Programming Manual

BCS Series

Battery Charger/Simulator and DC Power Supply

bkprecision.com

Contents

About Commands & Queries

This section lists and describes the remote control commands and queries recognized by the instrument. All commands and queries can be executed in both local or remote mode.

The description, command syntax, query syntax, example and respond can be found in a section. The commands are given in both long and short form. All examples are shown in short form. Queries obtain information, and are recognized by the question mark (?) following the header.

1.1 How They are Listed

The commands are listed by subsystem.

1.2 How They are Described

In the descriptions themselves, a brief explanation of the function performed is given. This is followed by a presentation of the formal syntax, with the header given in Upper-and-Lower-Case characters and the short form derived from it in ALL UPPER-CASE characters. Where applicable, the syntax of the query is given with the format of its response.

1.3 When can they be used?

The commands and queries listed here can be used for BCS640X Series.

1.4 Command Notation

The following notation is used in the commands:

< > Angular brackets enclose words that are used as placeholders, of which there are two types: the header path
and the data parameter of a command.

:= A colon followed by an equals sign separates a placeholder from the description of the type and range of values that
may be used in a command instead of the placeholder.

{ } Braces enclose a list of choices, one of which one must be made.

[ ] Square brackets enclose optional items.

… An ellipsis indicates that the items both to its left and right may be repeated a number of times.

Common Command Introduction

IEEE standard defines the common commands used for querying the basic inSyntaxion of the instrument or executing basic operations. These commands usually start with "*" and and have a 3 character length.

Short	Long	Subsystem	Description
*CLS	*CLS	SYSTEM	Clears the status byte by emptying the error queue and clearing all event registers and preceding *OPC.
*ESE	*ESE	SYSTEM	Sets bits in the standard event status enable register.
*ESE?	*ESE?	SYSTEM	Returns the results of the standard event enable register. The register is cleared after reading it.
*ESR?	*ESR?	SYSTEM	Reads and clears the contents of the Event Status Register.
*IDN	*IDN	SYSTEM	Returns a string that uniquely identifies the instrument.
*OPC	*OPC	SYSTEM	Generates the OPC message in the standard event status register when all pending overlapped operations are complete.
*OPC?	*OPC?	SYSTEM	Returns an ASCII "+1" when all pending overlapped operations are complete.
*PSC?	*PSC?	SYSTEM	Gets or sets the OPC bit (0) in the Event Status Register.
*RCL	*RCL	SYSTEM	Recalls a saved instrument state.
*SAV	*SAV	SYSTEM	Save instrument state.
*RST	*RST	SYSTEM	Initiates a device reset.
*SRE	*SRE	SYSTEM	Set status byte enable register.
*SRE?	*SRE?	SYSTEM	Query status byte enable register.
*STB	*STB	SYSTEM	Query status byte.
*TRG	*TRG	SYSTEM	Generates an immediate trigger.
*TST?	*TST?	SYSTEM	Returns the result of the self-test.
*WAI	*WAI	SYSTEM	Prohibits the instrument from executing any new commands until all pending overlapped commands have been completed

Table 2.1 Common Commands

2.1 *CLS

Description This command clears all status data structures in a device.
For a device which minimally complies with SCPI, these registers are:

SESR	
OPERation Status	
Register	**(IEEE 488.2)**
QUEStionable Status	**(SCPI)**
Register	**(SCPI)**
Error/Event Queue	**(SCPI)**

Execution of *CLS shall also clear any additional status data structures implemented in the device. The corresponding enable registers are unaffected.

***CLS** forces the device into OCIS and OQIS (see **2.5**, ??, and ??) without setting the **No Operation Pending** flag TRUE and without setting the OPC bit of the SESR TRUE and without placing a “1” into the Output Queue.

For example, suppose a device implements **INITiate[:IMMediate]** as an overlapped command. Assuming that the trigger model is programmed so that it will eventually return to the IDLE state, and that INITiate[:IMMediate] takes longer to execute than *OPC, sending these commands to this device:

INITiate;*OPC

results in initiating the trigger model and, after some time, setting the OPC bit in the SESR. However, sending these commands:

INITiate;*OPC;*CLS

still initiates the trigger model. Since the operation is still pending when the device executes ***CLS**, the device does not set the OPC bit until it executes another ***OPC** command.

Example *CLS

2.2 *ESE

Description **Event Status Enable Command and Query**. Enables bits in the enable register for the Standard Event Register group. The selected bits are then reported to bit 5 of the Status Byte Register. The following types of events are reported: power-on detected, command syntax errors, command execution errors, self-test or calibration errors, query errors, or the *OPC command has been executed. Any or all of these conditions can be reported to the Standard Event summary bit through the enable register. To set the enable register mask, you must write a decimal value to the register using the *ESE command. See **Table 2.2**

The *ESE? query returns a decimal value which corresponds to the binary-weighted sum of all bits enabled by the *ESE command.

The **Standard Event Enable Register** is cleared when:

- The ***ESE 0** command is executed.
- The instrument was powered on and configured so the function generator clears the enable register using the ***PSC 1** command.
- ***CLS** does not clear the enable register but it does clear all bits in the event register.
- A STATus:PRESet does not clear the bits in the Status Byte enable register.

Note:

The enable register will not be cleared at power-on if the function generator was configured using the ***PSC 0** command.

Syntax *ESE <NR1>
<NR1> := 0 to 128.

For example, to enable bit 2 (value 4), bit 3 (value 8), and bit 7 (value 128), the decimal sum would be 140 (4 + 8 + 128). The default decimal sum is 0.

Query *ESE?

Response *ESE <NR1>

Example *ESE?

Bit	Bit Name	Decimal Value	Definition
0	OPC	1	All commands prior to and including *OPC have completed and the overlapped command (e.g., *TRG for burst) has completed.
1	not used	not used	Not used. Returns 0.
2	QYE	4	The instrument tried to read the output buffer but it was empty. Or, a new command line was received before a previous query has been read. Or, both the input and output buffers are full.
3	DDE	8	A self-test, cal, or other device-specific error has occurred.
4	EXE	16	An execution error has occurred.
5	CME	32	A command syntax error has occurred.
6	not used	64	Not used. Returns 0.
7	PON	128	Power has been cycled on since the last time the event register was read or cleared.

Table 2.2 Standard Event Register

2.3 *ESR

Description Query the Standard Event Status Register. Once a bit is set, it remains set until cleared by a ***CLS** (clear status) command or queried by this command. A query of this register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

Syntax *ESR <NR1>
<NR1> := 0 to 255

Query *ESR?

Example *ESR?

Return: 0

Related *CLS, *ESE

2.4 *IDN?

Description The *IDN? query causes the instrument to identify itself. The response comprises manufacturer, model, serial number, software version and firmware version.

Query *IDN?

Response *IDN, <device id>,<model>,<serial number>, <software version>, <hardware version>.

<device id>:="BK" is used to identify instrument.

<model>:= A model identifier less than 14 characters will contain the model number.

<serial number>:= Number that uniquley identifies the instrument.

<firmware version>:= Firmware revision number.

<hardware version>:= Hardware revision number.

Example *IDN?

Returns: B&K Precision,BCS6402,XXXXXXXX,0.25_1029A-0.15_0804A

2.5 *OPC

Description Sets the **Operation Complete** bit (bit 0) in the **Standard Event Register** after all of the previous commands have been completed. Other commands may be executed before the bit is set.

This command is used to stops the controller until all pending commands are completed. ***OPC?** returns **"1"** to the output buffer after the previous commands have been completed.

Other commands cannot be executed until this command completes.

Syntax *OPC

Query *OPC?

Example INITiate;*OPC

Response **"1"**

2.6 *PSC

Description **Power-On Status Clear**. Clears the Standard Event Enable Register and Status Byte Condition Register at power on (*PSC 1). When *PSC 0 is in effect, these two registers are not cleared.

The ***PSC?** query returns the power-on status clear setting. Returns **“0”** (do not clear at power on) or **“1”** (clear at power on).

Syntax *PSC <bool>
<bool> := {0 |1}
Default := *PSC 1

Query *PSC?

Example *PSC 1

Response **"1"**

2.7 *RCL

Description Recalls the instrument state stored in the specified non-volatile storage location. The instrument’s state cannot be recalled from an empty storage location. When shipped from the factory, storage locations “1” through “9” are empty (location “0” has the power-on state).

After attempting to recall an empty location the message **"Failed to recall settings"** will be displayed in the top left corner of the instruments’s display.

Setting OUTPut:PN:STATe to USER will load the settings saved in the selected location when the instrument is powered on.

Syntax *RCL <memory address>
<memory address> := {0 to 9}

Example *RCL 1

Related ***SAV**

2.8 *SAV

Description Store (save) the current instrument state in the specified non-volatile storage location. Any state previously stored in the same location will be overwritten. The instrument state can be stored in any of the 10 storage locations (0-9).

An instrument reset (***RST** command) does not affect the configurations stored in memory. Once a state is stored, it remains until it is overwritten or specifically deleted.

If OUTPut:PN:STATe is set to LAST the settings set when the instrument is powered off will be savein the selected User Settings, overwritting any previous saved settings in theis location.

Syntax *SAV <NR1>
<NR1> := {0 to 9}

Example *SAV 0

Related *RCL

2.9 *RST

Description Reset the instrument to its factory default state. *RST does not affect stored instrument states, or the I/O settings, which are stored in non-volatile memory.

Syntax *RST

Example *RST

2.10 *SRE

Description The Status Byte Summary Register reports conditions from the other status registers. Data waiting in the instrument's output buffer is immediately reported on the **"Message Available"** bit (bit 4). Clearing an event register from one of the other register groups will clear the corresponding bits in the **Status Byte Condition Register**. Reading all messages from the output buffer, including any pending queries, will clear the **"Message Available"** bit.

To enable specific bits, you must write a decimal value which corresponds to the binary-weighted sum of the bits in the register. The selected bits are summarized in the **"Master Summary"** bit (bit 6) of the Status Byte Register. If any of the selected bits change from **"0"** to **"1"**, a Service Request Signal (SRQ) is generated.

The ***SRE?** query returns a decimal value which corresponds to the binary-weighted sum of all bits enabled by the ***SRE** command

The *SRE? query returns a value that, when converted to a binary number represents the bit settings of the SRE register. Note that bit 6 (MSS) cannot be set and it's returned value is always zero.

- ***CLS** (clear status) does not clear the enable register but it does clear all bits in the event register.
- STATus:PRESet does not clear the bits in the Status Byte enable register.
- ***PSC 0** preserves the contents of the enable register through power cycles.

Syntax *SRE <NR1>
<NR1> := 0 to 255

Query *SRE?

Example *SRE 4

Response **"4"**

2.11 *STB?

Description Query the summary (condition) register in this register group. This command is similar to a Serial Poll but it is processed like any other instrument command. This command returns the same result as a Serial Poll but the **"Master Summary"** bit (bit 6) is not cleared by the ***STB?** command.

Query *STB?

Example *STB?

