (@<chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

CURRent:UNDer:PROTection:WARM 10, (@1)

[SOURce:]CURRent:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

This command sets the current rise slew rate.

If the setting state of the [SOURce:]CURRent:SLEW:COUPle is ON, this command sets the current rise slew rate while simultaneously setting the current fall slew rate.

Syntax

[SOURce:]CURRent:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

Arguments

0.001-30|MINimum|MAXimum (unit: A/ms)

Query syntax

[SOURce:]CURRent:SLEW[:POSitive][:IMMediate]? [MIN|MAX,] (@<chanlist>)

Example

CURRent:SLEW 10, (@1)

[SOURce:]CURRent:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>)

This command sets the current fall slew rate.

If the setting state of the [SOURce:]CURRent:SLEW:COUPle is ON, this command sets the current fall slew rate while simultaneously setting the current rise slew rate.

Syntax

[SOURce:]CURRent:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>)

Query syntax

[SOURce:]CURRent:SLEW:NEGative[:IMMediate]? [MIN|MAX,] (@<chanlist>)

Example

CURRent:SLEW:NEGative 10, (@1)

[SOURce:]CURRent:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

This command sets and queries the tracking status of the current slope.

Syntax

[SOURce:]CURRent:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]CURRent:SLEW:COUPle? (@<chanlist>)

Example

CURRent:SLEW:COUPle ON, (@1)

[SOURce:]CURRent:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>)

Enable or disable current rise slope maximum.

Syntax

[SOURce:]CURRent:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

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

Example

CURRent:SLEW:MAXimum ON, (@1)

[SOURce:]CURRent:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

Enable or disable current drop slope maximum.

Syntax

[SOURce:]CURRent:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

Query syntax

[SOURce:]CURRent:SLEW:NEGative:MAXimum? (@<chanlist>)

Example

CURRent:SLEW:NEGative:MAXimum ON, (@1)

[SOURce:]POWer:LIMit[:POSitive][:IMMediate][:AMPLitude] <value>, (<@chanlist>)

This command sets the positive power limit of the specified output channel. If [SOURce:]POWer:LIMit:COUPle is enabled, this command also sets the value of the negative power limit.

Syntax

[SOURce:]POWer:LIMit[:POSitive][:IMMediate][:AMPLitude] <value>, (<@chanlist>)

Arguments

<value>0-maximum|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 2
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]POWer:LIMit[:POSitive][:IMMediate][:AMPLitude]? [MINimum|MAXimum,] (<@chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

POW:LIM 75,(@1,2)

[SOURce:]POWer:LIMit:NEGative[:IMMediate][:AMPLitude]<value>, (<@chanlist>)

This command sets the negative power limit of the specified output channel. If [SOURce:]POWer:LIMit:COUPle is enabled, this command also sets the value of the positive power limit.

Syntax

[SOURce:]POWer:LIMit:NEGative[:IMMediate][:AMPLitude]<value>, (<@chanlist>)

Arguments

MAXimum|MINimum|DEFault

Query syntax

[SOURce:]POWer:LIMit:NEGative[:IMMediate][:AMPLitude]? [MINimum|MAXimum,] (<@chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

POWer:LIMit:NEGative 20, (@1)

[SOURce:]POWer:LIMit:COUPle 0|OFF|1|ON, (@<chanlist>)

This command sets and queries the tracking status of the power slope.

Syntax

[SOURce:]POWer:LIMit:COUPle 0|OFF|1|ON, (@<chanlist>)

Query syntax

[SOURce:]POWer:LIMit:COUPle? (@<chanlist>)

Arguments

0|OFF|1|ON

*RST

OFF

Example

POWer:LIMit:COUPle ON, (@1)

[SOURce:]POWer:PROTection:DELay[:TIME] <value>, (@<chanlist>)

The commands set and query the delay time of overpower protection.

Syntax

[SOURce:]POWer:PROTection:DELay[:TIME] <value>, (@<chanlist>)

Arguments

0-maximum|MINimum|MAXimum

Query syntax

[SOURce:]POWer:PROTection:DELay[:TIME]? [MINimum|MAXimum,] (@<chanlist>)

Example

POWer:PROTection:DELay 20, (@1)

[SOURce:]POWer:PROTection:STATe 0|OFF|1|ON, (@<chanlist>)

This command enables or disables the over-power protection (OPP) function. The enabled state is On (1); the disabled state is Off (0). If the over-power protection function is enabled and the output is disabled.

Syntax

[SOURce:]POWer:PROTection:STATe 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

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

Example

VOLTage:PROTection:DELay 30, (@1)

[SOURce:]POWer:PROTection[:LEVel][:POSitive] <value>,(@<chanlist>)

This command sets the positive power protection value for over-power protection.

Syntax

[SOURce:]POWer:PROTection[:LEVel][:POSitive] <value>,(@<chanlist>)

Query syntax

[SOURce:]POWer:PROTection[:LEVel][:POSitive]?
[MINimum|MAXimum,](@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

Example

POWer:PROTection 100, (@1)

[SOURce:]POWer:PROTection[:LEVel]:NEGative <value>,(@<chanlist>)

This command sets the negative power protection value for over-power protection, This command only applies to bidirectional DC power supply models.

Syntax

[SOURce:]POWer:PROTection[:LEVel]:NEGative <value>,(@<chanlist>)

Query syntax

[SOURce:]POWer:PORTection[:LEVel]:NEGative?
[MINimum|MAXimum,](@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

Example

POWer:PORTection:NEGative 10, (@1)

[SOURce:]POWer:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

This command sets the power rise slew rate.

If the setting state of the [SOURce:]POWer:SLEW:COUPle is ON, this command sets the power rise slew rate while simultaneously setting the power fall slew rate.

Syntax

[SOURce:]POWer:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

[SOURce:]POWer:SLEW[:POSitive][:IMMediate]? [MIN|MAX,] (@<chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

POWer:SLEW 2 (@1)

[SOURce:]POWer:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>)

This command sets the power rise slew rate.

If the setting state of the [SOURce:]POWer:SLEW:COUPle is ON, this command sets the power fall slew rate while simultaneously setting the power rise slew rate.

Syntax

[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

Query syntax

[SOURce:]VOLTage:SLEW[:POSitive][:IMMediate]? [MIN|MAX,] (@<chanlist>)

Example

VOLTage:SLEW 10, (@1)

[SOURce:]POWer:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

This command sets and queries the tracking status of the power slope.

Syntax

[SOURce:]POWer:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]POWer:SLEW:COUPle? (@<chanlist>)

Example

POWer:SLEW:COUPle ON, (@1)

[SOURce:]POWer:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>)

This command set and query the status of the maximum slope of the power rise slope.

Syntax

[SOURce:]POWer:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]POWer:SLEW[:POSitive]:MAXimum? (@<chanlist>)

Example

POWer:SLEW:MAXimum ON, (@1)

[SOURce:]POWer:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

This command sets and queries the status of the maximum slope of the power fall slope.

Syntax

[SOURce:]POWer:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]POWer:SLEW:NEGative:MAXimum? (@<chanlist>)

Example

POWer:SLEW:NEGative:MAXimum ON, (@1)

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

The commands set and query the immediate value of the resistor under CR mode.

Syntax

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

Arguments

0-maximum|MINimum|MAXimum

Query syntax

[SOURce:]RESistance[:LEVel][:IMMediate][:AMPLitude]? [MIN|MAX,] (@<chanlist>)

初始值：

0

Return Arguments

0-maximum|MINimum|MAXimum

Example：

RESistance 50, (@1)

[SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>)

This command sets the triggered resistance level of the specified output channel. Units are in Ω.

Syntax

[SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude] <value>, (@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

Query syntax

[SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude]? [MIN|MAX,] (@<chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example：

RESistance:TRIGgered 50, (@1)

[SOURce:]RESistance:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

This command sets the resistance rise slew rate.

If the setting state of the [SOURce:]RESistance:SLEW:COUPle is ON, this command sets the resistance rise slew rate while simultaneously setting the resistance fall slew rate.

Syntax

[SOURce:]RESistance:SLEW[:POSitive][:IMMediate] <value>, (@<chanlist>)

Arguments

MAXimum|MINimum|DEFault

Query syntax

[SOURce:]RESistance:SLEW[:POSitive][:IMMediate]? [MIN|MAX,] (@<chanlist>)

Return Arguments

MAXimum|MINimum|DEFault

Example

RESistance:SLEW 1, @1

[SOURce:]RESistance:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>)

This command sets the resistance fall slew rate.

If the setting state of the [SOURce:]RESistance:SLEW:COUPle is ON, this command sets the resistance fall slew rate while simultaneously setting the resistance rise slew rate.

Syntax

[SOURce:]RESistance:SLEW:NEGative[:IMMediate] <value>, (@<chanlist>)

Query syntax

[SOURce:]RESistance:SLEW:NEGative[:IMMediate]? [MIN|MAX,] (@<chanlist>)

Example

RESistance:SLEW:NEGative 10, (@1)

[SOURce:]RESistance:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

This command sets and queries the tracking status of the resistance slope.

Syntax

[SOURce:]RESistance:SLEW:COUPle 0|OFF|1|ON, (@<chanlist>)

Query syntax

[SOURce:]RESistance:SLEW:COUPle? (@<chanlist>)

Example

RESistance:SLEW:COUPle ON, (@1)

[SOURce:]RESistance:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>)

This command set and query the status of the maximum slope of the resistance rise slope.

Syntax

[SOURce:]RESistance:SLEW[:POSitive]:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]RESistance:SLEW[:POSitive]:MAXimum? (@<chanlist>)

Example

RESistance:SLEW:MAXimum ON, (@1)

[SOURce:]RESistance:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

This command sets and queries the status of the maximum slope of the resistance fall slope.

Syntax

[SOURce:]RESistance:SLEW:NEGative:MAXimum 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

[SOURce:]RESistance:SLEW:NEGative:MAXimum? (@<chanlist>)

Example

RESistance:SLEW:NEGative:MAXimum ON, (@1)

Copyright ©ITECH Electronic Co., Ltd.

