Programming Manual

MDL4U Series
Modular Programmable DC Electronic Load

bkprecision.com

BK PRECISION

Contents

1 Introduction to Programming 6
1.1 GPIB Capabilities of the Electronic Load 6
1.2 RS232 Capabilities of the Electroni Load 7
1.3 USB Interface 9
1.4 Programming the Status Register 9
1.4.1 Condition Registers 13
1.4.2 Event registers 13
1.4.3 Enable Registers 13
1.4.4 Output Queue 13
1.4.5 Error Queue 14
1.4.6 Status Byte and Service Request (SRQ) 14
1.4.7 Status Byte Register 14
1.4.8 Service Request Enable Register 14
1.5 Serial Poll and SRQ 15
1.6 Trigger Model (GPIB Operation) 15
1.6.1 Idle and Iniate 15
1.6.2 Trigger Model Operation 16

2 Introduction to SCPI 17
2.1 Types of SCPI Commands 17
2.1.1 Multiple commands in a Message 18
2.1.2 Moving Among Sybsystems 18
2.1.3 Including Common Commands 18
2.1.4 Case Sensitivity 19
2.1.5 Long-form and Short-form Versions 19
2.1.6 Using Queries 19
2.2 Types of SCPI Messages 20
2.2.1 The Message Unit 20
2.2.2 Headers 20
2.2.3 Query Indicator 20
2.2.4 Message Unit Separator 20
2.2.5 Root Specifier 21
2.2.6 Command Execution Rules 21
2.3 SCPI Data 21
2.4 Response Data Types 21
2.5 SCPI Command Completion 23
2.6 Language Dictionary Information 24
2.7 Subsystem Commands 24

3 IEEE-488 Command Refernce 25
3.1 *CLS 26
3.2 *ESE <NRf> 26
3.3 *ESR? 26
3.4 *IDN? 27
3.5 *RDT? 27
3.6 *OPC 28
3.7 *RCL 28
3.8 *RST 29
3.9 *SAV 30
3.10 *SRE 30
3.11 *STB? 30
3.12 *TRG 31
3.13 *TST? 31

3.14 *WAI 31

4 Channel Subsystem 32
4.1 CHANnel 32
4.2 CHANnel:ID? 32

5 Trigger Subsystem 33
5.1 TRIGer:SOURce 33
5.2 TRIGger:TIMer 33

6 System Subsystem 34
6.1 SYSTem:PRESet 34
6.2 SYSTem:POSetup 34
6.3 SYSTem:VERSion? 34
6.4 SYSTem:ERRor? 34
6.5 SYSTem:CLEar 35
6.6 SYSTem:LOCal 35
6.7 SYSTem:REMote 35
6.8 SYSTem:RWLock 35

7 Status Subsystem 37
7.1 STATus:CHANnel? 37
7.2 STATus:CHANnel:CONDition? 37
7.3 STATus:CHANnel:ENABle 37
7.4 STATus:CSUM? 38
7.5 STATus:CSUMmary:ENABle 38
7.6 STATus:OPERation? 38
7.7 STATus:OPERation:CONDition? 39
7.8 STATus:OPERation:ENABle 39
7.9 STATus:QUEStionable? 39
7.10 STATus:QUEStionable:CONDition? 40
7.11 STATus:QUEStionable:ENABle 40
7.12 STATus:PRESet 40

8 Trace Subsystem 41
8.1 TRACe:CLEar 41
8.2 TRACe:FREE? 41
8.3 TRACe:POINts 41
8.4 TRACe:FEED 42
8.5 TRACe:FEED:CONTrol 42
8.6 TRACe:DATA? 42
8.7 TRACe:FILTer 43
8.8 TRACe:DELay 43
8.9 TRACe:TIMer 43

9 Source Subsystem 44
9.1 [SOURce:]INPut:ALL 44
9.2 [SOURce:]INPut 44
9.3 [SOURce:]INPut:SYNCon 44
9.4 [SOURce:]INPut:SHORt 45
9.5 [SOURce:]REMote:SENSe 45
9.6 [SOURce:]FUNCtion 45
9.7 [SOURce:]FUNCtion:MODE 46
9.8 [SOURce:]TRANsient 46
9.9 [SOURce:]PROTection:CLEar 46
9.10 [SOURce:]INPut:TIMer 47
9.11 [SOURce:]INPut:TIMer:DELay 47
9.12 [SOURce:]CURRent 47
9.13 [SOURce:]CURRent:RANGe 48