Response **0 to 255**

Related *CLS, *SRE

Bit	Bit Name	Decimal Value	Definition
0 to 2	not used	not used	Not used.
3	QUES	8	One or more bits are set in the **Questionable Status Register**. (bits must be enabled)
4	MAV	16	Data is available in the instrument's output buffer.
5	ESB	32	One or more bits are set in the **Standard Event Register**. (bits must be enabled)
6	RQS	64	The Service Request Line (SRQ) on the GPIB is designed to signal the Controller when a service request is pending.
7	OPER	128	One or more bits are set in the **Operation Statys Byte Register**. (bits must be enabled)

Table 2.3 Status Byte Register

2.12 *TRG

Description The *TRG command generates an immediate trigger when the trigger source is set to **BUS**.

Syntax *TRG

Example *TRG

Related TRIGger:SOURce

2.13 *TST?

Description The *TST? query performs an internal self-test of the power supply. Returns **“0”** (PASS) or **“1”** (FAIL). If the test fails, one or more error messages will be generated to provide additional information on the failure. Use the SYSTem:ERRor? command to read the error queue

Query *TST?

Example *TST?

Response **"0"**

2.14 *WAI

Description Wait for all pending operations to complete before executing any additional commands over the interface.

Syntax *WAI

Example The following command string guarantees that the first trigger is accepted and the operation is executed before the second trigger is recognized.

TRIG:SOUR BUS;*TRG;*WAI;*TRG;*WAI

Related *OPC?

Status Subsystem

The commands in the **Status Subsystem** program all of the instruments status registers.

3.1 STATus:QUEStionable?

Description Returns the value of the questionable event register.The Event register is a read-only register that saves all the events that pass into it. Reading the Questionable Event register clears it. This command is not channel specific, it applies to the entire mainframe

Query STATus:QUEStionable:[EVENt]?

Example STAT:QUES?

Response Returns: {Register Value}

3.2 STATus:QUEStionable:ENABle

Description Sets or reads the value of the Questionable Enable register.This register is a mask for enabling specific bits from the **Questionable Event** register to set the questionable summary bit of the Status Byte register.This bit (bit 3) is the logical OR of all the Questionable Event register bits that are enabled by the Questionable Status Enable register. This command is not channel specific, it applies to the entire mainframe.

Syntax STATus:QUEStionable:ENABle <bit>
<bit> := {0 to 65535}

Query STATus:QUEStionable:ENABle?

Example STAT:QUES:ENAB 1

Response Returns: {Enabled Register Number}

3.3 STATus:QUEStionable:PTRansition

Description Set or read the positive value of the questionable condition register. When the bit value of the questionable condition register changes from 0 to 1, and the corresponding bit of ptransition register is 1, then the corresponding bit value of questionable event register turns into 1.

Syntax STATus:QUEStionable:PTRansition <bit>
<bit> := {0 to 65535}

Query STATus:QUEStionable:PTRansition?

Example STAT:QUES:PTR 64

Response Returns: 64 {Register Number}

3.4 STATus:QUEStionable:NTRansition

Description Set or read the negative value of the questionable condition register. When the bit value of the questionable condition register changes from 1 to 0, and the corresponding bit of ntransition register is 1, then the corresponding bit value of questionable event register turns into 1.

Syntax STATus:QUEStionable:NTRansition <bit>

Query STATus:QUEStionable:NTRansition?

Example STAT:QUES:NTR 2

Response Returns: 2 {Register Number}

3.5 STATus:QUEStionable:CONDition?

Description Returns the condition of the **Questionable Status** register. The questionable conditional register is a read-only register that holds the real-time status of the power supply. This command is not channel specific, it applies to the entire mainframe.

Query STATus:QUEStionable:CONDition?

Example STAT:QUES:COND?

Response Returns: Returns the sum of triggered conditions in the **Questionable Status**. See table 3.1

3.6 STATus:OPERation?

Description Returns the value of the Operation Event register. The Event register is a read-only register that holds all events that are passed by the Operation NTR and/or PTR filter. Reading the Operation Event register clears it. This command is not channel specific, it applies to the entire mainframe.

Query STATus:OPERation[:EVENt]?

Example STAT:OPER?

Response Returns: {Register Number}

3.7 STATus:OPERation:ENABle

Description Write and read the value of the **Operation Enable** register. This register is a mask for enabling specific bits from the Operation Event register to set the operation summary bit of the Status Byte register. The operation summary bit is the logical OR of all enabled Operation Event register bits. This command is not channel specific, it applies to the entire mainframe.

Syntax STATus:OPERation:ENABle <bit>
<bit> := {0 to 65535}

Query STATus:OPERation:ENABle?

Example STAT:OPER:ENAB 2

Response Returns: 2 {Enabled Register Number}

3.8 STATus:OPERation:CONDition?

Description Returns the value of the **Operation Condition** register. The register is a read-only register that holds the real-time operational status of the power supply. This command is not channel specific, it applies to the entire mainframe.

Query STATus:OPERation:CONDition?

Example STAT:OPER:COND?

Response Returns: Returns the sum of triggered conditions in the **Operation Condition** register. See table 3.2

3.9 STATus:PRESet

Description Clears all bits of the following registers:

- Questionable Event Enable Register.
- Channel summary Event Enable Register.
- Operation Event Enable Register

Syntax STATus:PRESet

Example STAT:PRES

Note:

Registers not included in the above list are not affected by this command.

Bit	Bit Name	Decimal	Definition
0	OVP	1	Bit is 1 when OVP (Over Voltage Protection) occurs.
1	OVP	2	Bit is 1 when OVP2 (Over Voltage Protection) occurs.
2	OCP	4	Bit is 1 when OVP (Over Current Protection) occurs
3	OCP2	8	Bit is 1 when OVP (Over Current Protection) occurs
4	FP	16	Bit is 1 when any key is pressed and the unit under Local Mode.
5	OTP	32	Bit is 1 when OTP (Over Temperature Protection) occurs.
6	UNR	64	Bit is 1 when output is unregulated.
7	UNR2	128	Bit is 1 when output is unregulated.
8	RI	256	Bit is 1 when **Ext_ON/OFF State** is set to Off.
9	OVERLOAD	512	Bit is 1 when CH1 has overload current.
10	OVERLOAD	1024	Bit is 1 when CH2 has overload current.
11 to 15	not used	not used	not used

Table 3.1 Questionable Status Summary

Bit	Bit Name	Decimal	Definition
0	CAL	1	Bit is 1 when calculating new calibration parameters .
1	ERR	2	Bit is 1 when an error occured in CH1
2	ERR2	4	Bit is 1 when an error occured in CH2.
3	WTG	8	Bit is 1 when instrument is waiting for a trigger.
4	ON	16	Bit is 1 when CH1 output is enabled.
5	ON2	32	Bit is 1 when CH2 output is enabled.
6	CV	64	Bit is 1 when CH1 is operating in constant voltage mode.
7	CV2	128	Bit is 1 when CH2 is operating in constant voltage mode.
8	CC	256	Bit is 1 when CH1 is operating in constant current mode.
9	CCN	512	Bit is 1 when CH1 is operating in negative constant current mode.
10	CC2	1024	Bit is 1 when CH2 is operating in constant current mode.
11	CCN2	2048	Bit is 1 when CH1 is operating in negative constant current mode.
12	BATTRUN	4096	Bit is 1 when CH1 is under battry running status.
13	BATTRUN2	8192	Bit is 1 when CH2 is under battry running status.
14 & 15	not used	not used	not used

Table 3.2 Operation Status Summary

Bit	Bit Name	Decimal	Definition
0	OPC	1	Bit is 1 when the operation specified has been completed.
1	not used	not used	not used
2	QYE	4	Query Error. Data of the output array is missing.
3	DDE	8	Device-Dependent Error. Data stored in register is missing or error occurs in preliminary checkout.
4	EXE	16	Execution Error. Order parameter overflows or the condition is not right.
5	CME	32	Command Error.Command syntax or semantic error occurs when receiving information.
6	not uesed	not used	not used
7	PON	128	Bit is 1 when the device is reset.

Table 3.3 Standard Event Summary

Bit	Bit Name	Decimal	Definition
0 to 2	not used	not used	not used
3	QUES	8	Bit is 1 when one or more bits are set in the Questionable Data Register. See **STATus:QUEStionable:ENABle** .
4	MAV	16	Bit is 1 when data is available in the instrument's output buffer.
5	ESB	32	Bit is 1 when one or more bits are set in the status byte register and may generates a service request.
6	RQS	64	Request Service (RQS) Summary Bit. A 1 in this bit position indicates that the signal generator has at least one reason to require service. This bit is also called the Master Summary Status bit (MSS). The individual bits in the Status Byte are individually ANDed with their corresponding service request enable register, then each individual bit value is ORed and input to this bit.
7	OPER	128	Bit is 1 when one or more bits are set in the operation status register.

Table 3.4 Status Byte Summary

Output

The output subsystem provides all the commands available to program the output of both channels.

4.1 INSTrument:COUPle:OUTPut:STATe

Description Set the couple output of instrument's channel. If state is 1, the ON and OFF function of both channels are simultaneous. If state is 0, the ON and OFF function of each channel is separate.

Syntax INSTrument:COUPle:OUTPut:STATe <state>
<state> := {1 | 0 or On | Off }

Query INSTrument:COUPle:OUTPut:STATe?

Example INST:COUP:OUTP:STAT 1
INST:COUP:OUTP:STAT?

Response Returns: 1

4.2 OUTPut[n][:STATe]

Description Enable/disable the output of the selected channel.

Syntax OUTPut<channel>[:STATe] <state> <channel> := {1 | 2}
<state> := {1 | 0 or On | Off }

Query OUTPut<channel>[:STATe]?

Example OUTP2 1 Enable channel 2 output
OUTP2?

Response Returns: 1

4.3 OUTPut:RELay:MODE

Description Set the output relay mode.This function only applies to channnel 1.

Syntax OUTPut:RELay:MODE <mode> <mode> := {BATTery | NORMal}

Query OUTPut:RELay:MODE?

Example OUTP:REL:MODE BATT Set battery mode
OUTP:REL:MODE?

Response Returns: BATTery

4.4 OUTPut[n]:COMPensation:MODE

Description Programs the output compensation circuit. This circuit compensates the output of the dc source according to the input capacitance of the device being tested as well as the type of output connections being used.

Syntax OUTPut<channel>:COMPensation:MODE <mode>
<channel> := {1 | 2}
<mode> := {see table **4.1**}

Mode	Description
LLocal	Used with short load leads. Produces the slowest output response, but provides the best stability without a need of an external capacitor.
LRemote	Used with long load leads and remote sense. Produces a slower output response.
HLocal	Used with short load leads. Produces a faster output response without the need of an external capacitor.
HRemote	Used with long load leads and remote sense. Produces the fastest output response, but requires an external capacitor for stability.

Table 4.1 Compensation Modes

Note:

Factory default is set to HRemote mode.

Query OUTPut:COMPensation:MODE?

Example OUTP1:COMP:MODE LLOCAL
OUTP1:COMP:MODE LLOCAL

Response Returns: LLOCAL

4.5 OUTP[n]:TYPE[:CAPacitance]

Description Set the output type to either LOW (slow) or HIGH (fast).

Syntax OUTP<channel>:TYPE[:CAPacitance] <range>
<channel> := {1 | 2} <range> := {LOW | HIGH}

Query OUTPut<channel>:TYPE[:CAPacitance]?