6.5 OUTPut Subsystem

INPut|OUTPut[:STATe] 0|OFF|1|ON , (@<chanlist>)

This command is used to control the output state of the instrument.

Syntax

INPut|OUTPut[:STATe] 0|OFF|1|ON , (@<chanlist>)

Arguments

0|OFF|1|ON

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut[:STATe]? (@<chanlist>)

Return Arguments

0|1

Example

OUTPut ON , (@1)

INPut|OUTPut[:STATe]:COUPle:CHANnel ALL|NONE|[<channel> {,<channel>}]

Sets and queries the channels for output coupling.

Syntax

INPut|OUTPut[:STATe]:COUPle:CHANnel ALL|NONE|[<channel> {,<channel>}]

Arguments

ALL|NONE|<1,2,3,4,5,6,7,8>

Query syntax

INPut|OUTPut[:STATe]:COUPle:CHANnel?

Example

OUTP:COUP:CHAN 1,2,4

INPut|OUTPut[:STATe]:COUPle:DOFFset <value>

This instruction is used to set and query the delay offset.

Syntax

INPut|OUTPut[:STATe]:COUPle:DOFFset <value>

Arguments

MIN|MAX|

Query syntax

INPut|OUTPut[:STATe]:COUPle:DOFFset? [MIN|MAX]

Return Arguments

MIN|MAX

Example

OUTPut:COUPle:DOFFset 10

INPut|OUTPut[:STATe]:COUPle:DOFFset:MODE AUTO|MANual

Set software watchdog timeout, unit is s.

Syntax

INPut|OUTPut[:STATe]:COUPle:DOFFset:MODE AUTO|MANual

Arguments

AUTO|MANual

Query syntax

INPut|OUTPut[:STATe]:COUPle:DOFFset:MODE?

Return Arguments

AUTO|MANual

Example

OUTPut:COUPle:DOFF:MODE AUTO

INPut|OUTPut[:STATe]:COUPle:MAX:DOFFset?

Sets the maximum delay offset for querying the machine.

Syntax

INPut|OUTPut[:STATe]:COUPle:MAX:DOFFset?

Return Arguments

NRf

Example

OUTPut:COUPle:MAX:DOFFset?

INPut|OUTPut[:STATe]:DELay:FALL <value>, (@<chanlist>)

Sets the delay time for output shutdown.

Syntax

INPut|OUTPut[:STATe]:DELay:FALL <value>, (@<chanlist>)

Arguments

<value>MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut[:STATe]:DELay:FALL? [MIN|MAX,] (@<chanlist>)

Return Arguments

MINimum|MAXimum

Example

OUTPut:DELay:FALL 10, (@1)

INPut|OUTPut[:STATe]:DELay:RISE <value>, (@<chanlist>)

Sets the delay time for the output to turn on.

Syntax

INPut|OUTPut[:STATe]:DELay:RISE <value>, (@<chanlist>)

Arguments

<value>MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut[:STATe]:DELay:RISE? [MIN|MAX,] (@<chanlist>)

Return Arguments

MINimum|MAXimum

Example

OUTPut:DELay:RISE 10, (@1)

INPut|OUTPut:INHibit:COUPle:CHANnel ALL|NONE|[<channel>{,<channel>}]

Sets and queries the channels for output suppression coupling.

Syntax

INPut|OUTPut:INHibit:COUPle:CHANnel ALL|NONE|[<channel> {,<channel>}]

Arguments

ALL|NONE|[<channel> {,<channel>}]

Query syntax

INPut|OUTPut:INHibit:COUPle:CHANnel?

Example

OUTPut:INHibit:COUPle:CHANnel ALL

INPut|OUTPut:INHibit:MODE LATChing|LIVE|OFF

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

Syntax

INPut|OUTPut:INHibit:MODE LATChing|LIVE|OFF

Arguments

LATChing|LIVE|OFF

Query syntax

INPut|OUTPut:INHibit:MODE?

Return Arguments

LATChing|LIVE|OFF

Example

OUTPut:INHibit:MODE LATChing

INPut|OUTPut:PON:STATe RST|LAST|LOFF

Setting and querying the status of the power-up. RST means restoring the factory value, LAST means the status before the last power-off, and LOFF means LAST+OFF.

Syntax

INPut|OUTPut:PON:STATe RST|LAST|LOFF

Arguments

RST|LAST|LOFF

Query syntax

INPut|OUTPut:PON:STATe?

Example

OUTPut:PON:STATe RST

INPut|OUTPut:PROTection:CLEar (@<chanlist>)

This command is used to clear the protection status.

Syntax

INPut|OUTPut:PROTection:CLEar (@<chanlist>)

Example

OUTPut:PROT:CLE (@1)

INPut|OUTPut:PROTection:COUPle 0|OFF|1|ON

This command is used to set enable or disable protection coupling.

Syntax

INPut|OUTPut:PROTection:COUPle 0|OFF|1|ON

Arguments

0|OFF|1|ON

Query syntax

INPut|OUTPut:PROTection:COUPle?

Return Arguments

0|OFF|1|ON

Example

OUTP:PROT:COUPle ON

INPut|OUTPut:PROTection:COUPle:CHANnel ALL|NONE|[<channel>{,<channel>}]

The commands set and query the channel for protection coupling.

Syntax

INPut|OUTPut:PROTection:COUPle:CHANnel ALL|NONE|[<channel>{,<channel>}]

Arguments

ALL|NONE|[<channel> {,<channel>}]

Query syntax

INPut|OUTPut:PROTection:COUPle:CHANnel?

Return Arguments

ALL|NONE|[<channel> {,<channel>}]

Example

OUTP:PROT:COUP:CHAN 1,2,3

INPut|OUTPut:PROTection:WDOG[:STATe] 0|OFF|1|ON

The commands set and query the enable state of the software watchdog.

Syntax

INPut|OUTPut:PROTection:WDOG[:STATe] 0|OFF|1|ON

Arguments

0|OFF|1|ON

Query syntax

INPut|OUTPut:PROTection:WDOG[:STATe]?

Return Arguments

0|1

Example

OUTP:PROT:WDOG ON

INPut|OUTPut:PROTection:WDOG:DELay <value>

This command sets the delay time of the software watchdog.

Syntax

INPut|OUTPut:PROTection:WDOG:DELay <value>

Arguments

[MIN|MAX]

Query syntax

INPut|OUTPut:PROTection:WDOG:DELay? [MIN|MAX]

Return Arguments

[MIN|MAX]

Example

INPut|OUTPut:PROTection:WDOG:DELay 10

INPut|OUTPut:PROTection:FOLDback[:MODE] <OFF|CC|CV>,(@<chanlist>)

The commands set and query the mode of foldback protection.

Syntax

INPut|OUTPut:PROTection:FOLDback[:MODE] <OFF|CC|CV>,(@<chanlist>)

Arguments

OFF|CC|CV

<chanlist>: Channel number.

Copyright ©ITECH Electronic Co., Ltd.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut:PROTection:FOLDback[:MODE]? (@<chanlist>)

Return Arguments

OFF|CC|CV

Example

OUTPut:PROTection:FOLDback CC,(@1)

INPut|OUTPut:PROTection:FOLDback:DELay <value>,(@<chanlist>)

The commands set and query the foldback protection delay time.

Syntax

INPut|OUTPut:PROTection:FOLDback:DELay <value>,(@<chanlist>)

Arguments

<value>MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut:PROTection:FOLDback:DELay? (@<chanlist>)

Return Arguments

[MIN|MAX]

Example

OUTPut:PROTection:FOLDback:DELay 10,(@1)

INPut|OUTPut:RELay:LOCK[:STATe] 0|1|OFF|ON,(@<chanlist>)

This command enables or disables the latched-off state of the output relay.

Syntax

INPut|OUTPut:RELay:LOCK[:STATe] 0|1|OFF|ON,(@<chanlist>)

Arguments

0|1|OFF|ON

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Copyright ©ITECH Electronic Co., Ltd.

Query syntax

INPut|OUTPut:RELay:LOCK[:STATe]? (@<chanlist>)

Return Arguments

0|1|OFF|ON

Example

OUTPut:RELay:LOCK ON,(@1)

INPut|OUTPut:REGulation:SPEed <LOW|HIGH1|HIGH2|HIGH3> (@<chanlist>)

The commands set and query the regulation speed.

Syntax

INPut|OUTPut:REGulation:SPEed <LOW|HIGH1|HIGH2|HIGH3> (@<chanlist>)

Arguments

LOW|HIGH1|HIGH2|HIGH3

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut:REGulation:SPEed? <LOW|HIGH1|HIGH2|HIGH3> (@<chanlist>)

Return Arguments

LOW|HIGH1|HIGH2|HIGH3

Example

OUTPut:REGulation:SPEed LOW (@1)

INPut|OUTPut:FUNCtion:MODE <FIXed|ARB|BCHarge|BEMulation|TRANsient|OCP|OPP|SEQuence|CDARB|SSWeep|LIST|BDIScharge>,(@<chanlist>)

This command sets the function mode of the inputs and outputs.

Syntax

INPut|OUTPut:FUNCtion:MODE <FIXed|ARB|BCHarge|BEMulation|TRANsient|OCP|OPP|SEQuence|CDARB|SSWeep|LIST|BDIScharge>,(@<chanlist>)

Arguments

<FIXed|ARB|BCHarge|BEMulation|TRANsient|OCP|OPP|SEQuence|CDARB|SSWeep|LIST|BDIScharge>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut:FUNCtion:MODE? (@<chanlist>)

Example

OUTPut:FUNCtion:MODE FIXed,(@1)

INPut|OUTPut:OFF:VOLTage:RTZero <0||OFF|1|ON>,(@<chanlist>)

The commands set and query the state of zeroing the voltage when output OFF .

Syntax

INPut|OUTPut:OFF:VOLTage:RTZero <0||OFF|1|ON>,(@<chanlist>)

Arguments

<0||OFF|1|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

INPut|OUTPut:OFF:VOLTage:RTZero? (@<chanlist>)

Example

OUTPut:OFF:VOLTage:RTZero ON,(@1)

INPut|OUTPut:SHORt[:STATe] 0|OFF|1|ON, (@<chanlist>)

The commands set and query the status of the short-circuit simulation.