9.14 [SOURce:]CURRent:SLEW 48
9.15 [SOURce:]CURRent:SLEW:POSitive 49
9.16 [SOURce:]CURRent:SLEW:NEGative 49
9.17 [SOURce:]CURRent:PROTection:STATe 49
9.18 [SOURce:]CURRent:PROTection 50
9.19 [SOURce:]CURRent:PROTection:DELay 50
9.20 [SOURce:]CURRent:TRANsient:MODE 51
9.21 [SOURce:]CURRent:TRANsient:ALEVel [SOURce:]CURRent:TRANsient:BLEVel 51
9.22 [SOURce:]CURRent:TRANsient:AWIDth [SOURce:]CURRent:TRANsient:BWIDth 51
9.23 [SOURce:]CURRent:HIGH [SOURce:]CURRent:LOW 52
9.24 [SOURce:]VOLTage 52
9.25 [SOURce:]VOLTage:RANGe 52
9.26 [SOURce:]VOLTage:RANGe:AUTO 53
9.27 [SOURce:]VOLTage:ON 53
9.28 [SOURce:]VOLTage:LATCh 53
9.29 [SOURce:]VOLTage:HIGH [SOURce:]VOLTage:LOW 54
9.30 [SOURce:]VOLTage:TRANsient:MODE 54
9.31 [SOURce:]VOLTage:TRANsient:ALEVel [SOURce:]VOLTage:TRANsient:BLEVel 55
9.32 [SOURce:]VOLTage:TRANsient:AWIDth [SOURce:]VOLTage:TRANsient:BWIDth 55
9.33 [SOURce:]RESistance 55
9.34 [SOURce:]RESistance: RANGe 56
9.35 [SOURce:]RESistance:HIGH [SOURce:]RESistance:LOW 56
9.36 [SOURce:]RESistance:TRANsient:MODE 56
9.37 [SOURce:]RESistance:TRANsient:ALEVel [SOURce:]RESistance:TRANsient:BLEVel 57
9.38 [SOURce:]RESistance:TRANsient:AWIDth [SOURce:]RESistance:TRANsient:BWIDth 57
9.39 [SOURce:]POWer 58
9.40 [SOURce:]POWer:RANGe 58
9.41 [SOURce:]POWer:HIGH [SOURce:]POWer:LOW 58
9.42 [SOURce:]POWer:PROTection 59
9.43 [SOURce:]POWer:PROTection:DELay 59
9.44 [SOURce:]POWer:CONF 60
9.45 [SOURce:]IMPedance 60
9.46 [SOURce:]IMPedance:RANGe 60
9.47 [SOURce:]IMPedance:RESistance 61
9.48 [SOURce:]IMPedance:INDuction 61
9.49 [SOURce:]IMPedance:CAPacitance 61
9.50 [SOURce:]IMPedance:HIGH 62

10 List Subsystem 63
10.1 [SOURce:]LIST:RANGe 63
10.2 [SOURce:]LIST:COUNt 63
10.3 [SOURce:]LIST:STEP 63
10.4 [SOURce:]LIST:LEVel? 64
10.5 [SOURce:]LIST:SLEW 64
10.6 [SOURce:]LIST:WIDth 64
10.7 L 65
10.8 [SOURce:]LIST:RCL 65

11 Measurement Subsystem 66
11.1 :FETCh:VOLTage? :FETCh:CURRent? :FETCh:POWer[:DC]? 66
11.2 :MEASure:VOLTage[:DC]? :MEASure:CURRent[:DC]? 67
11.3 :MEASure:ALLVoltage? :MEASure:ALLCurrent? 67
11.4 :FETch :ALLVoltage? :FETch:ALLCurrent? 67

12 Sense Subsystem 68
12.1 SENSe:AVERage:COUNt 68
12.2 SENSe:VOLTage[:DC]:RANGe:AUTO 68

13 Programming Subsystem 69
13.1 :PROGram:ACTive:CHANnel 69
13.2 :PROGram:ACTive:SEQuence 69
13.3 :PROGram:SEQuence:SHORt 70
13.4 :PROGram:SEQuence:ONTime 70
13.5 :PROGram:SEQuence:OFFTime 70
13.6 :PROGram:SEQuence:PFDTime 71
13.7 :PROGram:SEQuence:PAUSe 71
13.8 :PROGram:STOP:CONDition 72
13.9 :PROGram:CHAin 72
13.10 :PROGram:SAVe 72
13.11 :PROGram:RCL 72

14 Other Commands 73
14.1 FAC:MAC? 73

15 SCPI Command Tree 74

16 Programming Examples 81
16.1 Introduction 81
16.1.1 Power-on Initialization 81
16.1.2 Enabling the Input 81
16.1.3 Input Voltage 81
16.1.4 Input Current 82
16.1.5 Overcurrent Protection 82
16.1.6 Generating Triggers 82
16.1.7 Programming Transients 82
16.1.8 Continuous Transients 83
16.1.9 Pulse Transients 83
16.1.10 Toggled Transients 84
16.2 Programming Lists 84
16.2.1 4-Step Current Change List Example 84

17 Error Messages 86
17.1 Error Number List 86
17.2 Command Errors 86
17.3 Execution Errors 87
17.4 System Errors 87
17.5 Self-test Errors 88
17.6 Device-Dependent Errors 89

Introduction to Programming

This guide contains programming information for the B&K Precision MDL4U Series DC Electronic Load. Models in this series include the MDL4U001, MDL4U002, MDL4U200, MDL4U252, MDL4U305, MDL4U400, MDL4U505, and MDL4U600. Unless otherwise noted, this document will refer to all of these instruments as "electronic load".

1.1 GPIB Capabilities of the Electronic Load

All electronic load functions except for setting the communication parameters are programmable over the GPIB. The IEEE 488.2 capabilities of the electronic load are described in the table below.