Example OUTPut2:TYPE HIGH
OUTPut2:TYPE?

Response Returns: HIGH

4.6 OUTPut:DFI[:STATe]

Description Enable/disable the discrete fault indicator (DFI) output of the dc source.

Syntax OUTPut:DFI[:STATe] <state>
<state> := {1 | 0 or On | Off}

Query OUTPut:DFI[:STATe]?

Example OUTPut:DFI 1
OUTPut:DFI?

Response Returns: 1

4.7 OUTPut:DFI:SOURce

Description

Syntax OUTPut:DFI:SOURce <indicator>
<indicator> := {see table **4.2**}

Mode	Description
QUES	Set the indicator to bit 3 of the **Questionable Event Summary**.
OPER	Set the indicator to bit 3 of the **Operation Event Summary**.
ESB	Set the indicator to bit 5 of the **Statndard Event Summary**.
RQS	Set the indicator to bit 6 of the **Reqest Service Status Byte Register**.
OFF	Assign no indicator to the DFI source.

Table 4.2 DFI Indicators

Query OUTPut:DFI:SOURce?

Example OUTP:DFI:SOUR QUES OUTP:DFI:SOUR?

Response Returns: QUES

4.8 OUTPut:PROTection:CLEar

Description Clears all events that disable the output when one or more of the listed protections are triggered:

- OVP (Over-Voltage Protection)
- OCP (Over-Current Protection)
- OTP (Over-Temperature Protection)
- REM INH (Remote Inhibit)

Syntax OUTPut:PROTection:CLEar

Example OUTP:PROT:CLE

4.9 OUTPut[n]:PROTection:DELay

Description Set the OCP delay. The delay will prevents the subsystem status bit from being recompiled and causing mis-operation due to transient-varied CC status.

Syntax OUTPut<channel>:PROTection:DELay <time>
<channel> := {1 | 2}
<state> := {0 to 60 s}

Query OUTPut<channel>:PROTection:DELay?

Example OUTPut2:PROTection:DELay 30
OUTPut2:PROTection:DELay?

Response Returns: 30 s {time}

4.10 OUTPut:RI:MODE

Description Set themode of opereation of the Remore Inhibit protection. The mode will be stored in non-volatile memory.

Syntax OUTPut:RI:MODE <mode>
<mode> := {See table **4.3**}

Mode	Description
LATCh	After the detection of level change (from high to low) at the external On/Off control terminals, the power supply output will be switched off. The unit will not output anything until the latch is released.
LIVE	The output state of power supply changes with the level of external On/Off control terminal. When the On/Off control terminal input is high-level, the power supply output is on; and when the external On/Off control terminal input is low-level, the power supply output is off.

Table 4.3 Remote Inhibit Modes

Query OUTPut:RI:MODE?

Example OUTPut:RI:MODE LATCh
OUTPut:RI:MODE?

Response Returns: LATCh {mode}

4.11 OUTPut[n]:SPEed

Description Set the rising voltage slew rate.

Syntax OUTPut<channel>:SPEed <mode>
<mode> :{See table **4.4**}

Mode	Description
NORMal	Allows for a maximum rise time of 5 ms.
FAST	Allows for a maximum rise time of 500 us
TIME	Set the desired rise time. {1 ms to 86400 s}

Table 4.4 Output Speed Modes

Query OUTPut<channel>:SPEed?

Example OUTPut1:SPEed TIME
OUTPut1:SPEed?

Response Returns: TIME {mode}

4.12 OUTPut[n]:SPEed:TIME

Description Set the voltage rise time of the selected channel. In order to select a time **Output Speed** must be set to TIME.

Syntax OUTPut<channel>:SPEed:TIME <time> <time> := {1 ms to 86400 s}

Query OUTPut<channel>:SPEed:TIME?

Example OUTP2:SPE:TIME 10 OUTP2:SPE:TIME?

Response Returns: 10 s {time}

4.13 OUTPut[n]:VOLTage[:DC]:RANGe

Description Set the voltage range.The current range will automatically be assigned based on the voltage range selected.

Syntax OUTPut<channel>:VOLTage[:DC]:RANGe <range>
<channel> := {1 | 2}
<range> := {LOW | HIGH}

High

Channel	Range	Voltage
CH 1	High	-15 to 15 V
CH 2	High	0 to 15 V

Table 4.5 High Voltage Range

Low

Channel	Range	Voltage
CH 1	Low	-9 to 9 V
CH 2	Low	0 to 9 V

Table 4.6 Low Voltage Range

Query OUTPut<channel>:VOLTage[:DC]:RANGe?

Example OUTP1:VOLT:RANG HIGH
OUTP1:VOLT:RANG?

Response Returns: HIGH {range}

4.14 OUTPut[n]:DELay

Description Set an output on delay for the selected channel. Output delay mode must be enabled.

Syntax OUTPut<channel>:DELay <time>
<channel> := {1 | 2}
<range> := {0 to 999.999 s}

Query OUTPut<channel>:DELay?

Example OUTPut2:DELay 10 OUTPut2:DELay?

Response Returns: 10 s {time}

Source Subsystem

The commands under the **Source Subsystem** program various parameters of the outputting: current, voltage, and resistance.

5.1 [SOURce:]CURRent[n][:LEVel][:IMMediate][:AMPLitude]

Description Set the current value of the specified channel. The current value must be within the set range.

Syntax [SOURce:]CURRent<channel>[:LEVel][:IMMediate][:AMPLitude] <value>
<channel> := {1 | 2}
<value> := {Low range: 0 to 3.05 A | High range: 0 to 5.05 A}

Query [SOURce:]CURRent<channel>[:LEVel][:IMMediate][:AMPLitude]?

Example CURRent2 4
CURRent2?

Response Returns: 4 A {value}

5.2 [SOURce:]CURRent:PROTection:STATe

Description Set the Over-Current Protection state. This command is not channel depending. The selected state will apply to both states.

Syntax [SOURce:]CURRent:PROTection:STATe <state>
<state> := {1 | 0 or On | Off}

Query [SOURce:]CURRent:PROTection:STATe?

Example CURR:PROT:STAT 1
CURR:PROT:STAT?

Response Returns: 1 {state}

5.3 [SOURce:]CURRent[n]:TRIGgered

Description Set a preset current value. This preset value will output if a trigger is generated. To generate a trigger, the trigger subsystem must be initialized (in trigger subsystem, INITiate command is used for initialize trigger).

Syntax [SOURce:]CURRent<channel>:TRIGgered
<value> := {Low range: 0 to 3.05 A | High range: 0 to 5.05 A}

Query [SOURce:]CURRent<channel>:TRIGgered?

Example CURRent2:TRIGgered 2
CURRent2:TRIGgered?

Response Returns: 2 A {value}

5.4 [SOURce:]DIGital:DATA

Description Sets the system IO output and controls the two system IO output voltages: High 5V and Low 0V.

Syntax [SOURce:]DIGital:DATA <value>
<value> := {0 | 1 | 2 | 3 }

Query DIGital:DATA?

Example DIGital:DATA 2
DIGital:DATA?

Response Returns: 2 {value}

5.5 [SOURce:]DIGital:FUNCtion

Description Set the mode of the two control ports.

RIDFi

To select the **RIDFI** press **F7**.

Pins 3 and 4 will indicate the power supply fault.

The terminal sources is be divided into QUES, OPER, ESB, RQS and OFF.

- QUES: The EXT RIDFi terminal output level changes with the QUES bit of the state byte of the power supply.
  - When the QUES bit is 0, the RIDFi outputs low level. When the QUES bit is 1, the RIDFi outputs high level.
- OPER: The EXT RIDFi terminal output level changes with the OPER bit of the state byte of the power supply.
- ESB: The EXT RIDFi terminal output level changes with the ESB bit of the state byte of the power supply.
- RQS: The EXT RIDFi terminal output level changes with the RQS bit of the state byte of the power supply.
- OFF: The EXT RIDFi terminal output level keeps low

DIGio

To select the **DIGio** press **F8**.

DIGio1 (Pin 3) and DIGio2 (Pin 4) will be used as the common digital IO output terminals. The terminals can read and control the output state through communication command.

Under remote mode, the SCPI command (DIGital:DATA? and DIGital: DATA) can be sent to read and configure the output terminal state. High level is 5V and low level is 0V.

Syntax [SOURce:]DIGital:FUNCtion <mode>
<mode> := {RIDFi | DIGio}

Query [SOURce:]DIGital:FUNCtion?

Example DIGital:FUNCtion DIGio
DIGital:FUNCtion?

Response Returns: DIGio {mode}

5.6 [SOURce:]CURRent[n]:LIMit:STATe

Description Set the current limit of the selected channel.

Syntax [SOURce:]CURRent<channel>:LIMit:STATe <state>
<channel> := {1 | 2}
<state> := {1 | 0 or On | Off}

Query [SOURce:]CURRent<channel>:LIMit:STATe?

Example CURR2:LIMit:STAT On
CURR2:LIMit:STAT?

Response Returns: 1 {state}

5.7 [SOURce:]CURRent[n]:LIMit

Description Set an upper limit for the current output. The output will not exceed the set value. Unlike OCP outputting the set value will not disable the output nor trigger OCP.

Syntax [SOURce:]CURRent<channel>:LIMit <value>
<value> := {0 to 5.05 A}

Query [SOURce:]CURRent<channel>:LIMit?

Example CURR2:LIMit 4
CURR2:LIMit?

Response Returns: 4 {value}

5.8 [SOURce:]RESistance[n][:LEVel][:IMMediate][:AMPLitude]

Description Set the source internal resistance for the specified channel.

Syntax [SOURce:]RESistance⟨channel⟩[:LEVel][:IMMediate][:AMPLitude] ⟨value⟩
⟨value⟩ := {0 to 1 Ω}

Query [SOURce:]RESistance⟨channel⟩[:LEVel][:IMMediate][:AMPLitude]?

Example RESistance1 .5 RESistance1?

Response Returns: .5 {value}

5.9 [SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude]

Description Set a preset internal resistance to be recalled when a trigger signal is received.

Syntax [SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude] ⟨value⟩
⟨value⟩ := {0 to 1 Ω}

Query [SOURce:]RESistance[:LEVel]:TRIGgered[:AMPLitude]?

Example RESistance:TRIGgered 0
RESistance:TRIGgered?

Response Returns: 0 {value}

5.10 [SOURce:]VOLTage[n][:LEVel][:IMMediate][:AMPLitude]

Description Set Vset for the specified channel. The voltage value must be within the chosen range.

Syntax [SOURce:]VOLTage<channel>[:LEVel][:IMMediate][:AMPLitude] <value>
<channel> := {1 | 2}
<value> := {See tables **4.5** and **4.6**}

High

Channel	Range	Voltage
CH 1	High	-15 to 15 V
CH 2	High	0 to 15 V

Table 5.1 High Voltage Range

Low

Channel	Range	Voltage
CH 1	Low	-9 to 9 V
CH 2	Low	0 to 9 V

Table 5.2 Low Voltage Range

Query [SOURce:]VOLTage<channel>[:LEVel][:IMMediate][:AMPLitude]?