Syntax

INPut|OUTPut:SHORt[:STATe] 0|OFF|1|ON, (@<chanlist>)

Arguments

0|OFF|1|ON

Query syntax

INPut|OUTPut:SHORt[:STATe]? (@<chanlist>)

Return Arguments

0|OFF|1|ON

Example

OUTPut:SHORt ON, (@1)

INPut|OUTPut:REGulation:SPEed <LOW|HIGH1|HIGH2|HIGH3> (@<chanlist>)

The commands set and query the regulation speed.

Syntax

INPut|OUTPut:REGulation:SPEed <LOW|HIGH1|HIGH2|HIGH3> (@<chanlist>)

Arguments

LOW|HIGH1|HIGH2|HIGH3

Query syntax

INPut|OUTPut:REGulation:SPEed? <LOW|HIGH1|HIGH2|HIGH3> (@<chanlist>)

Return Arguments

LOW|HIGH1|HIGH2|HIGH3

Example

OUTPut:REGulation:SPEed LOW (@1)

INPut|OUTPut:FUNCtion:MODE <FIXed|ARB|BCHarge|BEMulation|TRANsient|OCP|OPP|SEQuence|CDARB|SSWeep|LIST|BDIScharge>,(@<chanlist>)

This command sets the function mode of the inputs.

Syntax

INPut|OUTPut:FUNCtion:MODE <FIXed|ARB|BCHarge|BEMulation|TRANsient|OCP|OPP|SEQuence|CDARB|SSWeep|LIST|BDIScharge>,(@<chanlist>)

Arguments

<FIXed|ARB|BCHarge|BEMulation|TRANsient|OCP|OPP|SEQuence|CDARB|SSWeep|LIST|BDIScharge>

Query syntax

INPut|OUTPut:FUNCtion:MODE? (@<chanlist>)

Example

OUTPut:FUNCtion:MODE FIXed,(@1)

Copyright ©ITECH Electronic Co., Ltd.

6.6 FETCh & MEASureSubsystem

FETCh[:SCALar]:CURRent[:DC]? (@<chanlist>)

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

This command is used to read the average value of current.

Syntax

FETCh[:SCALar]:CURRent? (@<chanlist>)

MEASure[:SCALar]:CURRent[:AC]? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETC:CURR? (@1)

MEAS:CURR? (@1)

FETCh[:SCALar]:CURRent:ACDC? (@<chanlist>)

MEASure[:SCALar]:CURRent:ACDC? (@<chanlist>)

This command is used to read the current RMS value.

Syntax

FETCh[:SCALar]:CURRent:ACDC? (@<chanlist>)

MEASure[:SCALar]:CURRent:ACDC? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>…

Example

FETC:CURR:ACDC? (@1)

MEAS:CURR:ACDC? (@1)

FETCh[:SCALar]:CURRent:MAXimum? (@<chanlist>)

MEASure[:SCALar]:CURRent:MAXimum? (@<chanlist>)

This command is used to read the current maximum value.

Syntax

FETCh[:SCALar]:CURRent:MAXimum? (@<chanlist>)

MEASure[:SCALar]:CURRent:MAXimum? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:CURRent:MAX? (@1)

MEASure:CURRent:MAX? (@1)

FETCh[:SCALar]:CURRent:MINimum? (@<chanlist>)

MEASure[:SCALar]:CURRent:MINimum? (@<chanlist>)

This command is used to read the current min.

Syntax

FETCh[:SCALar]:CURRent:MINimum? (@<chanlist>)

MEASure[:SCALar]:CURRent:MINimum? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:CURR:MIN? (@1)

MEAS:CURRent:MIN? (@1)

FETCh[:SCALar]:POWer[:DC]? (@<chanlist>)

MEASure[:SCALar]:POWer[:DC]? (@<chanlist>)

This command is used to read the power average.

Syntax

FETCh[:SCALar]:POWer[:DC]? (@<chanlist>)

MEASure[:SCALar]:POWer[:DC]? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:POWer? (@1)

MEASure:POWer? (@1)

FETCh[:SCALar]:POWer:MAXimum? (@<chanlist>)

MEASure[:SCALar]:POWer:MAXimum? (@<chanlist>)

This command is used to read the maximum value of the power value.

Syntax

FETCh[:SCALar]:POWer:MAXimum? (@<chanlist>)

MEASure[:SCALar]:POWer:MAXimum? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>…

Example

FETCh:POWer:MAX? (@1)

MEASure:POWer:MAX? (@1)

FETCh[:SCALar]:POWer:MINimum? (@<chanlist>)

MEASure[:SCALar]:POWer:MINimum? (@<chanlist>)

This command is used to read the power min.

Syntax

FETCh[:SCALar]:POWer:MINimum? (@<chanlist>)

MEASure[:SCALar]:POWer:MINimum? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:POWer:MIN? (@1)

MEAS:POWer:MIN? (@1)

FETCh[:SCALar]:VOLTage[:DC]? (@<chanlist>)

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

This command is used to read the average value of voltage.

Syntax

FETCh[:SCALar]:VOLTage[:DC]? (@<chanlist>)

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

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>…

Example

FETC:VOLT? (@1)

MEAS:VOLT? (@1)

FETCh[:SCALar]:VOLTage:ACDC? (@<chanlist>)

MEASure[:SCALar]:VOLTage:ACDC? (@<chanlist>)

This command is used to read the voltage RMS value.

Syntax

FETCh[:SCALar]:VOLTage:ACDC? (@<chanlist>)

MEASure[:SCALar]:VOLTage:ACDC? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETC:VOLT:ACDC? (@1)

MEAS:VOLT:ACDC? (@1)

FETCh[:SCALar]:VOLTage:MINimum? (@<chanlist>)

MEASure[:SCALar]:VOLTage:MINimum? (@<chanlist>)

This command is used to read the voltage min.

Syntax

FETCh[:SCALar]:VOLTage:MINimum? (@<chanlist>)

MEASure[:SCALar]:VOLTage:MINimum? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETC:VOLT:MIN? (@1)

MEAS:VOLT:MIN? (@1)

FETCh[:SCALar]:VOLTage:MAXimum? (@<chanlist>)

MEASure[:SCALar]:VOLTage:MAXimum? (@<chanlist>)

This command is used to read the maximum value of voltage.

Syntax

FETCh[:SCALar]:VOLTage:MAXimum? (@<chanlist>)

MEASure[:SCALar]:VOLTage:MAXimum? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Arguments

<NRf>

Example

FETC:VOLTage:MAX? (@1)

MEAS:VOLTage:MAX? (@1)

FETCh[:SCALar]:AHOur? (@<chanlist>)

MEASure[:SCALar]:AHOur? (@<chanlist>)

This command is used to read the channel's corresponding amp-hours data.

Syntax

FETCh[:SCALar]:AHOur? (@<chanlist>)

MEASure[:SCALar]:AHOur? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:AHOur? (@1)

Copyright ©ITECH Electronic Co., Ltd.

FETCh[:SCALar]:WHOur? (@<chanlist>)

MEASure[:SCALar]:WHOur? (@<chanlist>)

This command is used to read the watt-hour data of the channel.

Syntax

FETCh[:SCALar]:WHOur? (@<chanlist>)

MEASure[:SCALar]:WHOur? (@<chanlist>)

Return Arguments

<NRf>

Example

FETC:WHOur? (@1)

MEAS:WHOur? (@1)

FETCh[:SCALar]:TIME? (@<chanlist>)

MEASure[:SCALar]:TIME? (@<chanlist>)

This command is used to read the load time.

Syntax

FETCh[:SCALar]:TIME? (@<chanlist>)

MEASure[:SCALar]:TIME? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETC:TIME? (@1)

MEAS:TIME? (@1)

FTECh[:SCALar][:ALL]? (@<chanlist>)

MEASure[:SCALar][:ALL]? (@<chanlist>)

This command is used to read all measurement data at once.

(CURR_DC,VOLT_DC,POW_DC,CURR_ACDC,VOLT_ACDC,CURR_MAX,VOLT_MAX,POW_MAX,CURR_MIN,VOLT_MIN,POW_MIN,AH,WH,TIME)

Syntax

FTECh[:SCALar][:ALL]? (@<chanlist>)

MEASure[:SCALar][:ALL]? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FTEC? (@1)

MEAS? (@1)

FETCh:ARRay:CURRent[:DC]? (@<chanlist>)

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

This query returns an array containing the digitized output current in amperes. The data returned by the FETCh command is the result of the last measurement command or triggered acquisition.

Syntax

FETCh:ARRay:CURRent[:DC]? (@<chanlist>)

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

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4

Return Arguments

<NRf>

Example

FETCh:ARRay:CURRent? (@1)

MEASure:ARRay:CURRent? (@1)

FETCh:ARRay:VOLTage[:DC]? (@<chanlist>)

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

This query returns an array containing the digitized output voltage in volts. The data returned by the FETCh command is the result of the last measurement

command or triggered acquisition.

Syntax

FETCh:ARRay:VOLTage[:DC]? (@<chanlist>)

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

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:ARRay:VOLTage? (@1)

MEASure:ARRay:VOLTage? (@1)

FETCh:ARRay:POWer[:DC]? (@<chanlist>)

MEASure:ARRay:POWer[:DC]? (@<chanlist>)

This query returns an array containing the digitized output power in watts. The data returned by the FETCh command is the result of the last measurement command or triggered acquisition.

Syntax

FETCh:ARRay:POWer[:DC]? (@<chanlist>)

MEASure:ARRay:POWer[:DC]? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Example

FETCh:ARRay:POWer? (@1)

MEASure:ARRay:POWer? (@1)

6.7 SENSe Subsystem

SENSe[:REMote][:STATe] 0|1|OFF|ON, (@<channlist>)

Set and query the status of remote sensing.