GPIB Capabilities	Response	Interface Function
Talker/Listener	All electronic load functions except for setting the communication parameters are programmable over the GPIB. The electronic load can send and receive messages over the GPIB. Status information is sent using a serial poll.	AH1, SH1, T6, L4
Service Request	The electronic load sets the SRQ line true if there is an enabled service request condition.	SR1
Remote/Local	In local mode, the electronic load is controlled from the front panel but will also execute commands sent over the GPIB. The electronic load powers up in local mode and remains in local mode until it receives a command over the GPIB. Once the electronic load is in remote mode, the front panel REM annunciator turns on, all front panel keys (except Shift + Local) are disabled, and the display is in normal metering mode. Pressing Shift + Local on the front panel returns the electronic load to local mode. This can be disabled using local lockout so that only the controller or the power switch can return the electronic load to local mode.	RL1
Device Trigger	The electronic load will respond to the device trigger function.	DT1
Group Execute Trigger	The electronic load will respond to the group execute trigger function.	GET
Device Clear	The electronic load responds to the Device Clear (DCL) and Selected Device Clear (SDC) interface commands. They cause the electronic load to clear any activity that would prevent it from receiving and executing a new command (including *WAI and *OPC?). DCL and SDC do not change any programmed settings.	DCL,SDC

Table 1.1 GPIB Capabilities of the Electronic Load

GPIB Addres

The electronic load operates from a GPIB address that is set from the front panel. To set the GPIB address, press Shift + 7 (System menu) on the front panel and enter the address using the Entry keys. The address can be set from 0 to 30. The GPIB address is stored in non-volatile memory.

1.2 RS232 Capabilities of the Electroni Load

Use a cable with two serial interfaces (DB9) to connect the electronic load and PC. It can be activated by selecting <RS-232> in <Communication> of the System menu (Shift + 7 on the front panel). NOTE: There are two serial interfaces on the rear panel of the MDL4U001: the left 9-pin COM interface is the RS-232 communication interface and the right 9-pin COM serial port connection is not for use. All SCPI commands are available through RS-232 programming. The EIA RS-232 standard defines the interconnections between data terminal equipment (DTE) and data communications equipment (DCE). The electronic load is designed to be a DTE and can be connected to another DTE such as a PC COM port through a null modem cable.

Note:

The RS232 settings in your program must match the settings specified in the front panel System menu. Press Shift + 7 on the front panel to enter the System menu if you need to change the settings. You can break data transmissions by sending a ^C or ^X character string to the electronic load. This clears any pending operation and discards any pending output.

RS232 Data Format

The RS-232 data is a 10-bit word with one start bit and one stop bit.

Parity = None	Start Bit	8 Data Bits	Stop Bits

The number of start and stop bits are not programmable. However, the following parameters are selectable in the System menu using the front panel shift + 7 key.

Baud Rate

The System menu lets you select one of the following baud rates, which are stored in non-volatile memory: 4800, 9600, 19200, 38400, 57600, or 115200.

Parity

None - eight data bits without parity

Even - seven data bits with even parity

Odd - seven data bits with odd parity

RS232 Flow Control

The RS232 interface supports the following flow control options. For each case, the electronic load will send a maximum of five characters after hold-off is asserted by the controller. The electronic load is capable of receiving as many as fifteen additional characters after it asserts hold-off.

- **CTS/RTS**: The electronic load asserts its Request to Send (RTS) line to signal hold-off when its input buffer is almost full, and it interprets its Clear to Send (CTS) line as a hold-off signal from the controller.
- **XON/XOFF**: When the input queue of the electronic load becomes more than ¾ full, the instrument issues an X-OFF command. The control program should respond to this and stop sending characters until the electronic load issues the X-ON, which it will do once its input buffer has dropped below half-full. The electronic load recognizes X_ON

and X_OFF sent from the controller. An X-OFF will cause the electronic load to stop outputting characters until it sees an X-ON.

- **NONE**: No flow control.

Flow control options are stored in non-volatile memory.

RS232 Connections

The RS-232 serial port can be connected to the serial port of a controller (i.e., personal computer) using a straight-through RS-232 cable terminated with DB-9 connectors. Do not use a null modem cable. Figure 33 shows the pinout for the connector.

If your computer uses a DB-25 connector for the RS-232 interface, you will need a cable or adapter with a DB-25 connector on one end and a DB-9 connector on the other. It must be a straightthrough (not null modem) a cable.

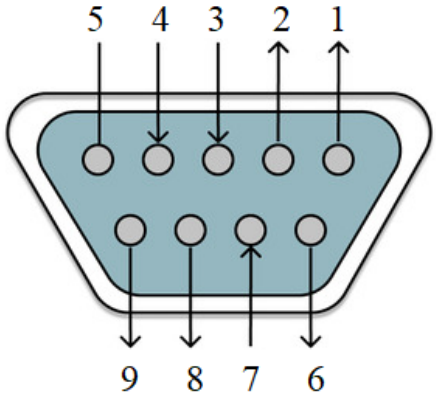

Figure 1.1 DB9 Pinout

Pin Number	Signal	Function
1	NC	No Connection
2	TXD	Transmit Data
3	RXD	Receive Data
4	NC	No Connection
5	GND	Ground
6	NC	No Connection
7	CTS	Clear to Send
8	RTS	Ready to Send
9	NC	No Connection

Table 1.2 DB9 Pinout

RS232 Troubleshooting

If you are having trouble communicating over the RS-232 interface, check the following:

- The computer and the electronic load must be configured for the same baud rate, parity, number of data bits, and flow control options. Note that the electronic load is configured for 1 start bit and 1 stop bit (these values are fixed).
- The correct interface cables or adapters must be used, as described under the RS-232 connector. Note that even if the cable has the proper connectors for your system, the internal wiring may be incorrect.
- The interface cable must be connected to the correct serial port on your computer (COM1, COM2, etc.) and the correct 9-pin serial port on the mainframe.

Communication Settings

Before communicating, please make sure that the following parameters of the electronic load match that of the PC. You can enter the System menu (Shift + 7) to make any changes.
Baud rate : 9600 (5800, 9600, 19200, 38400, 57600, 115200)

Data bit : 8

Stop bit : 1

Parity : None (None, Even, Odd)

Local address : 0 (0 through 31, default setting is 0)

Note:

When communicating with a PC, you can only use one communication interface at a time.