Example VOLT2 7
VOLT2?

Response Returns: 7 {value}

5.11 [SOURce:]VOLTage[n]:PROTection

Description Set the voltage value to trigger over-voltage protection. To configure the value **VOLT:PROT:STAT** must be enabled.

Syntax [SOURce:]VOLTage<channel>:PROTection <value>

Query [SOURce:]VOLTage<channel>:PROTection?

Example VOLTage2:PROT 7
VOLTage2:PROT?

Response Returns: 7 {value}

5.12 [SOURce:]VOLTage[n]:PROTection:STATe

Description Enable/disable over-voltage protection.The unit has a **Hardware Over Voltage Protection** to avoid damage to the internal components when OVP is disabled. The trigger point for Hardware Over Voltage Protectionis approximately 120% of the specified maximum voltage. It's recommended that OVP be enabled.

Syntax [SOURce:]VOLTage<channel>:PROTection:STATe <state>
<channel> := {1 | 2}
<state> := {0 | 1 or On | }

Query [SOURce:]VOLTage<channel>:PROTection:STATe?

Example VOLTage2:PROTection:STATe On
VOLTage2:PROTection:STATe?

Response Returns: 1 {state}

5.13 [SOURce:]VOLTage[n]:TRIGger

Description Set a reset voltage value for the voltage output. The preset value will be recalled when a trigger signal is received. To generate trigger, the trigger subsystem must be initialized (in trigger subsystem, INITiate command is used for initialize trigger).

Syntax [SOURce:]VOLTage<channel>:TRIGger <value>
<channel> := {1 | 2}
<value> := {0 to 15 V}

Query [SOURce:]VOLTage<channel>:TRIGger

Example VOLTage2:TRIGger 5
VOLTage2:TRIGge?

Response Returns: 5 {value}

5.14 [SOURce:]VOLTage[n]:LIMit:STATe

Description Enable/disable voltage limit of the selected channel.

Syntax [SOURce:]VOLTage<channel>:LIMit:STATe <state>
<channel> := {1 | 2}
<value> := {1 | 0 or On | Off}

Query [SOURce:]VOLTage<channel>:LIMit:STATe?

Example VOLTage1:LIMit:STATe On VOLTage1:LIMit:STATe?

Response Returns: 1 {value}

5.15 [SOURce:]VOLTage[n]:LIMit

Description Set the value of the voltage limit. Voltage limit prevent the output from exceed the set value. Unlike OVP it does not disable the output nor trigger OVP once the set value has been reached.

Syntax [SOURce:]VOLTage<channel>:LIMit <value>
<channel> := {1 | 2}
<value> := {0 to 15 V}

Query [SOURce:]VOLTage<channel>:LIMit?

Example VOLTage2:LIMi 10 VOLTage2:LIMi?

Response Returns: 1 {value}

5.16 [SOURce:]VOLTage[n]:MAXSet?

Description Query the maximum settable voltage. The maximum settable voltage will vary depending on the selected voltage range.

Query [SOURce:]VOLTage<channel>:MAXSet?

Example VOLTage2:MAXSet?

Response Returns: {Low Voltage Range: 9.05 V | High Voltage Range: 15.1 V }

5.17 [SOURce:]VOLTage[n]:MINSet?

Description Query the minimum settable voltage. The minimum settable voltage will vary depending on the selected channel.

Query [SOURce:]VOLTage<channel>:MINSet?

Example VOLTage2:MINSet?

Response Returns: {Channel 1: -15.1 V | Channel 2: 15.1 V }

5.18 [SOURce:]CURRent[n]:MAXSet?

Description Query the maximum settable current. The maximum settable current will vary depending on the selected voltage range.

Query [SOURce:]CURRent<channel>:MAXSet?

Example CURRent2:MAXSet?

Response Returns: {Low Voltage Range: 5.05 A | High Voltage Range: 3.05 A }

5.19 [SOURce:]CURRent[n]:MINSet?

Description Query the minimum settable current.

Query [SOURce:]CURRent<channel>:MINSet?

Example CURR2:MINSet?

Response Returns: {.002 A for both channels }

Measure Subsystem

The **Measure Subsystem** consist of commands to control all measurement aspects of the instrument. Measure commands measure the output voltage or current. Measurements are performed by digitizing the instantaneous output voltage or current for a specified number of samples, storing the results in a buffer, and calculating the measured result. Two types of measurement commands are available: **MEASure** and **FETCh**.

MEASure commands trigger the acquisition of new data before returning the reading.

FETCh commands return a reading computed from previously acquired data.

Use the **MEASure** commands when the measurement does not need to be synchronized with any other event. **FETCh** commands should be used when it is important that the measurement be synchronized with either a trigger or with a particular part of the output.

To set the time interval of the collected data see section SENSe:SWEep:TINTerval. The position of the trigger relative to the beginning of the data buffer is determined by SENSe:SWEep:OFFSet. The number of points returned is set by **SENSe:SWEep:POINts**.

6.1 MEASure:ARRay:CURRent[n][:DC]? FETCh:ARRay:CURRent[n][:DC]?

Description Query an array containing the output current of the selected channel. The return data will vary depending on the sweep configuration.

Query MEASure:ARRay:CURRent[n][:DC]?
FETCh:ARRay:CURRent[n][:DC]?

Example MEAS:ARR:CURR2?
FETC:ARR:CURR2?

Response Returns:**example of a 20 point array**

```
4.33114e-06, 3.83147e-06, 4.16458e-06, 4.33114e-06, 4.16458e-06, 3.66491e-06,
3.83147e-06, 4.49769e-06, 3.66491e-06, 3.99802e-06, 4.33114e-06, 4.49769e-06,
4.16458e-06, 3.66491e-06, 3.99802e-06, 3.83147e-06, 4.33114e-06, 3.99802e-06,
4.49769e-06, 4.33114e-06,
```

6.2 MEASure:ARRay:VOLTage[n][:DC]? FETCh:ARRay:VOLTage[n][:DC]?

Description Query an array ontaining the measured voltage of the selected channel. The return data will vary depending on the sweep configuration.

Query MEASure:ARRay:VOLTage<channel>[:DC]?
FETCh:ARRay:VOLTage<channel>[:DC]?

Example MEAS:ARR:VOLT2? FETC:ARR:VOLT2?

Response Returns: **example of a 50 point array**

```
3.01106, 3.01131, 3.01082, 3.01131, 3.01106, 3.01106, 3.01106, 3.01082, 3.01131,
3.01082, 3.01106, 3.01131, 3.01131, 3.01082, 3.01106, 3.01082, 3.01131, 3.01082,
3.01131, 3.01131, 3.01131, 3.01131, 3.01131, 3.01131, 3.01106, 3.01106, 3.01131,
3.01106, 3.01106, 3.01106, 3.01131, 3.01106, 3.01106, 3.01082, 3.01106, 3.01131,
3.01106, 3.01082, 3.01156, 3.01082, 3.01106, 3.01131, 3.01106, 3.01106, 3.01106,
3.01106, 3.01082, 3.01131, 3.01131, 3.01106,
```

6.3 MEASure:CURRent[n]? FETCh:CURRent[n]?

Description Query the output current of the selected channel.

Query MEASure:CURRent[n]?
FETCh:CURRent[n]?

Example MEAS:CURR?
FETC:CURR?

Response Returns:0.990765 A

6.4 MEASure:CURRent[n]:ACDC? FETCh:CURRent[n]:ACDC?

Description Query the AC + DC rms output current.

Query MEASure:CURRent<channel>:ACDC?
FETCh:CURRent<channel>:ACDC?

Example MEAS:CURR2:ACDC?
FETC:CURR2:ACDC?

Response Returns: 4.89132e-06 A

6.5 MEASure:CURRent[n]:HIGH? FETCh:CURRent[n]:HIGH?

Description Query the High level current of a current pulse waveform. The instrument first measures the minimum and maximum data points of the pulse waveform. It then generates a histogram of the pulse waveform using 16 bins between the maximum and minimum data points. The bin containing the most data points above the 50% point is the high bin. The average of all the data points in the high bin is returned as the High level. If no high bin contains more than 1.25% of the total number of acquired points, then the maximum value is returned by these queries.

Query MEASure:CURRent<channel>:HIGH?
FETCh:CURRent<channel>:HIGH?

Example MEAS:CURR:HIGH?
FETC:CURR:HIGH?

Response Returns: 0.991245 A

6.6 MEASure:CURRent[n]:LOW? FETCh:CURRent[n]:LOW?

Description Query the Low level current of a current pulse waveform. The instrument first measures the minimum and maximum data points of the pulse waveform. It then generates a histogram of the pulse waveform using 16 bins between the maximum and minimum data points. The bin containing the most data points above the 50% point is the low bin. The average of all the data points in the low bin is returned as the low level. If no low bin contains more than 1.25% of the total number of acquired points, then the maximum value is returned by these queries.

Query MEASure:CURRent<channel>:LOW?
FETCh:CURRent<channel>:LOW?

Example MEAS:CURR2:LOW?
FETC:CURR2:LOW?

Response Returns: 3.3318e-06 A

6.7 MEASure:CURRent[n]:MAXimum? FETCh:CURRent[n]:MAXimum?

Description Query the maximum current measured in the sample.

Query MEASure:CURRent<channel>:MAXimum?
FETCh:CURRent<channel>:MAXimum?

Example MEAS:CURR:MAX?
FETC:CURR:MAX?

Response Returns: 0.991405 A

6.8 MEASure:CURRent[n]:MINimum? FETCh:CURRent[n]:MINimum?

Description Query the mnimum current measured in the sample.

Query MEASure:CURRent<channel>:MINimum?
FETCh:CURRent<channel>:MINimum?

Example MEAS:CURR:MIN?
FETC:CURR:MIN?

Response Returns: 0.989804 A

6.9 MEASure:DVM[n]? FETCh:DVM[n]?

Description Query the measured dc voltage.

Query MEASure:DVM<channel>?
FETCh:DVM<channel>?

Example MEAS:DVM2?
FETC:DVM2?

Response Returns: 0.0152739 V

6.10 MEASure:DVM[n]:ACDC? FETCh:DVM[n]:ACDC?

Description Query the measred AC + DC rms voltage.

Query MEASure:DVM<channel>:ACDC?
FETCh:DVM<channel>:ACDC?

Example MEAS:DVM2:ACDC?
FETC:DVM2:ACDC?

Response Returns: 0.0152739 V

6.11 MEASure:VOLTage[n]? FETCh:VOLTage[n]?

Description Query the output voltage.

Query MEASure:VOLTage<channel>?
FETCh:VOLTage<channel>?

Example MEAS:VOLT2?
FETC:VOLT2?

Response Returns: 4.01212 V

6.12 MEASure:VOLTage[n]:ACDC? FETCh:VOLTage[n]:ACDC?

Description Query the AC + DC rms output voltage.

Query MEASure:VOLTage<channel>:ACDC?
FETCh:VOLTage<channel>:ACDC?

Example MEAS:VOLT2:ACDC?
FETC:VOLT2:ACDC?