Syntax

FETCh:ARRay:POWer[:DC]? (@<chanlist>)

MEASure:ARRay:POWer[:DC]? (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Query syntax

SENSe[:REMote][:STATe]? (@<channlist>)

Example

SENS ON,(@1)

SENSe:SWEep:OFFSet:POINts <value>, (@<chanlist>)

This command defines the offset in a data sweep when the measurement is triggered. Programmed values can range from –(SENS:SWE:POIN setting –1) through 2,000,000,000 (2E9). Positive values represent the delay after the trigger occurs but before the samples are acquired. Negative values represent data samples taken prior to the trigger. The maximum negative value that can be programmed is one less than the maximum allowable SENSe:SWEep:POINts value.

Syntax

SENSe:SWEep:OFFSet:POINts <value>, (@<chanlist>)

Arguments

<Value>[MINimum|MAXimum]

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NRf>

Query syntax

SENSe:SWEep:OFFSet:POINts? [MINimum|MAXimum,] (@<chanlist>)

Example

SENS:SWE:OFFS:POIN -1024,(@1)

SENSe:SWEep:POINts <value>, (@<chanlist>)

This command defines the number of points in a measurement. The number of samples (points) that can be specified depends on the number of measurement parameters selected. You can measure up to eight parameters (voltage and current X 4 outputs).

Syntax

SENSe:SWEep:POINts <value>, (@<chanlist>)

Arguments

<Value>1 to 16384|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

[MINimum|MAXimum]

Query syntax

SENSe:SWEep:POINts? [MINimum|MAXimum,] (@<chanlist>)

Example

SENS:SWE:POIN 1024,(@1)

SENSe:SWEep:TINTerval <value>, (@<chanlist>)

This command defines the time period between samples in seconds. Programmed values can range from 5 us to 2100 seconds. Note that the shortest time interval (fastest speed) that can be specified depends on the number of parameters that are being measured and the model that is doing the measuring.

Syntax

SENSe:SWEep:TINTerval <value>, (@<chanlist>)

Arguments

<Value>0.000005-2100.0|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

[MINimum|MAXimum]

Query syntax

SENSe:SWEep:TINTerval? [MINimum|MAXimum,] (@<chanlist>)

Example

SENS:SWE:TINT 0.002,(@1)

SENSe:NPLCycles? [MINimum|MAXimum,](@<chanlist>)

This command queries the number of measured power frequency cycles.

Syntax

SENSe:NPLCycles? [MINimum|MAXimum,](@<chanlist>)

Arguments

<Value>0.000005-2100.0|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

10 (50HZ)

12 (60HZ)

Example

SENS:NPLC? (@1)

SENSe:APERture <time|MINimum|MAXimum>,(@<chanlist>)

The commands set and query the measured aperture time.

Syntax

SENSe:APERture <time|MINimum|MAXimum>,(@<chanlist>)

Arguments

<Value>0.02-34406400|MINimum|MAXimum

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

[MINimum|MAXimum]

Query syntax

SENSe:APERture? [MINimum|MAXimum,] (@<chanlist>)

Example

SENS:APER 0.1,(@1)

SENSe:MEASure:TIME:MODE <NPLC|APERture>,(@<chanlist>)

Set and query the time mode of the measurement.

Syntax

SENSe:MEASure:TIME:MODE <NPLC|APERture>,(@<chanlist>)

Arguments

<NPLC|APERture>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Return Arguments

<NPLC|APERture>

Query syntax

SENSe:MEASure:TIME:MODE?(@<chanlist>)

Example

SENS:MEAS:TIME:MODE NPLC, (@1)

SENSe:AHOur:CLEar (@<chanlist>)

This instruction is used to clear the amp-hours data.

Syntax

SENSe:AHOur:CLEar (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Example

SENS:AHO:CLE (@1)

SENSe:WHOur:CLEar (@<chanlist>)

This instruction is used to clear the watt-hour data.

Syntax

SENSe:AHOur:CLEar (@<chanlist>)

Arguments

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Example

SENS:AHO:CLE (@1)

SENSe:AHOur:CLEar:AUTO <0|1|OFF|ON>,(@<chanlist>)

This instruction is used to set up the automatic clearing of the amp-hours data.

Syntax

SENSe:AHOur:CLEar (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:AHOur:CLEar:AUTO? (@<chanlist>)

Example

SENS:AHO:CLE:AUTO 1,(@1)

SENSe:WHOur:CLEar:AUTO <0|1|OFF|ON>,(@<chanlist>)

This instruction is used to set up the automatic clearing of the watt-hour data.

Syntax

SENSe:WHOur:CLEar:AUTO <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:WHOur:CLEar:AUTO? (@<chanlist>)

Example

SENS:WHO:CLE:AUTO ON,(@1)

SENSe:DLOG:FUNCtion:CURRent 0|OFF|1|ON, (@<chanlist>)

This command is used to set and query the current measurement status of the internal recorder.

Syntax

SENSe:DLOG:FUNCtion:CURRent 0|OFF|1|ON, (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:DLOG:FUNCtion:CURRent? (@<chanlist>)

Example

SENSe:DLOG:FUNCtion:CURRent ON,(@1)

SENSe:DLOG:FUNCtion:VOLTage 0|OFF|1|ON, (@<chanlist>)

This command is used to set and query the voltage measurement status of the internal recorder.

Syntax

SENSe:DLOG:FUNCtion:VOLTage 0|OFF|1|ON, (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:DLOG:FUNCtion:VOLTage? (@<chanlist>)

Example

SENSe:DLOG:FUNCtion:CURRent ON,(@1)

SENSe:DLOG:FUNCtion:MINMax 0|OFF|1|ON

This command is used to set and query the maximum and minimum value measurement function of the internal recorder.

Syntax

SENSe:DLOG:FUNCtion:MINMax 0|OFF|1|ON

Arguments

<0|1|OFF|ON>

Query syntax

SENSe:DLOG:FUNCtion:MINMax?

Example

SENSe:DLOG:FUNCtion:MINMax ON

SENSe:DLOG:MARKer1:POINt <value>

This command is used to set and query the start position and end position of data analysis.

Syntax

SENSe:DLOG:MARKer1:POINt <value>

Arguments

<value|MINimum|MAXimum> (unit is s)

MINimum0s

MAXimum: setting value

Query syntax

SENSe:DLOG:MARKer1:POINt?

Example

SENSe:DLOG:MARKer1:POINt 20

SENSe:DLOG:MARKer2:POINt <value>

This command is used to set and query the start position and end position of data analysis.

Syntax

SENSe:DLOG:MARKer2:POINt <value>

Arguments

<value|MINimum|MAXimum> (unit is s)

MINimum: 0s

MAXimum: setting value

Query syntax

SENSe:DLOG:MARKer2:POINt?

Example

SENSe:DLOG:MARKer2:POINt 20

SENSe:DLOG:OFFSet <value>

This command is used to set and query the trigger offset time percentage of the data logger.

Syntax

SENSe:DLOG:OFFSet <value>

Arguments

<0.00-100.00|MINimum|MAXimum> (unit is %)

MINimum0.00

MAXimum100.00

Query syntax

SENSe:DLOG:OFFSet?

Example

SENSe:DLOG:OFFSet 20

SENSe:DLOG:PERiod <value>

This command is used to set and query the integration time of the data logger.

Syntax

SENSe:DLOG:PERiod <value>

Arguments

<value|MINimum|MAXimum> (unit is s)

MINimum0.00002s

MAXimum60s

Query syntax

SENSe:DLOG:PERiod? [MINimum|MAXimum]

Example

SENSe:DLOG:PERiod 5

SENSe:DLOG:TIME <value>

This command is used to set and query the time of the data logger.

Syntax

SENSe:DLOG:TIME <value>

Arguments

<value|MINimum|MAXimum> (unit is s)

MINimum0s

MAXimum86400s

Query syntax

SENSe:DLOG:TIME? [MINimum|MAXimum]

Example

SENSe:DLOG:TIME 50

SENSe:DLOG:TINTerval <value>

This command is used to set and query the time interval of the data logger.

Syntax

SENSe:DLOG:TINTerval <value>

Arguments

<value|MINimum|MAXimum> (unit is s)

MINimum0.00002s

MAXimum60s

Query syntax

SENSe:DLOG:TINTerval? [MINimum|MAXimum]

Example

SENSe:DLOG:TINTerval 10

SENSe:ELOG:FUNCtion:CURRent 0|OFF|1|ON, (@<chanlist>)

This command is used to set and query the current measurement status of the internal recorder.

Syntax

SENSe:ELOG:FUNCtion:CURRent 0|OFF|1|ON, (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:ELOG:FUNCtion:CURRent? (@<chanlist>)

Example

SENSe:ELOG:FUNCtion:CURRent ON,(@1)

SENSe:ELOG:FUNCtion:VOLTage 0|OFF|1|ON, (@<chanlist>)

This command is used to set and query the voltage measurement status of the internal recorder.

Syntax

SENSe:ELOG:FUNCtion:VOLTage 0|OFF|1|ON, (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:ELOG:FUNCtion:VOLTage? (@<chanlist>)

Example

SENSe:ELOG:FUNCtion:CURRent ON,(@1)

SENSe:ELOG:FUNCtion:CURRent:MINMax 0|OFF|1|ON, (@<chanlist>)

This command is used to set and query the current maximum and minimum measurement function of the internal recorder.

Syntax

SENSe:ELOG:FUNCtion:CURRent:MINMax 0|OFF|1|ON, (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:ELOG:FUNCtion:CURRent:MINMax? (@<chanlist>)

Example

SENSe:ELOG:CURRent:FUNCtion:MINMax ON,(@1)

SENSe:ELOG:FUNCtion:VOLTage:MINMax 0|OFF|1|ON, (@<chanlist>)

This command is used to set and query the current maximum and minimum measurement function of the internal recorder.

Syntax

SENSe:ELOG:FUNCtion:VOLTage:MINMax 0|OFF|1|ON, (@<chanlist>)

Arguments

<0|1|OFF|ON>

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:ELOG:FUNCtion:VOLTage:MINMax? (@<chanlist>)

Example

SENSe:ELOG:CURRent:FUNCtion:MINMax ON,(@1)

SENSe:ELOG:PERiod <value>, (@<chanlist>)

This command is used to set and query the integration time of the data logger.