1.3 USB Interface

Use Type A to Type B USB cables to connect the electronic load and the PC. All electronic load functions are programmable over the USB. Press shift + 7 on the front panel to enter the **System** menu. Select **Communication** and choose **USBTMC-USB488**.

The USB488 interface capabilities of the electronic load are described below:

- The interface is IEEE 488.2 standard USB488 interface.
- The interface accepts REN_CONTROL, GO_TO_LOCAL, and LOCAL_LOCKOUT requests.
- The interface accepts MsgID = TRIGGER USBTMC command message and forwards TRIGGER requests to the function layer.

The USB488 device capabilities of the electronic load are described below:

- The device understands all mandatory SCPI commands.
- The device is SR1 capable.
- The device is RL1 capable.
- The device is DT1 capable.

1.4 Programming the Status Register

You can use status register programming to determine the operating condition of the electronic load at any time. For example, you may program the electronic load to generate an interrupt (assert SRQ) when an event such as a current protection occurs. When the interrupt occurs, your program can then act on the event in the appropriate fashion.

The following table defines the status bits. Figure shows the status register structure of the electronic load. The Standard Event, Status Byte, and Service Request Enable registers and the Output Queue perform standard GPIB functions as defined in the IEEE 488.2 Standard Digital Interface for Programmable Instrumentation. The Operation Status and Questionable Status registers implement functions that are specific to the electronic load.

Bit Configurations of Status Registers

Bit	Signal	Definition
0	CAL	*Calibrating*. The electronic load is computing new calibration constants.
5	TRG	*Waiting*. The electronic load is waiting for a trigger.

Table 1.3 Bit Configurations of Status Registers

Channel Status Group

Bit	Signal	Definition
0	VF	*Voltage Fault*. Either an overvoltage or a reverse voltage has occurred. This bit reflects the active state of the FLT pin on the back of the unit. The bit remains set until the condition is removed and INP:PROT:CLE is programmed.
1	OC	*Overcurrent*. An overcurrent condition has occurred. This occurs if the current exceeds 102% of the rated current or if it exceeds the user-programmed current protection level. Removing the overcurrent condition clears the bit. If the condition persists beyond the user programmable delay time, PS bit is also set and the input is turned off. Both bits remain set until the condition is removed and INP:PROT:CLE is programmed.
2	RS	*Remote Sense*. When the remote sense is connected, this bit is set.
3	OP	*Overpower*. An overpower condition has occurred. This occurs if the unit exceeds the max power or it exceeds the user-programmed power protection level. Removing the overpower condition clears the bit. If the condition persists beyond the user programmable delay time, PS bit is also set and the input is turned off. Both bits remain set until the condition is removed and INP:PROT:CLE is programmed.
4	OT	*Overtemperature*. An overtemperature condition has occurred. Both this bit and PS bit are set and the input is turned off. Both bits remain set until the unit is cooled down and INP:PROT:CLE is programmed.
7	RUN	*List run or stop status*. When list is running, this bit is set.
8	EPU	*Extended Power Unavailable*. This bit is not used.
9	RRV	*Remote Reverse Voltage*. A reverse voltage condition has occurred on the sense terminals. Both this bit and VF bit are set. Removing the reverse voltage clears this bit but does not clear VF bit. VF bit remains set until INP:PROT:CLE is programmed.
10	UNR	*Unregulated*. The input is unregulated. When the input is regulated this bit is cleared.
11	LRV	*Local Reverse Voltage*. A reverse voltage condition has occurred on the input terminals. Both this bit and VF bit are set. Removing the reverse voltage clears this bit but does not clear PS bit. PS bit remains set until INP:PROT:CLE is programmed.
12	OV	*Overvoltage*. An overvoltage condition has occurred. Both this bit and VF bit 0 are set and the electronic loads are turned off. Both bits remain set until the condition is removed and INP:PROT:CLE is programmed.
13	PS	*Protection Shutdown*. The protection shutdown circuit has tripped because of an overcurrent, overpower, or overtemperature condition. The bit remains set until INP:PROT:CLE is programmed.
14	VON	*Voltage of sink current on*. When the voltage of input exceeds the user-programmed Von level, this bit is set.
15	TBF	*Trace Buffer Full*.

Table 1.4 Channel Status Group

Questionable Status Summary

Bit	Signal	Definition
0	OPC	*Operation Complete*. The load has completed all pending operations. *OPC must be programmed for this bit to be set when pending operations are complete.
2	QYE	*Query Error*. The output queue was read with no data present or the data was lost. Errors in the range of 499 through 400 can set this bit.
3	DDE	*Device-Dependent Error*. Memory was lost or self-test failed. Errors in the range of 399 through 300 can set this bit.
4	EXE	*Execution Error*. A command parameter was outside its legal range, inconsistent with the load's operation, or prevented from executing because of an operating condition. Errors in the range of 299 through 200 can set this bit.
5	CME	*Command Error*. A syntax or semantic error has occurred or the load received a <get> within a program message. Errors in the range of 199 through 100 can set this bit.
7	PON	*Power-On*. The unit has been turned off and then on since this bit was last read.