Response Returns: 0.00474167 Vrms

6.13 MEASure:VOLTage[n]:HIGH? FETCh:VOLTage[n]:HIGH?

Description Query the High level VOLTage of a VOLTage pulse waveform. The instrument first measures the minimum and maximum data points of the pulse waveform. It then generates a histogram of the pulse waveform using 16 bins between the maximum and minimum data points. The bin containing the most data points above the 50% point is the high bin. The average of all the data points in the high bin is returned as the High level. If no high bin contains more than 1.25% of the total number of acquired points, then the maximum value is returned by these queries.

Query MEASure:VOLTage<channel>:HIGH?
FETCh:VOLTage<channel>:HIGH?

Example MEAS:VOLT:HIGH?
FETC:VOLT:HIGH?

Response Returns: 4.01286 V

6.14 MEASure:VOLTage[n]:LOW? FETCh:VOLTage[n]:LOW?

Description Query the Low level VOLTage of a VOLTage pulse waveform. The instrument first measures the minimum and maximum data points of the pulse waveform. It then generates a histogram of the pulse waveform using 16 bins between the maximum and minimum data points. The bin containing the most data points above the 50% point is the low bin. The average of all the data points in the low bin is returned as the low level. If no low bin contains more than 1.25% of the total number of acquired points, then the maximum value is returned by these queries.

Query MEASure:VOLTage<channel>:LOW?
FETCh:VOLTage<channel>:LOW?

Example MEAS:VOLT2:LOW?
FETC:VOLT2:LOW?

Response Returns: 0.00420573 V

6.15 MEASure:VOLTage[n]:MAXimum? FETCh:VOLTage[n]:MAXimum?

Description Query the maximum VOLTage measured in the sample.

Query MEASure:VOLTage<channel>:MAXimum?
FETCh:VOLTage<channel>:MAXimum?

Example MEAS:VOLT:MAX?
FETC:VOLT:MAX?

Response Returns: 4.01286 V

6.16 MEASure:VOLTage[n]:MINimum? FETCh:VOLTage[n]:MINimum?

Description Query the mnimum VOLTage measured in the sample.

Query MEASure:VOLTage<channel>:MINimum?
FETCh:VOLTage<channel>:MINimum?

Example MEAS:VOLT:MIN?
FETC:VOLT:MIN?

Response Returns: 4.01188 V

6.17 MEASure[:SCALar]:POWer[n][:DC]? FETCh[:SCALar]:POWer[n][:DC]?

Description Query the power of the selected channel.

Query MEASure[:SCALar]:POWer<channel>[:DC]?
FETCh[:SCALar]:POWer<channel>[:DC]?

Example MEAS:POW2?
FETC:POW2?

Response Returns: 3.14102

Sense Subsystem

The commands listed in the **Sense Subsystem** program the filters for data acquisition.

7.1 SENSe:CURRent[n]:DETector

Description Select the detector to be used when making output current measurements.

Syntax SENSe:CURRent<channel>:DETector <detector>
<channel> := {1 | 2}
<detector> := {ACD | DC}

ACDC

Suitable for all dynamic current measurements. It is especially important to use ACDC detection when measuring pulse or other waveforms with frequency contents greater than several kilohertz.

DC

Used for dc current measurements requiring an offset accuracy better than 2mA on the High current measurement range.

Query SENSe:CURRent<channel>:DETector?

Example SENS:CURR2:DET DC
SENS:CURR2:DET?

Response Returns: DC {detector}

7.2 SENSe[n]:CURRent:MODE

Description Set the current sampling mode.

Syntax SENSe<channel>:CURRent:MODE <mode>
<channel> := {1 | 2}
<mode> := {AUTo | HIGH}

Auto

When current is lower than 5 mA, sampling will automatically switch to low-current measurement. When current is higher than 5 mA, sampling will automatically switch to high-current measurement

HIGH

Sampling is always on high-current.

Query SENSe<channel>:CURRent:MODE?

Example SENS2:CURR:MODE AUTo SENS2:CURR:MODE?

Response Returns: AUTO

7.3 SENSe[n]:FUNCtion

Description Set and query the sensing function for the triggered measurements.

Syntax SENSe<channel>:FUNCtion <mode>
<channel> := {1 | 2}
<mode> := {"VOLTage" | "CURRent" | "DVM"}

Mode	Description
"VOLTage"	Senses the current measurement at the instrument outputs.
"CURRent"	Senses the voltage measurement at the DVM inputs.
"DVM"	Senses the voltage measurement at instrument the outputs.

Table 7.1 Sense Function Modes

Query SENSe<channel>:FUNCtion?

Example SENS2:FUNC "CURRent"
SENS2:FUNC?

Response Returns: CURRent {mode}

7.4 SENSe[n]:SWEep:POINts

Description Set the number of points to be colleted in each measurement.

Syntax SENSe<channel>:SWEep:POINts <points>
<channel> := {1 | 2}
<points> := {0 to 600}

Query SENSe<channel>:SWEep:POINts?

Example SENS2:SWE:POINt 50

Response Returns: 50 {points}

7.5 SENSe[n]:SWEep:TINTerval

Description Set the time period between samples. The time interval will be rounded to the nearest 33.33 mircrosecond increment.

Syntax SENSe<channel>:SWEep:TINTerval <time>
<channel> := {1 | 2}
<time> := {33.33 ms to 1 s}

Query SENSe<channel>:SWEep:TINTerval?

Example SENSe2:SWE:TINT 30e-5 SENSe2:SWE:TINT?

Response Returns: 0.0003 s {time}

7.6 SENSe[n]:SWEep:FREQuency

Description Set the sampling rate.

Syntax SENSe<channel>:SWEep:FREQuency
<channel> := {1 | 2}
<value> := {1 to 30000}

Query SENSe<channel>:SWEep:FREQuency?

Example SENSe2:SWE:FREQ 1e3
SENSe2:SWE:FREQ?

Response Returns: 1000 {value}

7.7 SENSe[n]:WINDow

Description Set the window function that will be used in DC and DC+AC rms measurement calculations.

Average

The average filter takes an average of the collected measurements. The number of colllected measurements can be specified by configuring the **Sample Rate** and the **Sample Count**.

Hanning

The Hanning filter is a low pass filter that is very effective in reducing noise. The filter reaches zero quickly allowing it to filter noise, but also causing it to lose the edges of the signal.

Rectangle

The Rectangle filter is a low pass filter that is very effective in reducing noise. The filter reduces noise without losing the edges of the signal.
Press the **F9** softkey when the **Filter Type** setting is selected.

Syntax SENSe<channel>:WINDow[:TYPE] <type>
<channel> := {1 | 2}
<Type> := {HANNing | RECTangular | AVERaging}

Query SENSe<channel>:WINDow[:TYPE]?

Example SENS2:WIND AVER
SENS2:WIND?

Response Returns: AVERaging {type}

7.8 SENSe[n]:WINDow:STATe

Description Enable/disable the window function.

Syntax SENSe[n]:WINDow:STATe <state>
<channel> := {1 | 2}
<state> := {1 | 0 or On | Off}

Query SENSe<channel>:WINDow:STATe?

Example SENS2:WIND:STAT On
SENS2:WIND:STAT?

Response Returns: 1 {state}

7.9 SENSe[n]:SAVE

Description Save the **SENSE** setting of the selected channel.

Syntax SENSe[n]:SAVE

Example SENS2:SAVE

Graph

The commands listed in the **SCOPe Subsystem** all parameters of the scope function.

8.1 SCOPe:VOLTage:RANGe

Description Set the volage range of the graph function. The command applies to both channel.

Syntax SCOPe:VOLTage:RANGe <range>
<range> := {0.1 | 0.2 | 0.5 | 1 | 2 | 5 | 10}

Query SCOPe:VOLTage:RANGe?

Example SCOP:VOLT:RANG 1
SCOP:VOLT:RANG?

Response Returns: 1 Write or read the range of the graph function. The commands applies to both channel. {range}

8.2 SCOPe:CURRent:RANGe

Description Set the current range of the graph function. The command applies to both channel.

Syntax SCOPe:CURRent:RANGe <state>
<range> := {0.005 | 0.01 | 0.02 | 0.05 | 0.1 | 0.2 | 0.5 | 1 | 2}

Query SCOPe:CURRent:RANGe?

Example SCOP:CURR:RANG 2
SCOP:CURR:RANG?

Response Returns: 2 A {range}

8.3 SCOPe:TIME:BASE

Description Set the time base of the graph function. The time base is set in ms.

Syntax SCOPe:TIME:BASE <time>
<time> := {1 | 2 | 5 | 10 | 20 | 50 | 100 | 200 | 500}

Query SCOPe:TIME:BASE?

Example SCOP:TIME:BASE? 1
SCOP:TIME:BASE?

Response Returns: 1 ms {time}

8.4 SCOPe:RUNStop

Description Enable/disable the runstop state of the graph function. If enabled { 1 or On} the run will stop. If disabled { 0 or Off} the graph function will continuously update.

Syntax SCOPe:RUNStop <state>
<state> := {1 | 0 or On | Off}

Query SCOPe:RUNStop?

Example SCOP:RUNS On SCOP:RUNS?

Response Returns: 1 {state}

8.5 SCOPe:AUTO

Description Set the graph function to automatic adaption mode.

Syntax SCOPe:AUTO

Example SCOP:AUTO

8.6 SCOPe[n]:SHOW

Description Enable/disable the measurements to be displayed in the graph function. The parameters are controlled by 3 bits each bit indicating UAD (Voltage, Current, DVM).

Syntax SCOPe<channel>:SHOW <state>
<channel> := {1 | 2}
<state> := { 0 to 7 see table 8.1}

3 Bits	state	U	A	D
000	0	X	X	X
001	1	X	X	✓
010	2	X	✓	X
011	3	X	✓	✓
100	4	✓	X	X
101	5	✓	X	✓
110	6	✓	✓	X
111	7	✓	✓	✓

Table 8.1 UAD State

Query SCOPe<channel>:SHOW?

Example SCOP2:SHOW 1
SCOP2:SHOW?

Response Returns: 1

8.7 SCOPe[n]:VOLTage:BASE

Description Set the voltage position of the selected channel in the graph display.

Syntax SCOPe<channel>:VOLTage:BASE <position> <channel> := {1 | 2}
<position> := { -15 to 15 }

Query SCOPe<channel>:VOLTage:BASE

Example SCOP2:VOLT:BASE 0
SCOP2:VOLT:BASE?

Response Returns: 0 {position}

8.8 SCOPe[n]:CURRent:BASE

Description Set the current position of the selected channel in the graph display.

Syntax SCOPe<channel>:CURRent:BASE
<channel> := {1 | 2}
<position> := { -15 to 15 }

Query SCOPe<channel>:CURRent:BASE?

Example SCOPe2:CURRent:BASE -10
SCOPe2:CURRent:BASE?

Response Returns: -10 {position}

8.9 SCOPe[n]:DVM:BASE

Description Set the DVM position of the selected channel in the graph display.

Syntax SCOPe<channel>:DVM:BASE <position>
<channel> := {1 | 2}
<position> := { -15 to 15 }

Query SCOPe<channel>:DVM:BASE?

Example SCOP2:DVM:BASE 5
SCOP2:DVM:BASE?

Response Returns: 5 {position}

8.10 SCOPe[n]:VOLTage:TRIGer

Description Set the voltage trigger level of the graph function.