Syntax

SENSe:ELOG:PERiod <value>, (@<chanlist>)

Arguments

<value|MINimum|MAXimum>(unit is s)

MINimum0.0001s

MAXimum1s

<chanlist>: Channel number.

(@1) – Channel 1
(@1,4) – channels 1 and 4
(@1:4) – channels 1 through 4.

Query syntax

SENSe:ELOG:PERiod? [MINimum|MAXimum,] (@<chanlist>)

Example

SENSe:ELOG:PERiod 0.005,(@1)

6.8 List Subsystem

[SOURce:]LIST:COUNt <value>, (@<chanlist>)

This command is used to set and query the number of cycles that the list runs.

Syntax

[SOURce:]LIST:COUNt <value>, (@<chanlist>)

Arguments

0-65535|MINimum|MAXimum

Query syntax

[SOURce:]LIST:COUNt? (@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:COUN 10,(@1)

[SOURce:]LIST:CURRent:STEP[:LEVel] <value>,{<step>,(@<chanlist>)

The commands set and query the current value of the specified step of LIST. (Unit is A)

Syntax

[SOURce:]LIST:CURRent:STEP[:LEVel] <value>,{<step>,(@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:CURRent:STEP[:LEVel]?
[MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:CURRent 1.5,1,(@1)

[SOURce:]LIST:CURRent:SLEW <value>,<step>, (@<chanlist>)

The commands set and query the current slope of the specified step of LIST. (Unit is A)

Syntax

[SOURce:]LIST:CURRent:SLEW <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:CURRent:SLEW? [MINimum|MAXimum,]<step>(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:CURRent:SLEW 2,1,(@1)

[SOURce:]LIST:CURRent:POINts <value>,(@<chanlist>)

The commands set and query the number of current list points.

Syntax

[SOURce:]LIST:CURRent:POINts <value>,(@<chanlist>)

Arguments

1-2000|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:CURRent:POINts? (@<chanlist>)

Example

LIST:CURRent:POINts 1,(@1)

[SOURce:]LIST:VOLTage:STEP[:LEVel] <value>,<step>, (@<chanlist>)

The commands set and query the voltage value of the specified step of LIST. (Unit is V)

Syntax

[SOURce:]LIST:VOLTage:STEP[:LEVel] <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:VOLTage:STEP[:LEVel]?
[MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:VOLTage 10,1,(@1)

[SOURce:]LIST:VOLTage:SLEW <value>,<step>, (@<chanlist>)

The commands set and query the voltage slope of the specified step of LIST. (Unit is V)

Syntax

[SOURce:]LIST:VOLTage:SLEW <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:VOLTage:SLEW? [MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:VOLTage:SLEW? 1,(@1)

[SOURce:]LIST:VOLTage:POINts <value>,(@<chanlist>)

The commands set and query the number of voltage list points.

Syntax

[SOURce:]LIST:VOLTage:POINts <value>,(@<chanlist>)

Arguments

1-2000|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:VOLTage:POINts? (@<chanlist>)

Example

LIST:VOLTage:POINts 1,(@1)

[SOURce:]LIST:POWer:STEP[:LEVel] <value>,<step>, (@<chanlist>)

The commands set and query the power value of the specified step of LIST. (Unit is W)

Syntax

[SOURce:]LIST:POWer:STEP[:LEVel] <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:POWer:STEP[:LEVel]?
[MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:POWer 100,1,(@1)

[SOURce:]LIST:POWer:SLEW <value>,<step>, (@<chanlist>)

The commands set and query the power slope of the specified step of LIST. (Unit is W)

Syntax

[SOURce:]LIST:POWer:SLEW <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:POWer:SLEW? [MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:POWer:SLEW 100,1,(@1)

[SOURce:]LIST:POWer:POINts <value>,(@<chanlist>)

The commands set and query the number of current list points.

Syntax

[SOURce:]LIST:POWer:POINts <value>,(@<chanlist>)

Arguments

1-2000|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:POWer:POINts? (@<chanlist>)

Example

LIST:POWer:POINts 1,(@1)

[SOURce:]LIST:RESistance:STEP[:LEVel] <value>,<step>, (@<chanlist>)

The commands set and query the resistance value of the specified step of LIST. (Unit is Ω)

Syntax

[SOURce:]LIST:RESistance:STEP[:LEVel] <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:RESistance:STEP[:LEVel]?
[MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:RESistance 10,1,(@1)

[SOURce:]LIST:RESistance:SLEW <value>,<step>, (@<chanlist>)

The commands set and query the resistance value of the specified step of LIST. (Unit is Ω)

Syntax

[SOURce:]LIST:RESistance:SLEW <value>,<step>, (@<chanlist>)

Arguments：

MINimum-MAXimum|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:RESistance:SLEW? [MINimum|MAXimum,]<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:RESistance:SLEW? 1,(@1)

[SOURce:]LIST:RESistance:POINts <value>,(@<chanlist>)

The commands set and query the number of current list points.

Syntax

[SOURce:]LIST:RESistance:POINts <value>,(@<chanlist>)

Arguments

1-2000|MINimum|MAXimum

Query Syntax

[SOURce:]LIST:RESistance:POINts? (@<chanlist>)

Example

LIST:RESistance:POINts 1,(@1)

[SOURce:]LIST:TERMinate:MODE <NORMal|LAST|OFF>,(@<chanlist>)

This command is used to set and query the way to end the list operation. There are three ways to end:

NORMal: after the list runs, it will automatically return to normal.

LAST: keep the output of the last step after the list runs.

OFF: turn off the output after the list operation is over and still in the list mode.

Syntax

[SOURce:]LIST:TERMinate:MODE <NORMal|LAST|OFF>,(@<chanlist>)

Arguments

NORMal|LAST|OFF

Query Syntax

[SOURce:]LIST:TERMinate:MODE? (@<chanlist>)

Returns

NORMal|LAST|OFF

*RST

OFF

Example

LIST:TERM NORM,(@1)

[SOURce:]LIST:DWELl <value>,<step>, (@<chanlist>)

This command is used to set and query the dwell time of LIST for the specified number of steps.

Syntax

[SOURce:]LIST:DWELl <value>,<step>, (@<chanlist>)

Arguments

0.001-10737|MINimum|MAXimum

Query syntax

[SOURce:]LIST:DWELl? [MINimum|MAXimum,],<step>,(@<chanlist>)

Returns

[MINimum|MAXimum]

Example

LIST:DWELl 2,1,(@1)

[SOURce:]LIST:TOUTput:BOSTep:POINts? (@<chanlist>)

This query returns the number of points (steps) in the beginning of the step trigger list (BOSTep), not the point values.

Syntax

[SOURce:]LIST:TOUTput:BOSTep:POINts? (@<chanlist>)

Example

LIST:TOUTput:BOSTep:POINts? (@1)

[SOURce:]LIST:TOUTput:EOSTep:POINts? (@<chanlist>)

This query returns the number of points (steps) in the end of the step trigger list (EOSTep), not the point values.

Syntax

[SOURce:]LIST:TOUTput:EOSTep:POINts? (@<chanlist>)

Example

LIST:TOUTput:EOSTep:POINts? (@1)

[SOURce:]LIST:STEP ONCE|AUTO,<step>(@<chanlist>)

This command is used to Set and query step jump method, ONCE need to wait for the trigger, AUTO automatic jump.

Syntax

[SOURce:]LIST:STEP ONCE|AUTO,<step>(@<chanlist>)

Arguments

ONCE|AUTO

Query Syntax

[SOURce:]LIST:STEP? <step>,(@<chanlist>)

Example

LIST:STEP AUTO,1,(@1)

[SOURce:]LIST:TOUTput:BOSTep[:DATA] 0|OFF|1|ON,<step>,(@<chanlist>)

This command specifies which list steps generate a trigger signal at the beginning of the step (BOSTep).

A trigger is only generated when the state is set to ON. The trigger signal can be used as a trigger source for measurements and transients of other channels, and for digital port pins configured as trigger outputs.

Syntax

[SOURce:]LIST:TOUTput:BOSTep[:DATA] 0|OFF|1|ON,<step>,(@<chanlist>)

Query Syntax

[SOURce:]LIST:TOUTput:BOSTep[:DATA]? <step>,(@<chanlist>)

Example

LIST:TOUTput:BOSTep ON,1,(@1)

[SOURce:]LIST:TOUTput:EOSTep[:DATA] 0|OFF|1|ON,<step>,(@<chanlist>)

This command specifies which list steps generate a trigger signal at the end of the step (EOSTep).

A trigger is only generated when the state is set to ON. The trigger signal can be used as a trigger source for measurements and transients of other channels, and

for digital port pins configured as trigger outputs.

Syntax

[SOURce:]LIST:TOUTput:EOSTep[:DATA] 0|OFF|1|ON,<step>,(@<chanlist>)

Query Syntax

[SOURce:]LIST:TOUTput:EOSTep[:DATA]? <step>,(@<chanlist>)

Example

LIST:TOUTput:EOSTep ON,1,(@1)

[SOURce:]LIST:COUNt:RUN? (@<chanlist>)

This command queries the number of repetitions of the current LIST run.

Syntax

[SOURce:]LIST:COUNt:RUN? (@<chanlist>)

Returns

<NRf+>

Example

LIST:COUN:RUN? (@1)

[SOURce:]LIST:STEP:RUN? (@<chanlist>)

This command is used to query the number of steps of the LIST run.

Syntax

[SOURce:]LIST:STEP:RUN? (@<chanlist>)

Returns：

<NRf+>

Example

LIST:STEP:RUN? (@1)

[SOURce]LIST:FILE:OPEN <string>,LOCal|USB,(@<chanlist>)

This command is used to open a list file in the specified storage path.

<string>: file name

<local_type>:LOCcal/USB

Syntax

[SOURce]LIST:FILE:OPEN <string>,LOCal|USB,(@<chanlist>)

Example

LIST:FILE:OPEN “test.csv”,USB,(@1)

[SOURce]LIST:FILE:SAVE <string>,LOCal|USB,(@<chanlist>)

This command is used to save the list file in the specified storage path.