Table 1.5 Questionable Status Summary

Status Byte and Service Request Enable Registers

Bit	Signal	Definition
0	CSUM	*Channel Summary*. Indicates if an enabled channel event has occurred.
2	EAV	*Error Available Summary*. Indicates if the Error Queue contains data.
3	QUES	*Questionable Status Summary*. Indicates if an enabled questionable event has occurred.
4	MAV	*Message Available Summary*. Indicates if the Output Queue contains data.
5	ESB	*Event Status Summary*. Indicates if an enabled standard event has occurred.
6	RQS/MSS	*Request Service*. During a serial poll, RQS is returned and cleared. Master Status Summary. For an *STB? query, MSS is returned without being cleared.
7	OPER	*Operation Status Summary*. Indicates if an operation event has occurred.

Table 1.6 Status Byte and Service Request Enable Registers

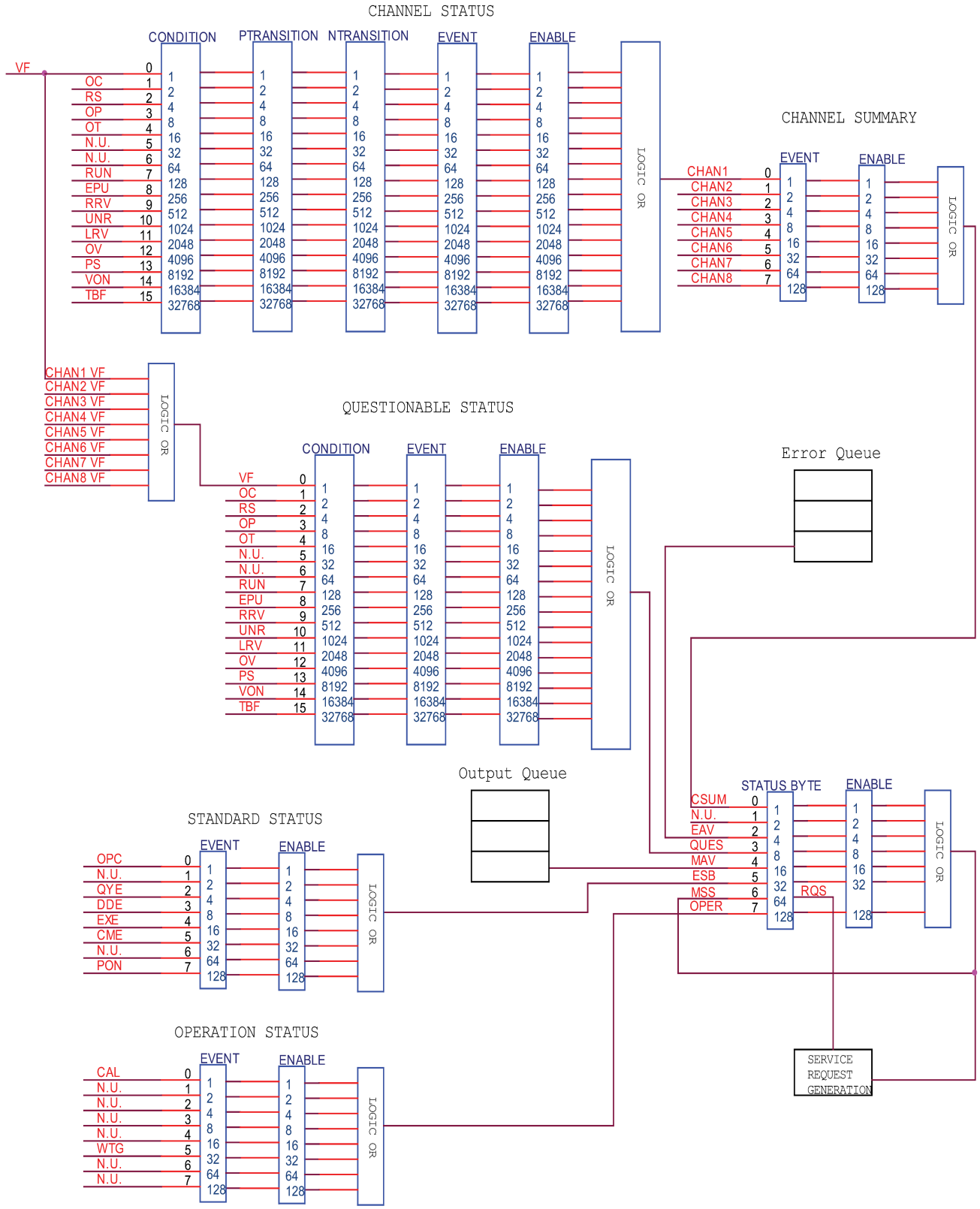

Figure 1.2 Load Status Register Structure

1.4.1 Condition Registers

All status register sets have a condition register. A condition register is a real-time, read-only register that constantly updates to reflect the current operating conditions of the instrument.

Use the :CONDition? query commands in the STATus Subsystem to read the condition registers. See Chapter 3 for more information.

1.4.2 Event registers

Each status register set has an event register. An event register is a latched, read-only register whose bits are set by the corresponding condition register. Once a bit in an event register is set, it remains set (latched) until the register is cleared by a specific clearing operation. The bits of an event register are logically ANDed with the bits of the corresponding enable register and applied to an OR gate. The output of the OR gate is applied to the Status Byte Register.

Use the *ESR? Common Command to read the Standard Event Register. All other event registers are read using the :EVENt? query commands in the STATus Subsystem. See Chapter 3 for more information.