Syntax SCOPe<channel>:VOLTage:TRIGer <trigger>
<channel> := {1 | 2}
<trigger> := { -15.1 to 15.1 }

Query SCOPe<channel>:VOLTage:TRIGer?

Example SCOP2:VOLT:TRIG 5
SCOP2:VOLT:TRIG?

Response Returns: 5

8.11 SCOPe[n]:CURRent:TRIGer

Description Set the current trigger level of the graph function.

Syntax SCOPe<channel>:CURRent:TRIGer <trigger>
<channel> := {1 | 2}
<trigger> := { -5.05 to 5.05 }

Query SCOPe<channel>:CURRent:TRIGer?

Example SCOP2:CURR:TRIG .5
SCOP2:CURR:TRIG?

Response Returns: .5

8.12 SCOPe[n]:DVM:TRIGer

Description Set the DVM trigger level of the graph function.

Syntax SCOPe<channel>:DVM:TRIGer <trigger>
<channel> := {1 | 2}
<trigger> := { -30 to 30 }

Query SCOPe<channel>:DVM:TRIGer?

Example SCOP2:DVM:TRIG 5
SCOP2:DVM:TRIG?

Response Returns: 5

8.13 SCOPe[n]:TRIGer:SOURce

Description Set the trigger source type of the selected channel for the graph function.

Syntax SCOPe<channel>:TRIGer:SOURce <source>
<channel> := {1 | 2}
<source> := { VOLTage | CURRent | DVM}

Query SCOPe<channel>:TRIGer:SOURce?

Example SCOP2:TRIG:SOUR VOLT
SCOP2:TRIG:SOUR?

Response Returns: VOLTage {source}

8.14 SCOPe[n]:TRIGer:MODE

Description Set the trigger mode of the selected channel for the graph function.

Syntax SCOPe<channel>:TRIGer:MODE <mode>
<channel> := {1 | 2}
<mode> := { AUTO | NORM | SINGle}

Query SCOPe<channel>:TRIGer:MODE

Example SCOP2:TRIG:MODE AUTO
SCOP2:TRIG:MODE?

Response Returns: AUTO {mode}

8.15 SCOPe[n]:TRIGer:SLOPe

Description Set the triggering edge of the selected channel for the graph function

Syntax SCOPe<channel>:TRIGer:SLOPe <slope>
<channel> := {1 | 2}
<slope> := { UP | DOWN | UPDOwn}

Query SCOPe<channel>:TRIGer:SLOPe

Example SCOP2:TRIG:SLOP UP
SCOP2:TRIG:SLOP?

Response Returns: UP {slope}

8.16 SCOPe:TRIGer:DELay

Description Set a trigger delay in the graph function. The delay is set in ms.

Syntax SCOPe:TRIGer:DELay <time>
<time> := {-2000 to 2000}

Query SCOPe:TRIGer:DELay?

Example SCOP:TRIG:DEL 10 SCOP:TRIG:DEL?

Response Returns: 10 ms {time}

8.17 SCOPe[n]:DATA?

Description Query the data of the waveform displayed in the selected channel. Reply 900 float data at one time

Query SCOPe<channel>:DATA?

Example SCOPe<channel>:DATA?

Response Returns: #40906 | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |
r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |
r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |
r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |
r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |
r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |
r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |
r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |
r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |
r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Battery Subsystem

The commands listed in the **Battery Subsystem** program all available parameters for the battery function.

9.1 BATTery[n]:GROup

Description Set the battery group.

Syntax BATTery<channel>:GROup <group>
<channel> := {1 | 2}
<group> := {0 to 19}

Query BATTery<channel>:GROup?

Example BATT2:GRO 5

Response Returns: 5 {group}

9.2 BATTery[n]:POINt

Description Select a point in the selected group.

Syntax BATTery<channel>:POINt <point>
<channel> := {1 | 2}
<point> := {0 to 19}

Query BATTery<channel>:POINt?

Example BATT2:POIN 1

Response Returns: 1 {point}

9.3 BATTery[n]:TOTal

Description Set the number of points to be available in the selected group.

Syntax BATTery<channel>:TOTal <points>
<channel> := {1 | 2}
<points> := {0 to 19}

Query BATTery<channel>:TOTal ?

Example BATTery2:TOT 19

Response Returns:

9.4 BATTery[n]:PARamneter

Description Set the 3 parameters of the selected point.

Syntax BATTery<channel>:PARamneter <capacity>,<voltage>,<resistance>
<capacity> :={0 to 99999.9}
<voltage> :={0 to 15.1 }
<resistance> :={0 to 1}

Query BATTery<channel>:PARamneter?

Example BATT2:PAR 100,10,.1
BATT2:PAR?

Response Returns: 100,10,.1 {capacity,voltage,resistance}

9.5 BATTery[n]:VOLTage:SHUT

Description Set the voltage shut-off value.

Syntax BATTery<channel>:VOLTage:SHUT <voltage>
<channel> := {1 | 2}
<voltage> := {0 to 15.1 V}

Query BATTery<channel>:VOLTage:SHUT

Example BATT:VOLT:SHUT 12 BATT:VOLT:SHUT?

Response Returns: 12 V{voltage}

9.6 BATTery[n]:VOLTage:SHUT:STATe

Description Enable/disable the voltage shut-off state.

Syntax BATTery<channel>:VOLTage:SHUT:STATe <state>
<channel> := {1 | 2}
<state> := {1 | 0 or On | Off}

Query BATTery<channel>:VOLTage:SHUT:STATe?

Example BATT:VOLT:SHUT:STATe On
BATT:VOLT:SHUT:STATe?

Response Returns: 1 {state}

9.7 BATTery[n]:CAPacity:SHUT

Description Set the capacity shut-off value.

Syntax BATTery<channel>:CAPacity:SHUT <capacity>
<channel> := {1 | 2}
<capacity> := {0 to 99999.9 C}

Query BATTery<channel>:CAPacity:SHUT?

Example BATTery2:CAP:SHUT 10 BATTery2:CAP:SHUT?

Response Returns: 10 {capacity}

9.8 BATTery[n]:CAPacity:SHUT:STATe

Description Enable/disable the capacity shut-off state.

Syntax BATTery<channel>:CAPacity:SHUT:STATe <state>
<channel> := {1 | 2}
<state> := {1 | 0 or On | Off}

Query BATTery<channel>:CAPacity:SHUT:STATe?

Example BATT:CURR:SHUT:STATe On
BATT:CURR:SHUT:STATe?

Response Returns: 1 {state}

9.9 BATTery[n]:CURRent:SHUT

Description Set the curreent shut-off value.

Syntax BATTery<channel>:CURRent:SHUT <current>
<channel> := {1 | 2}
<current> := {0 t0 5.05 A}

Query BATTery<channel>:CURRent:SHUT?

Example BATTery2:CURR:SHUT 1 BATTery2:CURR:SHUT?

Response Returns: 1 {CURRent}

9.10 BATTery[n]:CURRent:SHUT:STATe

Description Enable/disable the current shut-off state.

Syntax BATTery<channel>:CURRent:SHUT:STATe <state>
<channel> := {1 | 2}
<state> := {1 | 0 or On | Off}

Query BATTery<channel>:CURRent:SHUT:STATe?

Example BATT:CAP:SHUT:STATe On
BATT:CAP:SHUT:STATe?

Response Returns: 1 {state}

9.11 BATTery[n]:TIME:SHUT

Description Set the run time of the battery function. The selected operation will end once the set time has been elapsed.

Syntax BATTery<channel>:TIME:SHUT <time>
<channel> := {1 | 2}
<time> := {0 to 999999 s}

Query BATTery<channel>:TIME:SHUt?

Example BATT:TIME:SHUT 300 BATT:TIME:SHUT?

Response Returns:

9.12 BATTery[n]:TIME:SHUT:STATe

Description Enable/disable the time shut-off state.

Syntax BATTery<channel>:TIME:SHUT:STATe <state>
<channel> := {1 | 2}
<state> := {1 | 0 or On | Off}

Query BATTery<channel>:TIME:SHUT:STATe?

Example BATT:TIME:SHUT:STATe On
BATT:TIME:SHUT:STATe?

Response Returns: 1 {state}

9.13 BATTery[n]:STARt

Description Start the selected operation on the selected channel.

Syntax BATTery[n]:STARt
<channel> := {1 | 2}

Example BATTery2:STARt

9.14 BATTery[n]:STOP

Description Stop the selected operation on the selected channel.

Syntax BATTery[n]:STOP
<channel> := {1 | 2}

Example BATTery2:STOP

9.15 BATTery[n]:CAPacity:CLEar

Description Clear the capacity measurement of the battery.

Syntax BATTery<channel>:CAPacity:CLEar
<channel> := {1 | 2}

Example BATTery2:CAP:CLE

9.16 BATTery[n]:SAVE

Description Save the parameters of the selected group.

Syntax BATTery<channel>:SAVE
<channel> := {1 | 2}

Example BATT2:SAVE

9.17 BATTery[n]:CLEar

Description Clear all parameters of the selected group. The group will be blank, no default point will be assigned.

Syntax BATTery<channel>:CLEar
<channel> := {1 | 2}

Example BATT2:CLEar

9.18 BATTery[n]:RECall:SELect

Description Recall a previously saved group.

Syntax BATTery<channel>:RECall:SELect <group>
<channel> := {1 | 2} <group> := {0 to 19}

Example BATT2:REC:SEL 1

9.19 BATTery[n]:MODE

Description Set the operation mode of the selected channel.

Syntax BATTery<channel>:MODE <mode>
<channel> := {1 | 2}
<mode> := {CHARge | DISCharge | SIMulator}

Query BATTery<channel>:MODE

Example BATT2:MODE CHARge
BATT2:MODE?

Response Returns: CHARge {mode}

9.20 BATTery[n]:VOLTage

Description Set the charge or discharge voltage value of the selected channel.

Syntax BATTery<channel>:VOLTage <voltage> <channel> := {1 | 2}
<voltage> := {-15.1 to 15.1 V}

Query BATTery<channel>:VOLTage?

Example BATT:VOLT 12
BATT:VOLT?

Response Returns: 12 V {voltage}

9.21 BATTery[n]:CURRent

Description Set the charge or discharge current value of the selected channel.

Syntax BATTery<channel>:CURRent <current> <channel> := {1 | 2}
<current> := { 0 to 5.05 V}

Query BATTery<channel>:CURRent?

Example BATT:CURR 1
BATT:CURR?

Response Returns: 1 V {current}

List Subsystem

The commands listed in the **List Subsystem** provide remote access for configuration of list mode.

10.1 LIST[n]:GROup

Description Select a list group.

Syntax LIST<channel>:GROup <channel>
<channel> := {1 | 2}
<group> := {0 to 19}

Query LIST<channel>:GROup?

Example LIST2:GRO 2
LIST2:GRO?

Response Returns: 2 {group}

10.2 LIST[n]:PERiod

Description Set the times of times the selected group runs. If 0 is selected the list will not run at all.

Syntax LIST<channel>:PERiod <cycles>
<channel> := {1 | 2}
<cycles> := { 0 to 65535}

Query LIST<channel>:PERiod?