<string>: file name

<local_type>LOCal/USB

Syntax

[SOURce]LIST:FILE:SAVE <string>,LOCal|USB,(@<chanlist>)

Example

LIST:FILE:SAVE “test.csv”,USB,(@1)

[SOURce:]LIST:FUNCtion:TYPE <type>,(@<chanlist>)

This command is used to set list mode.

Syntax

[SOURce:]LIST:FUNCtion:TYPE <type>,(@<chanlist>)

Arguments

CURRent|VOLTage|POWer|RESistance

Example

LIST:FUNCtion:TYPE VOLTage,(@1)

6.9 Trigger Subsystem

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

This command is used to generate a sampling system immediate trigger signal.

Syntax

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

Example

TRIG:ACQ (@1)

TRIGger:ACQuire:SOURce <source>, (@<chanlist>)

The commands set and query the trigger source of the acquisition system.

Syntax

TRIGger:ACQuire:SOURce <source>, (@<chanlist>)

Arguments

IMMediate|BUS|PIN<1-7>|CURRent<1-8>|VOLTage<1-8>

Query Syntax

TRIGger:ACQuire:SOURce? (@<chanlist>)

Returns

IMM,BUS,,PIN<1-7>,CURR<1-8>,VOLT<1-8>

Example

TRIG:ACQ:SOUR IMM,(@1)

TRIG:ACQ:SOUR VOLTage8

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

The commands set and query the trigger level of the acquisition voltage. (Unit is V)

Syntax

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

Arguments

0-MAXimum|MINimum|MAXimum

Query Syntax

TRIGger:ACQuire:VOLTage[:LEVel]? [MINimum|MAXimum,] (@<chanlist>)

Returns

[MINimum|MAXimum,]

Example

TRIG:ACQ:VOLT? (@1)

TRIGger:ACQuire:VOLTage:SLOPe POSitive|NEGative, (@<chanlist>)

The commands set and query the trigger edge selection of the acquisition voltage.

Syntax

TRIGger:ACQuire:VOLTage:SLOPe POSitive|NEGative, (@<chanlist>)

Arguments

POSitive|NEGative

Query Syntax

TRIGger:ACQuire:VOLTage:SLOPe? (@<chanlist>)

Returns

[MINimum|MAXimum,]

Example

TRIG:ACQ:VOLT:SLOP NEG,(@1)

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

The commands set and query the trigger level of the acquisition current. (Unit is A)

Syntax

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

Arguments

0-MAXimum|MINimum|MAXimum

Query Syntax

TRIGger:ACQuire:CURRent[:LEVel]? [MINimum|MAXimum,] (@<chanlist>)

Returns

<voltage trig level>

Example

TRIG:ACQ:CURR? (@1)

TRIGger:ACQuire:CURRent:SLOPe POSitive|NEGative, (@<chanlist>)

The commands set and query the trigger edge selection of the acquisition current.

Syntax

TRIGger:ACQuire:CURRent:SLOPe POSitive|NEGative, (@<chanlist>)

Arguments

POSitive|NEGative

Query Syntax

TRIGger:ACQuire:CURRent:SLOPe? (@<chanlist>)

Returns

POS or NEG

Example

TRIG:ACQ:CURR:SLOP NEG,(@1)

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

This command is used to generate an external data logger immediate trigger signal.

Syntax

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

Example

TRIG:ACQ (@1)

TRIGger:ELOG:SOURce <source>, (@<chanlist>)

The commands set and query the trigger source of external data records.

Syntax

TRIGger:ELOG:SOURce? (@<chanlist>)

Arguments

IMMediate|BUS|PIN<1-7>

Query Syntax

TRIGger:ELOG:SOURce? (@<chanlist>)

Returns

IMMediate|BUS|PIN<1-7>

Example

TRIG:ELOG:SOUR BUS,(@1)

TRIG:ELOG:SOUR? (@1)

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

This command is used to generate an internal data logger immediate trigger signal.

Syntax

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

Example

TRIG:DLOG (@1)

TRIGger:DLOG:SOURce <source>

The commands set and query the trigger source of internal data logger.

Syntax

TRIGger:DLOG:SOURce <source>

Arguments

IMMediate|BUS|MANual|PIN<1-7>|CURRent<1-8>|VOLTage<1-8>|TRSK|OOOK

TRSK: The transient system function switch is the trigger source;

OOOK: The on/off switch is the trigger source;

Query Syntax

TRIGger:DLOG:SOURce?

Returns

IMM,BUS,MAN,PIN<1-7>,CURR<1-8>,VOLT<1-8>,TRSK,OOOK

Example

TRIG:DLOG:SOUR BUS

TRIG:DLOG:SOUR VOLT8

If CH8 does not exist or exists as a slave, VOLTage8 may not be set as the trigger source for CH1

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

The commands set and query the trigger level for the internal data logger voltage. (Unit is V)

Syntax

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

Arguments

0-MAXimum|MINimum|MAXimum

Query Syntax

TRIGger:DLOG:VOLTage[:LEVel]? [MINimum|MAXimum,] (@<chanlist>)

Returns

[MINimum|MAXimum,]

Example

TRIG:DLOG:VOLT 1.0,(@1)

TRIGger:DLOG:VOLTage:SLOPe POSitive|NEGative, (@<chanlist>)

The commands set and query the trigger edge selection of the internal data logger voltage.

Syntax

TRIGger:DLOG:VOLTage:SLOPe POSitive|NEGative, (@<chanlist>)

Arguments

POSitive|NEGative

Query Syntax

TRIGger:DLOG:VOLTage:SLOPe? (@<chanlist>)

Returns

[MINimum|MAXimum,]

Example

TRIG:ACQ:VOLT:SLOP NEG,(@1)

TRIGger:DLOG:CURRent[:LEVel] <value>, (@<chanlist>)

The commands set and query the trigger level of the internal data logger current. (Unit is A)

Syntax

TRIGger:DLOG:CURRent[:LEVel] <value>, (@<chanlist>)

Arguments

0-MAXimum|MINimum|MAXimum

Query Syntax

TRIGger:DLOG:CURRent[:LEVel]? [MINimum|MAXimum,] (@<chanlist>)

Returns

[MINimum|MAXimum,]

Example

TRIG:ACQ:CURR? (@1)

TRIGger:DLOG:CURRent:SLOPe POSitive|NEGative, (@<chanlist>)

The commands set and query the trigger edge selection of the internal data logger current.

Syntax

TRIGger:DLOG:CURRent:SLOPe POSitive|NEGative, (@<chanlist>)

Arguments

POSitive|NEGative

Query Syntax

TRIGger:DLOG:CURRent:SLOPe? (@<chanlist>)

Returns

POS or NEG

Example

TRIG:DLOG:CURR:SLOP NEG,(@1)

TRIGger:TRANsient[:IMMediate] (@<chanlist>)

This command is used to generate a transient system immediate trigger signal.

Syntax

TRIGger:TRANsient[:IMMediate] (@<chanlist>)

Example

TRIG:TRAN (@1)

TRIGger:MAP:DC[:STATe] 0|OFF|1|ON, (@chanlist)

This command is used to set and query the status of the external output of a trigger signal when the DC value is changed.

Syntax

TRIGger:MAP:DC[:STATe] 0|OFF|1|ON, (@chanlist)

Arguments

OFF|ON

Query Syntax

TRIGger:MAP:DC[:STATe]? (@chanlist)

Returns

0 or 1

Example

TRIG:MAP:DC? (@1)

TRIGger:MAP:DC:PIN PIN<1-7>, (@chanlist)

Set and query the pin that outputs a trigger signal externally when the DC value is changed.

Syntax

TRIGger:MAP:DC:PIN PIN<1-7>, (@chanlist)

Arguments

PIN<1-7>

Query Syntax

TRIGger:MAP:DC:PIN? (@chanlist)

Returns

PIN<1-7>

Example

TRIG:MAP:DC:PIN PIN1,(@1)

TRIGger:MAP:ASToutput[:STATe] 0|1|OFF|ON, (@chanlist)

The commands set and query the status of outputting one trigger signal when ARB step changes.

Syntax

TRIGger:MAP:ASToutput[:STATe] 0|1|OFF|ON, (@chanlist)

Arguments

OFF|ON

Query Syntax

TRIGger:MAP:ASToutput[:STATe]? (@chanlist)

Returns

0 or 1

Example

TRIG:MAP:AST ON,(@1)

TRIGger:MAP:ASToutput:PIN PIN<1-7>, (@chanlist)

This commands set and query the pin that outputs a trigger signal when the Arb step changes.

Syntax

TRIGger:MAP:ASToutput:PIN PIN<1-7>, (@chanlist)

Arguments

PIN<1-7>

Query Syntax

TRIGger:MAP:ASToutput:PIN? (@chanlist)

Returns

PIN<1-7>

Example

TRIG:MAP:AST:PIN PIN1,(@1)

6.10 Digital Subsystem

[SOURce:]DIGital:INPut:DATA?

This command is used to read the status of the digital IO inputs.

Syntax

[SOURce:]DIGital:INPut:DATA?

Returns

<bit value>

Example

DIG:INP:DATA?

[SOURce:]DIGital:OUTPut:DATA <value>

The commands set and query the status of digital IO output.

Syntax

[SOURce:]DIGital:OUTPut:DATA <value>

Arguments

<0-127>

Query Syntax

[SOURce:]DIGital:OUTPut:DATA?

Returns

<bit value>

Example

DIG:OUTP:DATA 21

21 corresponds to bit value 0001 0101

Indicates that the output state of IO-1 IO-3 IO-5 is on.

[SOURce:]DIGital:PIN<1-7>:FUNCtion <function>

The commands set and query the functions of digital IOs. DIO|DINPut|TINput|TOUTput are generic functions, and the all of the IO1-IO7 support the generic functions.