An event register is cleared when it is read. The following operations clear all event registers:

- Cycling power
- Sending *CLS

1.4.3 Enable Registers

Each status register set has an enable register. An enable register is programmed by you and serves as a mask for the corresponding event register. An event bit is masked when the corresponding bit in the enable register is cleared (0). When masked, a set bit in an event register cannot set a bit in the Status Byte Register (1 AND $0 = 0$). To use the Status Byte Register to detect events (i.e., serial poll), you must unmask the events by setting (1) the appropriate bits of the enable registers. To program and query the Standard Event Status Register, use the *ESE and *ESE? Common Commands respectively. All other enable registers are programmed and queried using the :ENABle and :ENABLe? commands in the STATus Subsystem. See Chapter 3 for more information.

An enable register is not cleared when it is read. The following operations affect the enable registers:

- Cycling power clears all enable registers
- :STATus:PREset clears the following enable registers:
  - Operation Event Enable Register
  - Questionable Event Enable Register
  - Channel Summary Event Enable Register
- *ESE 0 clears the Standard Event Status Enable Register.

1.4.4 Output Queue

The output queue holds data that pertains to the normal operation of the instrument. For example, when a query command is sent, the response message is placed on the output queue.

When data is placed in the output queue, the Message Available (MAV) bit in the Status Byte Register gets set. A data message is cleared from the output queue when it is read. The output queue is considered cleared when it is empty. An empty output queue clears the MAV bit in the Status Byte Register.

1.4.5 Error Queue

The error queue holds error and status messages. When an error or status event occurs, a message that defines the error/status is placed in the error queue. This queue will hold up to 10 messages. When a message is placed in the error queue, the Error Available (EAV) bit in the Status Byte Register is set. An error message is cleared from the Error/Status queue when it is read. The error queue is considered cleared when it is empty. An empty error queue clears the EAV bit in the Status Byte Register. Read an error message from the error queue by sending the following SCPI query command:

:SYSTem:ERRor?

1.4.6 Status Byte and Service Request (SRQ)

Service request is controlled by two 8-bit registers: the Status Byte Register and the Service Request Enable Register.

1.4.7 Status Byte Register

The summary messages from the status registers and queues are used to set or clear the appropriate bits (B0, B2, B3, B4, B5, and B7) of the Status Byte Register. These bits do not latch, and their states (0 or 1) are solely dependent on the summary messages (0 or 1). For example, if the Standard Event Status Register is read, its register will clear. As a result, its summary message will reset to 0, which in turn will clear the ESB bit in the Status Byte Register.

Bit B6 in the Status Byte Register is either:

- The Master Summary Status (MSS) bit, sent in response to the *STB? Command, indicates the status of any set bits with corresponding enable bits set.
- The Request for Service (RQS) bit, sent in response to a serial poll, indicates which device was requesting service by pulling on the SRQ line.

For a description of the other bits in the Status Byte Register, see "Common commands, *STB?" The IEEE-488.2 standard uses the following common query command to read the Status Byte Register:

*STB?

When reading the Status Byte Register using the *STB? command, bit B6 is called the MSS bit. None of the bits in the Status Byte Register are cleared when using the *STB? command to read it.

The IEEE-488.1 standard has a serial poll sequence that also reads the Status Byte Register and is better suited to detect a service request (SRQ). When using the serial poll, bit B6 is called the RQS bit. Serial polling causes bit B6 (RQS) to reset. Serial polling is discussed in more detail later in this section entitled "Serial Poll and SRQ." Any of the following operations clear all bits of the Status Byte Register:

- Cycling power
- Sending the *CLS common command

Note:

The MAV bit may or may not be cleared.

1.4.8 Service Request Enable Register

This register is programmed by you and serves as a mask for the Status Summary Message bits (B0, B2, B3, B4, B5, and B7) of the Status Byte Register. When masked, a set summary bit in the Status Byte Register cannot set bit B6

(MSS/RQS) of the Status Byte Register. Conversely, when unmasked, a set summary bit in the Status Byte Register sets bit B6.

A Status Summary Message bit in the Status Byte Register is masked when the corresponding bit in the Service Request Enable Register is cleared (0). When the masked summary bit in the Status Byte Register sets, it is ANDed with the corresponding cleared bit in the Service Request Enable Register. The logic "1" output of the AND gate is applied to the input of the OR gate and, thus, sets the MSS/RQS bit in the Status Byte Register. The individual bits of the Service Request Enable Register can be set or cleared by using the following common command:

*SRE <NRf>

To read the Service Request Enable Register, use the *SRE? query command. The Service Request Enable Register clears when power is cycled or a parameter (n) value of zero is sent with the *SRE command (*SRE 0).

1.5 Serial Poll and SRQ

Any enabled event summary bit that goes from 0 to 1 will set RQS and generate a service request (SRQ). In your test program, you can periodically read the Status Byte Register to check if a service request (SRQ) has occurred and what caused it. If an SRQ occurs, the program can, for example, branch to an appropriate subroutine that will service the request. Typically, service requests (SRQs) are managed by the serial poll sequence of the electronic load. If an SRQ does not occur, bit B6 (RQS) of the Status Byte Register will remain cleared and the program will simply proceed normally after the serial poll is performed. If an SRQ does occur, bit B6 of the Status Byte Register will set and the program can branch to a service subroutine when the SRQ is detected by the serial poll. The serial poll automatically resets RQS of the Status Byte Register. This allows subsequent serial polls to monitor bit B6 for an SRQ occurrence generated by other event types. After a serial poll, the same event can cause another SRQ, even if the event register that caused the first SRQ has not been cleared.

A serial poll clears RQS but does not clear MSS. The MSS bit stays set until all Status Byte event summary bits are cleared.