Example LIST2:PERiod10
LIST2:PERiod?

Response Returns: 10 {cycles}

10.3 LIST[n]:TOTal

Description Set the number of points to be programmed in the selected group.

Syntax LIST<channel>:TOTal <points>
<channel> := {1 | 2}
<points> := { 0 to 19}

Query LIST<channel>:TOTal?

Example LIST2:TOT 10

Response Returns: 19 {points}

10.4 LIST[n]:POINt

Description Sets the current point of current group.

Syntax LIST<channel>:POINt <point>
<channel> := {1 | 2}
<point> := { 0 to 19}

Query LIST<channel>:POINt?

Example LIST2:POIN 3

Response Returns: 3 {point}

10.5 LIST[n]:PARamneter

Description Sets 4 parameters of the current point of the current group, i.e., voltage, current, internal resistance and duration

Syntax LIST[n]:PARamneter <voltage>,<current>,<resistance>,<dwell time>
<voltage> :={0 to 15.1 V}
<current> :={0 to 5.05 A}
<resistance> :={0 to 1 Ω}
<dwell time> :={0 to 86400 s}

Query LIST[n]:PARamneter?

Example LIST2:PAR 12,1,1,60 LIST2:PAR?

Response Returns: 12,1,1,60 {voltage,current,resistance,dwell time}

10.6 LIST[n]:GROup:SELect

Description Enable the selected group in the sequence sel.

Syntax LIST<channel>:GROup:SELect <group>
<channel> := {1 | 2}
<group> := { 0 to 19}

Example LIST:GRO:SEL 1

10.7 LIST[n]:GROup:CLEar:SELect

Description Sets to clear the enable status of all groups.

Syntax LIST<channel>:GROup:CLEar:SELect
<channel> := {1 | 2}

Example LIST:GRO:CLE:SEL

10.8 LIST[n]

Description Enable/disable the list function of the selected channel.

Syntax LIST<channel> <state>
<channel> := {1 | 2}
<state> := {1 | 0 or On | Off}

Query LIST<channel> <state>?

Example LIST 2 1
LIST?

Response Returns: 1<state>

10.9 LIST[n]:RUN:STATe?

Description Retruns the state of the list operation.

Query LIST[n]:RUN:STATe?
<channel> := {1 | 2}

Example LIST2:RUN:STAT?

Response Returns: 0 or 1

10.10 LIST:TRIGer

Description Set the trigger to both channel list trigger.

Syntax LIST:TRIGer

Example LIST:TRIG

10.11 LIST[n]:STEP?

Description Returns the point and group currently running in the selected channel. If the list is not running it will return 0,0.

Query LIST<channel>:STEP?
<channel> := {1 | 2}

Example LIST:STEP?

Response Returns: 0,0

10.12 LIST[n]:SAVE

Description Save all parameters of the assigned group in the selected channel.

Syntax LIST<channel>:SAVE
<channel> := {1 | 2}

Example LIST2:SAVE

10.13 LIST[n]:TRIGer:ENABle

Description Enable the list channel trigger.

Syntax LIST<channel>:TRIGer:ENABle

Example LIST2:TRIG:ENABle
To enable both channels {**All**}: LIST1:TRIG:ENABle;:LIST2:TRIr:ENABle

10.14 LIST[n]:TRIGer:DISable

Description Disable the list channel trigger.

Syntax LIST<channel>:TRIGer:DISable

Example LIST2:TRIG:DISable

Initiate Subsystem

11.1 INITiate:SEQuence INITiate:NAME

Description Enable the output and measurement triggers. When a trigger is enabled, an event on the selected trigger source causes the specified triggering action to occur. If the trigger subsystem is not enabled, all triggers are ignored.

Syntax INITiate[:IMMediate]:SEQuence [1 | 2]
INITiate[:IMMediate]:NAME<name>

Example NIT:SEQ2
INIT:NAME TRAN

11.2 INITiate:CONTinuous:SEQuence1 INITiate:CONTinuous:NAME TRANsient

Description Enable/disable continuous triggering. If disabled the, INItiate:SEQuence1 must be used
to initiate all triggers.

Syntax INITiate:CONTinuous:SEQuence1 <state>
INITiate:CONTinuous:NAME TRANsient <state>
<state> := {1 | 0 or On | Off}

Example INITiate2:CONTinuous:SEQuence1 On
INITiate2:CONT:NAME TRAN 1

Trigger Subsystem

The commands listed in the **TRIGger Subsytem** provide remote access for configuration of all trigger parameters.

12.1 TRIGger

Description Generate a BUS trigger for the output transient trigger system.

If the transient trigger system is enabled, the trigger will then:

1. Initiate a pending level change specified by CURRent:TRIGger or VOLTage:TRIGger.

2. Clear the WTG bit in the Status Operation Condition register after both transient and acquire trigger sequences have completed. (WTG is the logical-or of both transient and acquire sequences.)

3. If INITiate:CONTinuous is enabled, the trigger subsystem is immediately re-enabled for subsequent triggers. As soon as it is cleared, the WTG bit is again set to 1.

Syntax TRIGger[:SEQuence1][:IMMediate]

Example TRIG

12.2 TRIGger[n]:SOURce

Description Set the trigger source for the transient trigger system. Trigger source must be set to BUS to trigger commands remotely.

Syntax TRIGger<channel>:SOURce <mode>
<channel> := {1 | 2} {INTernal | BUS | EXTernal}

Internal

Events are triggerd by pressing the **Trig** button in the front pannel.

BUS

Commands sent through the selected remote interface will trigger events.

External

External signals received through pins 1 and 6 of the **System I/O Interface** (DB9 terminal) will trigger events. A trigger will take place when a descending edge is registered.

Query TRIGger<channel>:SOURce ?

Example TRIG:SOUR BUS TRIG:SOUR?

Response Returns: BUS{mode}

12.3 TRIG[n]:SEQ2 TRIG[n]:ACQ

Description Generate a BUS trigger for the measurement trigger system.

When the measurement trigger system is enabled, the measurement trigger triggers a measurement of the selected signal and store the results in a buffer. **SENSe:FUNCtion** selects the signal.

Syntax TRIGger<channel>:SEQuence2
TRIGger<channel>:ACQuire

Example TRIG2:SEQ2 TRIG2:ACQ

12.4 TRIG[n]:SEQ2:LEV:CURR TRIG[n]:ACQ:LEV:CURR

Description Set the trigger level for internally triggered current measurements. A positive current trigger occurs when the current level changes from a value less than the lower hysteresis band limit to a value greater than the upper hysteresis band limit. Similarly, a negative current trigger occurs when the current level changes from a value greater than the upper hysteresis band limit to a value less than the lower hysteresis band limit.

Syntax

```
TRIGger<channel>:SEQuence2:LEVel:CURRent <current>
TRIGger<channel>:ACQuire:LEVel:CURRent <current>

<channel> := {1 | 2}
<current> := {0 to 5.05 A}
```

Query

```
TRIGger<channel>:SEQuence2:LEVel:CURRent
TRIGger<channel>:ACQuire:LEVel:CURRent
```

Example

```
TRIG:SEQ2:LEV:CURR .5
TRIG:ACQ:LEV:CURR 1
TRIG:SEQ2:LEV:CURR?
TRIG:ACQ:LEV:CURR?
```

Response

```
Returns: .5
1 A
```

12.5 TRIG[n]:SEQ2:LEV:DVM TRIG[n]:ACQ:LEV:DVM

Description Set the trigger level for DVM measurements.
A positive trigger occurs when the input signal changes from a value less than the lower hysteresis band limit to a value greater than the upper hysteresis band limit.

A negative trigger occurs when the input signal changes from a value greater than the upper hysteresis band limit to a value less than the lower hysteresis band limit

Syntax TRIGger<channel>:SEQuence2:LEVel:DVM <voltage>
TRIGger<channel>:ACQuire:LEVel:DVM <voltage>

<channel> := {1 | 2}
<voltage> := {0 to 30 v}

Query TRIGger<channel>:SEQuence2:LEVel:DVM?
TRIGger<channel>:ACQuire:LEVel:DVM?

Example TRIG:SEQ2:LEV:DVM 10 TRIG:SEQ2:LEV:DVM?

Response Returns: 10 V{voltage}

12.6 TRIG[n]:SEQ2:LEV:VOLT TRIG[n]:ACQ:LEV:VOLT

Description Set the trigger level for internally triggered voltage measurements.
A positive voltage trigger occurs when the voltage level changes from a value less than the lower hysteresis band limit to a value greater than the upper hysteresis band limit.

A negative voltage trigger occurs when the voltage level changes from a value greater than the upper hysteresis band limit to a value less than the lower hysteresis band limit.

Syntax

```
TRIGger<channel>:SEQuence2:LEVel:VOLTage <voltage>
TRIGger<channel>:ACQuire:LEVel:VOLTage <voltage>

<channel> := {1 | 2}
<voltage> := {0 to 15.1 v}
```

Query

```
TRIGger<channel>:SEQuence2:LEVel:VOLTage?
TRIGger<channel>:ACQuire:LEVel:VOLTage?
```

Example

```
TRIG:SEQ2:LEV:VOLT 10
TRIG:ACQ:LEV:VOLT 2
```

Response Returns: 10 V {voltage}

12.7 TRIG[n]:SEQ2:SLOP:CURR TRIG[n]:ACQ:SLOP:CURR

Description Set the slope of an internally triggered current measurement.

Syntax TRIGger<channel>:SEQuence2:SLOPe:CURRent <slope>
TRIGger<channel>:ACQuire:SLOPe:CURRent <slope>

<channel> := {1 | 2}
<slope> := {EITHer | POSitive | NEGative}

Slope	Description
POSitive	Trigger occurs on the rising edge.
NEGative	Trigger occurs on the falling edge.
EITHer	Trigger occurs on either edge.

Table 12.1 Slope Trigger

Query TRIGger<channel>:SEQuence2:SLOPe:CURRent?
TRIGger<channel>:ACQuire:SLOPe:CURRent?

Example TRIG:SEQ2:SLOP:CURR POS NEG TRIG:SEQ2:SLOP:CURR POS?

Response Returns: NEGative {slope}

12.8 TRIG[n]:SEQ2:SLOP:DVM TRIG[n]:ACQ:SLOP:DVM

Description Set the slope of an internally triggered DVM measurement.

Syntax TRIGger<channel>:SEQuence2:SLOPe:DVM <slope>
TRIGger<channel>:ACQuire:SLOPe:DVM <slope>

<channel> := {1 | 2}
<slope> := {EITHer | POSitive | NEGative}

Slope	Description
POSitive	Trigger occurs on the rising edge.
NEGative	Trigger occurs on the falling edge.
EITHer	Trigger occurs on either edge.

Table 12.2 Slope Trigger

Query TRIGger<channel>:SEQuence2:SLOPe:DVM? TRIGger<channel>:ACQuire:SLOPe:DVM?

Example TRIG:SEQ2:SLOP:DVM POS NEG TRIG:SEQ2:SLOP:DVM POS?

Response Returns: NEGative {slope}

12.9 TRIG[n]:SEQ2:SLOP:VOLT TRIG[n]:ACQ:SLOP:VOLT

Description Set the slope of an internally triggered voltage measurement.