INHibit: IO1 support

FAULt: IO2 support

PSCLear: IO3 support

ONOFfstate: IO4 support

SYNCin,SYNCout: IO5 support

ONCouple: IO6 support

OFFCouple: IO7 support

Syntax

[SOURce:]DIGital:PIN<1-7>:FUNCtion <function>

Arguments

<DIO|DINPut|FAULt|INHibit|ONCouple|OFFCouple|TINPut|TOUTput|ONOFfstate|SIN|SOUT|PSCLear>

Query Syntax

[SOURce:]DIGital:PIN<1-7>:FUNCtion?

Returns

DIO or DINP or FAUL or INH or ONC or OFFC or TINP or TOUT or ONOF or SIN or SOUT or PSCL

Example

DIGital:PIN1:FUNCtion DIO

[SOURce:]DIGital:PIN<1-7>:POLarity POSitive|NEGative

The commands set and query the polarity of the digital IO pin.

Syntax

[SOURce:]DIGital:PIN<1-7>:POLarity POSitive|NEGative

Arguments

POSitive|NEGative

Query Syntax

[SOURce:]DIGital:PIN<1-7>:POLarity?

Returns

POS or NEG

Example

DIGital:PIN1:POLarity POSitive

[SOURce:]DIGital:ONOFf:STATe:CHANnel <1-8>

This command is used to set and query the corresponding channel number of IO4 indicating onoff state.

Syntax

[SOURce:]DIGital:ONOFf:STATe:CHANnel <1-8>

Arguments

<1-8>

Query Syntax

[SOURce:]DIGital:ONOFf:STATe:CHANnel?

Returns

<1-8>

Example

DIGital:ONOFf:STATe:CHANnel 2

[SOURce:]DIGital:PLUSe:WIDTh <value>

The commands set and query the width of digital IO pulse.

Syntax

[SOURce:]DIGital:PLUSe:WIDTh <value>

Arguments

<value>: pulse width

MAXimum: 500us

MINimum: 5us

DEFault: 20us

Query Syntax

[SOURce:]DIGital:PLUSe:WIDTh? [MAXimum|MINimum|DEFault]

Returns

<value>

Example

DIGital:PLUSe:WIDTh 30

6.11 TRANsient Subsystem

[SOURce:]TRANsient:COUNt <value>|MINimum|MAXimum, (@<chanlist>)

This commands set and query the repetition number of continuous transient.

Syntax

[SOURce:]TRANsient:COUNt <value>|MINimum|MAXimum, (@<chanlist>)

Arguments

0-65535|MINimum|MAXimum

Query Syntax

[SOURce:]TRANsient:COUNt? [MIN | MAX] (@<chanlist>)

Returns

<repeat count>

Example

TRAN:coun 10,(@1)

[SOURce:]TRANsient:COUNt:RUN? (@<chanlist>)

This command queries the number of the continuous mode the transient has been run.

Syntax

[SOURce:]TRANsient:COUNt:RUN? (@<chanlist>)

Returns

<repeat count>

Example

TRAN:COUN:RUN? (@1)

[SOURce:]TRANsient:DCYCle <value>|MINimum|MAXimum, (@<chanlist>)

The commands set and query the duration ratio of continuous transient.

Syntax

[SOURce:]TRANsient:DCYCle <value>|MINimum|MAXimum, (@<chanlist>)

Arguments

1-99|MINimum|MAXimum

Query Syntax

[SOURce:]TRANsient:DCYCle? [MINimum | MAXimum,] (@<chanlist>)

Returns

<1-99>

Example

TRAN::DCYCle 10,(@1)

[SOURce:]TRANsient:FREQuency <value>|MINimum|MAXimum,(@<chanlist>)

The commands set and query the frequency of continuous transient.

Syntax

[SOURce:]TRANsient:FREQuency <value>|MINimum|MAXimum, (@<chanlist>)

Arguments

0.0025-1000|MINimum|MAXimum (单位为 Hz)

Query Syntax

[SOURce:]TRANsient:FREQuency? [MINimum | MAXimum,] (@<chanlist>)

Returns

<10.0025-1000>

Example

TRAN:FREQ 20,(@1)

[SOURce:]TRANsient:PERiod <value>|MINimum|MAXimum,(@<chanlist>)

The commands set and query the period of continuous transient.

Syntax

[SOURce:]TRANsient:PERiod <value>|MINimum|MAXimum,(@<chanlist>)

Arguments

minimum-maximum|MINimum|MAXimum （Unit is s）

Query Syntax

[SOURce:]TRANsient:PERiod? [MINimum | MAXimum,] (@<chanlist>)

Returns

< minimum-maximum >

Example

TRANsient:PERiod 20,(@1)

[SOURce:]TRANsient:MODE CONTinuous | PULSe | TOGGle,(@<chanlist>)

The commands set and query the continuous transient mode.

Syntax

[SOURce:]TRANsient:MODE CONTinuous | PULSe | TOGGle,(@<chanlist>)

Arguments

CONTinuous | PULSe | TOGGle

Query Syntax

[SOURce:]TRANsient:MODE? (@<chanlist>)

Returns

CONT,PULS or TOGG

Example

TRAN:MODE CONT (@1)

[SOURce:]TRANsient:TWIDth <value> | MINimum | MAXimum, (@<chanlist>)

The commands set and query the pulse width of transient pulse mode.

Syntax

[SOURce:]TRANsient:TWIDth <value> | MINimum | MAXimum, (@<chanlist>)

Arguments

minimum-maximum|MINimum|MAXimum （Unit is s）

Query Syntax

[SOURce:]TRANsient:TWIDth? [MINimum | MAXimum,] (@<chanlist>)

Returns

minimum-maximum

Example

TRANsient:TWIDth 10,(@1)

[SOURce:]TRANsient:CURRent:TLEVel <transient level>|MINimum|MAXimum,(@<chanlist>)

The commands set and query the transient current level.

Syntax

[SOURce:]TRANsient:CURRent:TLEVel <transient level>|MINimum|MAXimum,(@<chanlist>)

Arguments

minimum-maximum|MINimum|MAXimum （Unit is A）

Query Syntax

[SOURce:]TRANsient:CURRent:TLEVel? [MINimum|MAXimum,](@<chanlist>)

Returns

minimum-maximum

Example

TRANsient:CURRent:TLEVel 15,(@1)

[SOURce:]TRANsient:VOLTage:TLEVel <transient level>|MINimum|MAXimum,(@<chanlist>)

The commands set and query the transient voltage level.

Syntax

[SOURce:]TRANsient:VOLTage:TLEVel <transient level>|MINimum|MAXimum,(@<chanlist>)

Arguments

minimum-maximum|MINimum|MAXimum （Unit is V）

Query Syntax

[SOURce:]TRANsient:VOLTage:TLEVel? [MINimum|MAXimum,](@<chanlist>)

Returns

minimum-maximum

Example

TRANsient:VOLTage:TLEVel 15,(@1)

[SOURce:]TRANsient:POWer:TLEVel <transient level>|MINimum|MAXimum,(@<chanlist>)

The commands set and query the transient power level.

Syntax

[SOURce:]TRANsient:POWer:TLEVel <transient level>|MINimum|MAXimum,(@<chanlist>)

Arguments

minimum-maximum|MINimum|MAXimum （Unit is W）

Query Syntax

[SOURce:]TRANsient:POWer:TLEVel? [MINimum|MAXimum,](@<chanlist>)

Returns

minimum-maximum

Example

TRANsient:POWer:TLEVel 15,(@1)

[SOURce:]TRANsient:RESistance:TLEVel <transient level>|MINimum|MAXimum,(@<chanlist>)

The commands set and query the transient resistance level.

Syntax

[SOURce:]TRANsient:RESistance:TLEVel <transient level>|MINimum|MAXimum,(@<chanlist>)

Arguments

minimum-maximum|MINimum|MAXimum （Unit is Ω）

Query Syntax

[SOURce:]TRANsient:RESistance:TLEVel? [MINimum|MAXimum,](@<chanlist>)

Returns

minimum-maximum

Example

TRANsient:RESistance:TLEVel 2,(@1)

6.12 BATTery DISCharge Subsystem

BATTery:DISCharge:MODE <CURRent|VOLTage>,(@<chanlist>)

This commands set the battery discharging mode.

Syntax

BATTery:DISCharge:MODE <CURRent|VOLTage>,(@<chanlist>)

Arguments

CURRent|VOLTage

Example

BATT:DISC:MODE CURR,(@1)

Query Syntax

BATTery:DISCharge:MODE? (@<chanlist>)

Returns

CURRent|VOLTage

BATTery:DISCharge:VOLTage <value>,(@<chanlist>)

Set the voltage for battery discharging.

Syntax

BATTery:DISCharge:VOLTage <value>,(@<chanlist>)

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:DISC:VOLT 5.0,(@1)

Query Syntax

BATTery:DISCharge:VOLTage? [MINimum|MAXimum],(@<chanlist>)

Returns

NR3

BATTery:DISCharge:CURRent <value>,(@<chanlist>)

Set the current for battery discharging.

Syntax

BATTery:DISCharge:CURRent <value>,(@<chanlist>)

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:DISC:CURR 3.0,(@1)

Query Syntax

BATTery:DISCharge:CURRent? [MINimum|MAXimum],(@<chanlist>)

Returns

NR3

BATTery:DISCharge:POWer <value>,(@<chanlist>)

Set the power for battery discharging.

Syntax

BATTery:DISCharge:POWer <value>,(@<chanlist>)

Arguments

<MINimum-MAXimum|MINimum|MAXimum>

Example

BATT:DISC:POW 3.0,(@1)

Query Syntax

BATTery:DISCharge:POWer? [MINimum|MAXimum,] (@<chanlist>)

Returns

NR3

BATTery:DISCharge:RESistance <value>,(@<chanlist>)

Set and query the battery discharge resistance.

Syntax

BATTery:DISCharge:RESistance <value>,(@<chanlist>)

Arguments

0-MAXimum|MINimum|MAXimum

Example

BATT:DISC:RESistance 1000,(@1)

Query Syntax

BATTery:DISCharge:RESistance? [MINimum|MAXimum,] (@<chanlist>)

Returns

NR3

BATTery:DISCharge:STOP:VOLTage <value>,(@<chanlist>)

This command sets the stop discharging voltage level for battery discharging test.