1.6 Trigger Model (GPIB Operation)

This section describes how the electronic load operates over the GPIB bus. The flowchart in figure below summarizes operation over the bus and is called the trigger model. It is called the trigger model because operation is controlled by SCPI commands from the Trigger subsystem. Key SCPI commands are included in the trigger model.

1.6.1 Idle and Iniate

The instrument is considered to be in the idle state whenever it is not operating. While in the idle state, the instrument cannot perform any measure or scan functions. Two commands can be sent over the bus to remove the instrument from the idle state:

- :INITiate
- :INITiate:CONTinuous ON

With continuous initiation enabled (:INITiate:CONTinuous ON), the instrument will not remain in the idle state after all programmed operations are completed. However, the instrument can be set to idle state at any time by sending any of these commands:

- *RST
- ABORt
- *RCL

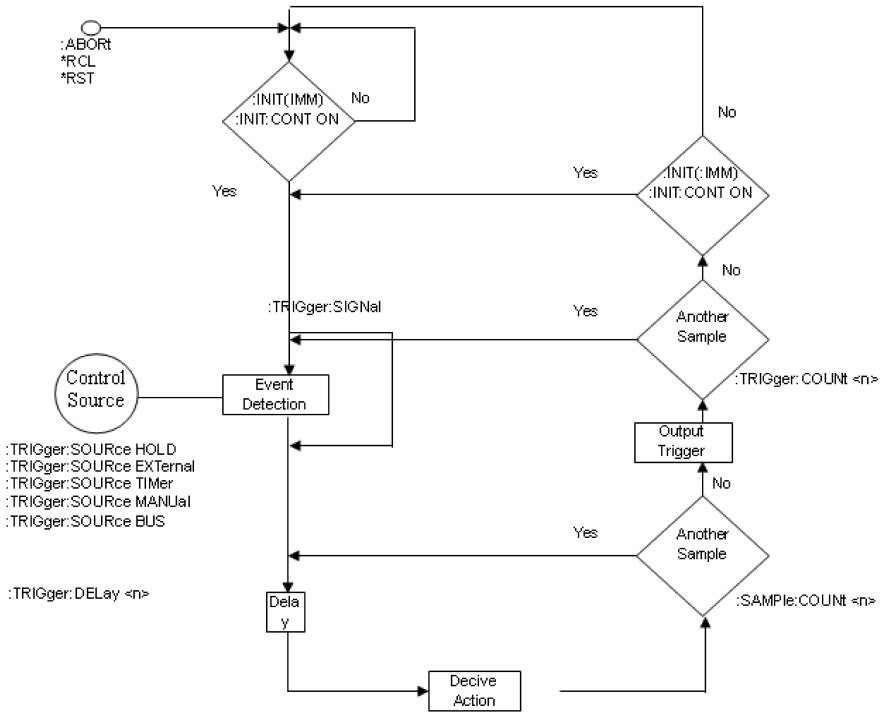

Figure 1.3 Trigger Model

1.6.2 Trigger Model Operation

Once the instrument is taken out of idle, operation proceeds through the trigger model down to the device action.

Control Source

As shown in Figure **1.3**, a control source is used to hold up operation until the programmed event occurs. The control source options are explained as follows:

- **HOLD** Only the TRIG:IMM command will generate a trigger in HOLD mode. All other trigger commands are ignored.
- **MANual** Event detection is satisfied by pressing the TRIG key.
- **TIMer** Generates triggers that are in synchronization with the electronic load's internal oscillator as the trigger source. The internal oscillator begins running as soon as this command is executed. Use TRIG:TIM to program the oscillator period.
- **EXTernal** Event detection is satisfied when an input trigger via the TRIGGER LINK connector is received by the electronic load.
- **BUS** Event detection is satisfied when a bus trigger (GET or *TRG) is received by the electronic load.
- **Delay** A programmable delay is available after the event detection. The delay can be manually set from 0 to 999999.999 seconds.

Introduction to SCPI

SCPI (Standard Commands for Programmable Instruments) is a programming language for controlling instrument functions over GPIB, RS-232, USB, and Ethernet interface. SCPI is layered on top of the hardware portion of IEEE 488.2. The same SCPI commands and parameters control the same functions in different classes of instruments.

Conventions Used in This Guide

Angle brackets	< >	Items within angle brackets are parameter abbreviations. For example, <NR1> indicates a specific form of numerical data.
Vertical bar	|	Vertical bars separate alternative parameters. For example, NORM |TEXT indicates that either "NORM" or "TEXT" can be used as a parameter.
Square Brackets	[ ]	Items within square brackets are optional. The representation [SOURce:] VOLTage means that SOURce: may be omitted.
Braces	{ }	Braces indicate parameters that may be repeated zero or more times. It is used especially for showing arrays. The notation <A>{<,B>} shows that parameter "A" must be entered, while parameter "B" may be omitted or may be entered one or more times.

2.1 Types of SCPI Commands

SCPI has two types of commands, common and subsystem.

Common

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

Subsystem

: Subsystem commands perform specific electronic load 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.