Syntax TRIGger<channel>:SEQuence2:SLOPe:VOLTage <slope>
TRIGger<channel>:ACQuire:SLOPe:VOLTage <slope>

<channel> := {1 | 2}
<slope> := {EITHer | POSitive | NEGative}

Slope	Description
POSitive	Trigger occurs on the rising edge.
NEGative	Trigger occurs on the falling edge.
EITHer	Trigger occurs on either edge.

Table 12.3 Slope Trigger

Query TRIGger<channel>:SEQuence2:SLOPe:VOLTage?
TRIGger<channel>:ACQuire:SLOPe:VOLTage?

Example TRIG:SEQ2:SLOP:VOLT POS NEG
TRIG:SEQ2:SLOP:VOLT POS?

Response Returns: NEGative {slope}

12.10 TRIG[n]:SEQ2:SOUR TRIG[n]:ACQ:SOUR

Description Set the source for the measurement trigger.

Syntax TRIGger<channel>:SEQuence2:SOURce <source>
TRIGger<channel>:ACQuire:SOURce <source>

<channel> := {1 | 2}
<slope> := {INTernal | BUS | EXTernal}

Slope	Description
INTernal	Pressing the **Trig** button will trigger an event.
BUS	Commands sent through the selected remote interface will trigger an events.
EXTernal	External signals received through pins 1 and 6 of the System I/O Interface (DB9 terminal) will trigger events. A trigger will take place when a descending edge is registered

Table 12.4 Trigger Source

Query TRIGger<channel>:SEQuence2:SOURce?
TRIGger<channel>:ACQuire:SOURce?

Example TRIG:ACQ:SOUR BUS
TRIG:ACQ:SOUR?

Response Returns: BUS {source}

12.11 TRIG[n]:SEQ2:MODE TRIG[n]:ACQ:MODE

Description Set the internal voltage trigger mode.

Syntax TRIGger<channel>:SEQuence2:MODE <mode>
TRIGger<channel>:ACQuire:MODE <mode>

<channel> := {1 | 2}
<mode> := {AUTo | NORMal | SINGel}

Query TRIGger<channel>:SEQuence2:MODE?
TRIGger<channel>:ACQuire:MODE?

Example TRIG:SEQ2:MODE AUTo
TRIG:SEQ2:MODE?

Response Returns: AUTo {mode}

Trace Subsystem

The commands listed in the **Trace Subsystem** provide remote access for configuration of all buffer parameters.

13.1 TRACe[n]:CLEar

Description Clear the data buffer.

Syntax TRACe<channel>:CLEar

Example TRAC2:CLE

13.2 TRACe[n]:DATA?

Description This command queries all data in the data buffer. Data type is set by **TRACe[n]:STATistics**.

Query TRACe<channel>:DATA?

Example TRAC:DATA?

Response Returns:
0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000

13.3 TRACe[n]:POINts:ACTual?

Description Query the count of points in the buffer.

Query TRACe:POINts:ACTual?

Example TRAC:POIN:ACT?

Response Returns: {Data in buffer}

13.4 TRACe[n]:POINt

Description Set the data buffer depth.

Syntax TRACe<channel>:POINt <point>
<point> := {1 to 1024 }

Query TRACe<channel>:POINt?

Example TRAC2:POINt TRAC2:POINt?

Response Returns: 1024{buffer size}

13.5 TRACe:FEED

Description Select the measurements saved in the buffer. Voltage, currrent, or both measurements can be stored in the buffer. To select both measurement select the **TWO** measurement.

Syntax TRACe:FEED <measurement>
<measurement> := {VOLTage | CURRent | TWO}

Query TRACe:FEED

Example TRAC:FEED VOLT
TRAC:FEED?

Response Returns: VOLTage

13.6 TRACe[n]:FEED:CONTrol

Description Set the storage mode of the buffer.

Syntax TRACe<channel>:FEED:CONTrol <mode>
<channel> := {1 | 2}
<mode> := {ALWays | NEXT | NEVer}

Modes	Description
ALWays	The buffer queue will be set to FIFO (first in first out). The measurements will be stored and read as they occur. When the buffer is full the oldest measurement will be deleted in order to inclued the newest measurement.
NEXT	Write protection will be enabled. Once the buffer is full new measurements will not be written into the buffer.
NEVer	Write protection will be disabled. No measurements will be stored in the buffer.

Table 13.1 Buffer Modes

Query TRACe<channel>:FEED:CONTrol?

Example TRAC:FEED:CONT ALW
TRAC:FEED:CONT?

Response Returns: ALW {mode}

13.7 TRACe[n]:STATistics

Description Set the data type of the emasurements stored in the buffer.

Syntax TRACe<channel>:STATistics <type> <channel> := {1 | 2}
<type> := {MEAN | PEAK | MAX | MIN}

Query TRACe<channel>:STATistics?

Example TRAC2:STAT MAX
TRAC2:STAT?

Response Returns: MAX {type}

13.8 TRACe[n]:CLEar

Description Enable/disable auto clear of the buffer once it is full.

Syntax TRACe<channel>:CLEar <state>
<channel> := {1 | 2}
<type> := {1 | 0 or On | Off}

Query TRACe<channel>:CLEar?

Example TRAC2:CLE On
TRAC2:CLE?

Response Returns: 1 {state}

13.9 TRACe[n]:SAVE

Description Import data from the buffer to external USB device.

Syntax TRACe<channel>:SAVE

Example TRAC:SAVE

Display Subsystem

The commands listed in the **Display Subsystem** provide remote access for selection of channel, menus, and screen brightness.

14.1 DISPlay:CHANnel

Description Select the channel view displayed on the front panel.

Syntax DISPlay:CHANnel <view>
<view> := {1 | 2 | 3}

Views	Description
1	Display single view of channel 1.
2	Display single view of channel 2.
3	Display dual view of channels 1 and 2.

Table 14.1 Front Panel View

Query DISPlay:CHANnel?

Example DISP:CHAN 2
DISP:CHAN?

Response Returns: 2 {view}

14.2 DISPlay:SCREen

Description Navigate to the selected interface.

Syntax DISPlay:SCREen <interface>
<interface> := {MENU | HOME | GRAPh | BATTery}

Query DISPlay:SCREen?

Example DISP:SCRE HOME

Response Returns: HOME {interface}

14.3 DISPlay:BRIGhtness

Description Set the LCD brightness.

Syntax DISPlay:BRIGhtness <brightness>
<brightness> := {1 to 10}

Query DISPlay:BRIGhtness?

Example DISP:BRIG 10
DISP:BRIG?

Response Returns: 10 {brightness}

System Subsystem

The commands listed in the **System Subsytem** provide remote access for configuration of all system parameters.

15.1 SYSTem:POSetup

Description Select the power-on defaults.
With RST selected, the instrument powers up to the *RST default conditions.
With the SAV0 parameter selected, the instrument powers-on to the setup that is saved in the specified location using the *SAV command.

Syntax SYSTem:POSetup <state>
<state> := {RST | SAV0}

Query SYSTem:POSetup?

Example SYST:POS SAV0

Response Returns: SAV0 {SAV0}

15.2 SYSTem:VERSion?

Description Query the SCPI version to which the power supply complies to. THe value is in the form YYYY.V, where YYYY is the year and V is the revision number for that year.

Query SYSTem:VERSion?

Example SYST:VERS?

Response Returns: 1991.0{version}

15.3 SYSTem:ERRor?

Description Query and clear the first erro from the erro queue (**FIFO**).

Query SYSTem:ERRor?

Example SYST:ERR?

Response Returns: 0, No error

15.4 SYSTem:CLEar

Description Clear the Error Queue.

Syntax SYSTem:CLEar

Example SYST:CLE

15.5 SYSTem:LOCal

Description Set the instrument to local mode. In local mode the front panel keys are functional.

Syntax SYSTem:LOCal

Example SYST:LOC

15.6 SYSTem:REMote

Description Set the instrument to remote mode. Remote mode disables the front panel keys except **Print**. To enable the front panels keys enter local mode using the **SYST:LOC** command.

Syntax SYSTem:REMote

Example

15.7 SYSTem:RWLock

Description Set the instrument in remote mode. in this mode all front panels keys will be disabled, including the **Print** key. To enable the front panels keys enter local mode using the **SYST:LOC** command.

Syntax SYSTem:RWLock

Example SYST:RWL

15.8 SYSTem:DATE

Description Set the system date.

Syntax SYSTem:DATE <YY>,<MM>,<DD>

Query SYSTem:DATE?

Example SYST:DATE 21,5,4

Response Returns: 21,5,4 {YY,MM,DD}

15.9 SYSTem:TIME

Description Set the system time.

Syntax SYSTem:TIME <HH>,<MM>,<SS>

Query SYSTem:TIME?

Example SYST:DATE 21,5,4

Response Returns: 21,5,4 {HH,MM,SS}

15.10 SYSTem:COMMunicate:SELect

Description Set the communication interface.

Syntax SYSTem:COMMunicate:SELect <interface>
<interface> := {USB | LAN}

Query SYSTem:COMMunicate:SELect?

Example SYST:COMM:SEL USB
SYST:COMM:SEL?

Response Returns: USB {interface}

15.11 SYSTem:COMMunicate:PROTocol?

Description Query the system communication protocol.

Query SYSTem:COMMunicate:PROTocol?

Example SYST:COMM:PROT?

Response Returns: SCPI

15.12 SYSTem:COMMunicate:LAN:IP

Description Set the static IP address.

Syntax SYSTem:COMMunicate:LAN:IP <address>

Query SYSTem:COMMunicate:LAN:IP?

Example SYST:COMM:LAN:IP "192.168.1.10"
SYST:COMM:LAN:IP?

Response Returns: 192.168.1.10 {address}

15.13 SYSTem:COMMunicate:LAN:GATEway

Description Set the network gateway.

Syntax SYSTem:COMMunicate:LAN:GATEway <gateway>

Query SYSTem:COMMunicate:LAN:GATEway?

Example SYSTem:COMMunicate:LAN:GATEway "192.168.0.1"

Response Returns: 192.168.0.1 {gateway}

15.14 SYSTem:COMMunicate:LAN:MASK

Description Set the network's subnet mask.

Syntax SYSTem:COMMunicate:LAN:MASK <mask>

Query SYSTem:COMMunicate:LAN:MASK?

Example SYSTem:COMMunicate:LAN:MASK "255.255.255.0"

Response Returns: 255.255.255.0 {mask}

15.15 SYSTem:COMMunicate:LAN:SOCKetport

Description Set the socket port of the system network.

Syntax SYSTem:COMMunicate:LAN:SOCKetport <port>

Query SYSTem:COMMunicate:LAN:SOCKetport>

Example SYSTem:COMMunicate:LAN:SOCKetport 5025

Response Returns: 5025 {port}

15.16 SYSTem:SAVe

Description Save all the set system parametes. If not saved at power off all changes made in the session will be lost.

Syntax SYSTem:SAVe

Example SYST:SAV

15.17 CALibrate:INITialize

Description Inititalize the calibration parameters.

Syntax CALibrate:INITialize

Example CAL:INIT

Version: September 1, 2022