Syntax

BATTery:DISCharge:STOP:VOLTage <value>,(@<chanlist>)

Arguments

0-MAXimum|MINimum|MAXimum (unit is V)

Example

BATT:DISCharge:STOP:VOLT 4.0,(@1)

Query Syntax

BATTery:DISCharge:STOP:VOLTage? [MINimum|MAXimum,] (@<chanlist>)

Returns

NR3

BATTery:DISCharge:STOP:VOLTage:STATe <0|1|OFF|ON>,(@<chanlist>)

This command sets the stop discharging voltage status for battery discharging test.

Syntax

BATTery:DISCharge:STOP:VOLTage:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATTery:DISC:STOP:VOLT:STAT 1,(@1)

Query Syntax

BATTery:DISCharge:STOP:VOLTage? [MINimum|MAXimum,] (@<chanlist>)

Returns

0|1

BATTery:DISCharge:STOP:CURRent <value>,(@<chanlist>)

This command sets the stop discharging current level for battery discharging test.

Syntax

BATTery:DISCharge:STOP:CURRent <value>,(@<chanlist>)

Arguments

0-MAXimum|MINimum|MAXimum (Unit is A)

Example

BATT:DISC:STOP:CURR 2.0, (@1)

Query Syntax

BATTery:DISC:STOP:CURRent? <value>,(@<chanlist>)

Returns

NR3

BATTery:DISCharge:STOP:CURRent:STATe <0|1|OFF|ON>,(@<chanlist>)

This command sets the stop discharging current status for battery discharging test.

Syntax

BATTery:DISCharge:STOP:CURRent:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATT:DISC:STOP:CURR:STAT 1,(@1)

Query Syntax

BATTery:DISCharge:STOP:CURRent:STATe? (@<chanlist>)

Returns

0|1

BATTery:DISCharge:STOP:AHOur <value>,(@<chanlist>)

This command sets the stop discharging AHour level for battery discharging test.

Syntax

BATTery:DISCharge:STOP:AHOur <value>,(@<chanlist>)

Arguments

0.01-999.99|MINimum|MAXimum (Unit is AH)

Example

BATT:DISC:STOP:AHO 0.1,(@1)

Query Syntax

BATTery:DISCharge:STOP:AHOur? [MINimum|MAXimum,] (@<chanlist>)

Returns

MINimum|MAXimum

BATTery:DISCharge:STOP:AHOur:STATe <0|1|OFF|ON>,(@<chanlist>)

This command sets the stop discharging AHour status for battery discharging test.

Syntax

BATTery:DISCharge:STOP:AHOur:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATT:DISCharge:STOP:AHO:STAT 1,(@1)

Query Syntax

BATTery:DISCharge:STOP:AHOur:STATe? (@<chanlist>)

Returns

0|1

BATTery:DISCharge:STOP:WHOur <value>,(@<chanlist>)

This command sets the stop discharging WHour level for battery discharging test.

Syntax

BATTery:DISCharge:STOP:WHOur <value>,(@<chanlist>)

Arguments

0.01-999.99|MINimum|MAXimum (Unit is WH)

Example

BATT:DISC:STOP:WHO 0.1,(@1)

Query Syntax

BATTery:DISCharge:STOP:WHOur? [MINimum|MAXimum,] (@<chanlist>)

Returns

MINimum|MAXimum

BATTery:DISCharge:STOP:WHOur:STATe <0|1|OFF|ON>,(@<chanlist>)

This command sets the stop discharging WHour status for battery discharging test.

Syntax

BATTery:DISCharge:STOP:WHOur:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATT:DISC:STOP:WHO:STAT 1,(@1)

Query Syntax

BATTery:DISCharge:STOP:WHOur:STATe? (@<chanlist>)

Returns

0|1

BATTery:DISCharge:STOP:TIME <value>,(@<chanlist>)

Set and query the stopping time of battery discharge.

Syntax

BATTery:DISCharge:STOP:TIME <value>,(@<chanlist>)

Arguments

0-86400|MINimum|MAXimum

Example

BATT:DISC:STOP:TIME 12.0,(@1)

Query Syntax

BATTery:DISCharge:STOP:TIME? [MINimum|MAXimum,] (@<chanlist>)

Returns

MINimum|MAXimum

BATTery:DISCharge:STOP:TIME:STATe <0|1|OFF|ON>,(@<chanlist>)

Setting and querying the status of the time stop of battery discharge.

Syntax

BATTery:DISCharge:STOP:TIME:STATe <0|1|OFF|ON>,(@<chanlist>)

Arguments

<0|1|OFF|ON>

Example

BATT:DISC:STOP:TIME:STAT 0,(@1)

Query Syntax

BATTery:DISCharge:STOP:TIME:STATe? (@<chanlist>)

Returns

0|1

BATTERY:DISCharge:SAVe <"filename">,(@<chanlist>)

This command saves the edit test item as a file, overwriting the current file if the file name already exists, otherwise, saving as a new file.

Syntax

BATTERY:DISCharge:SAVe <"filename">,(@<chanlist>)

Arguments

<string>

Example

BATT:DISC:SAVe (@1),"test.csv"

BATTery:DISCharge:RECall <"filename">,(@<chanlist>)

This command recalls the battery discharging test file stored in instrument.

Syntax

BATTery:DISCharge:RECall <"filename">,(@<chanlist>)

Arguments

<string>

Example

BATT:DISC:RECall "test.csv",(@1)

BATTery:DISCharge:DELete <"filename">,(@<chanlist>)

This command deletes the battery discharging test file stored in instrument.

Syntax

BATTery:DISCharge:DELete <"filename">,(@<chanlist>)

Arguments

<string>

Example

BATT:DISC:DEL "test.csv",(@1)

BATTery:DISCharge:TIME? (@<chanlist>)

This command queries how long the battery discharge has been running.

Syntax

BATTery:DISC:TIME? (@<chanlist>)

Returns

<NRF>

Example

BATT:DISC:TIME? (@1)

BATTery:DISCharge:RSTate? (@<chanlist>)

This command queries the battery discharge status.

Syntax

BATTery:DISCharge:RSTate? (@<chanlist>)

Returns

"Idle","Discharging ","Voltage Up","Current Up","Energy Up","Time up","Abort"

Example

BATT:DISCharge:RST? (@1)

Copyright ©ITECH Electronic Co., Ltd.

Chapter7IEEE-488 Common Commands

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

*CLS

This command clears the error queue and the bits of the following registers:

- Standard Event Register
- Questionable Event Register
- Status byte register

Syntax:

*CLS

Arguments:

None

*ESE

This command sets the bits in the Event Status Enable Register (ESER). The ESER is an eight-bit register that determines which bits in the Standard Event Status Register (SESR) will set the Event Summary Bit (ESB) in the Status Byte Register (SBR).

Syntax

*ESE <NR1>

Arguments

0～255

Default value

0

Example

*ESE 128

Query syntax

*ESE?

Returns

<NR1>

Related syntax

*ESR? *PSC *STB?

Bit definition of standard event enable register:

Bit	7	6	5	4	3	2	1	0
Name	PON	n.u	CME	EXE	DDE	QYE	n.u	OPC
Value Bit Weight	128		32	16	8	4		1

*ESE?

This command queries the bits in the Event Status Enable Register (ESER).

Query syntax

*ESE?

Arguments

None

Returns

<NR1> This is a decimal value with a binary weighted sum.

*ESR?

This command reads the value of Standard Event Status Register (SESR). Once this command executes, the SESR is reset. The bit definition for the SESR is the same as the Standard Event Status Enable Register.

Query syntax

*ESR?

Arguments

None

Returns

<NR1> This is a decimal value with a binary weighted sum.

Related syntax

*CLS *ESE *ESE? *OPC

*IDN?

This command reads information that identifies the electronic load. It returns a parameter that contains four segments divided by a comma.

Query syntax

*IDN?

Arguments

None

Returns

Manufacture, model, serial number, UI ver-DSP1 ver-DSP2 ver-PFC ver-Interface

Copyright ©ITECH Electronic Co., Ltd.

ver

Example

Itech\sElectronics,IT8215-350-90,804947011767740001,

000.001.038,83.R,13.R\n

*OPC

This command sets the Operation Complete (OPC) bit in the Standard Event Status Register to 1 when all other commands are complete.

Syntax

*OPC

Arguments

None

Query syntax

*OPC?

Returns

<NR1>

*RST

This command resets the electronic load to default settings.

Syntax

*RST

Arguments

None

*SRE <NR1>

This command sets or queries the bits in the Status Byte Enable Register. Setting this parameter can determine which byte of the Status Byte Register has a value of 1. The byte sets the RQS bit of the Status Byte Register to 1. The bit definition of the Status Byte Enable Register is as the same as the Status Byte Register.

Syntax

*SRE <NR1>

Arguments

0～255

Default value

Refer to *PSC command

Example

*SRE 128

Query syntax

*SRE?

Returns

<NR1>

Related syntax

*ESE *ESR? *PSC *STB?

*STB?

This command reads the data in the Status Byte Register (SBR).

Query syntax

*STB?

Arguments

None

Returns

<NR1>

Related syntax

*CLS *ESE *ESR

Bit definition of the Status Byte Register:

Bit	7	6	5	4	3	2	1	0
Name	OPER	RQS	ESB	MAV	QUES	EAV	n.u	n.u
Value Bit Weight	128	64	32	16	8	4		

*PSC

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.

Syntax

*PSC <ON|1|OFF|0>

Arguments

0|1|ON|OFF

Query syntax

*PSC?

Returns

<ON|1|OFF|0>

*SAV

This command saves the present setting values of the electronic load into specified memory.

Syntax

*SAV <NR1>

Arguments

0~9

*RCL

This command recalls the setups you saved in the specified memory location.

Syntax

*RCL <NR1>

Arguments

0～9

Connect with us

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

chat

You can chat with us on ITECH website

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

ITECH Web

ITECH News

Click and follow ITECH Electronics