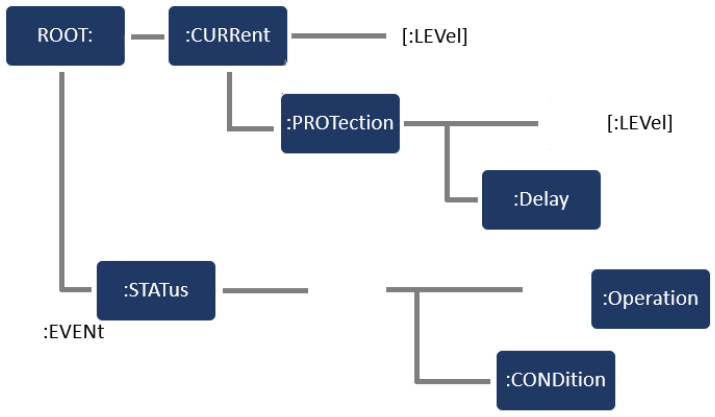

Figure 2.1 Partial Command Tree

2.1.1 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.
- There is an implied header path that affects how commands are interpreted by the electronic load.

The header path can be thought of as a string that gets inserted before each command within a message. For the first command in a message, the header path is a null string. For each subsequent command the header path is defined as the characters that make up the headers of the previous command in the message up to and including the last colon separator. An example of a message with two commands is:

```
CURR:LEV 3;PROT:STAT OFF
```

which shows the use of the semicolon separating the two commands, and also illustrates the header path concept. Note that with the second command, the leading header "CURR" was omitted because after the "CURR:LEV 3" command, the header path became defined as "CURR" and thus the instrument interpreted the second command as:

```
CURR:PROT:STAT OFF
```

In fact, it would have been syntactically incorrect to include the "CURR" explicitly in the second command, since the result after combining it with the header path would be:

```
CURR:CURR:PROT:STAT OFF
```

which is incorrect.

2.1.2 Moving Among Sybsystems

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
```

Observe 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.

2.1.3 Including Common Commands

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

```
VOLTage 17.5;*TRG
OUTPut OFF;*RCL 2;OUTPut ON
```

2.1.4 Case Sensitivity

Common commands and SCPI commands are not case sensitive. You can use upper or lower case and any case combination.

Example : *RST = *rst
:DATA? = :data?
:SYSTem:PRESet = :system:preset

2.1.5 Long-form and Short-form Versions

A SCPI command word can be sent in its long-form or short-form version. The command subsystem tables in Chapter 3 provide the long-form version. However, the short-form version is indicated by upper case characters.

Example :
:SYSTem:PRESet (long-form)
:SYST:PRES (short form)
:SYSTem:PRES (long-form and short-form combination)

Note:

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

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

2.1.6 Using Queries

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 electronic load. Otherwise a Query Interrupted error will occur and the unreturned data will be lost.

2.2 Types of SCPI Messages

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

1. A program message consists of one or more properly formatted SCPI commands sent from the controller to the electronic load. The message, which may be sent at any time, requests the electronic load to perform some action.

2. A response message consists of data in a specific SCPI format sent from the electronic load to the controller. The electronic load sends the message only when commanded by a program message called a "query."

The figure **2.2** illustrates SCPI message structure:

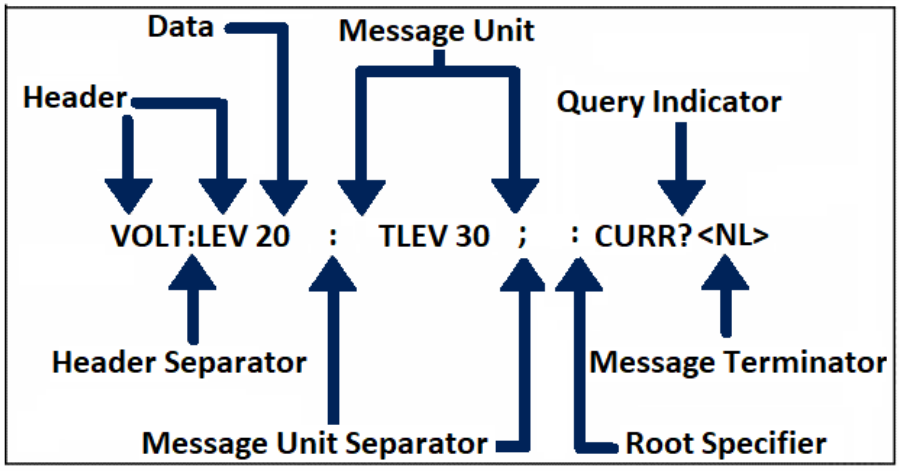

Figure 2.2 SCPI Message Structure

2.2.1 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>

2.2.2 Headers

Headers, also referred to as keywords, are instructions recognized by the electronic load. 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.

2.2.3 Query Indicator

Following a header with a question mark turns it into a query.

VOLTage?, CURRent:PROTection?

If a query contains a parameter, place the query indicator at the end of the last header.

CURRent:PROTection? MAX

2.2.4 Message Unit Separator

When two or more message units are combined into a compound message, separate the units with a semicolon.

STATus:OPERation?;QUEStionable?

2.2.5 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:

1. Newline (<NL>), which is ASCII decimal 10 or hex 0A.
2. End or identify (<END>).
3. Both of the above (<NL><END>).

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

2.2.6 Command Execution Rules

- Commands execute in the order that they are presented in the program message.
- An invalid command generates an error and 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.

2.3 SCPI Data

All data programmed to or returned from the electronic load is ASCII. The data may be numerical or a character string.

Symbol	Data Form
Talking Formats	
<NR1>	Digits with an implied decimal point assumed at the right of the least-significant digit. Example: 273
<NR2>	Digits with an explicit decimal point. Example: .0273
<NR3>	Digits with an explicit decimal point and an exponent. Example: 2.73E+2
Listening Formats	
<NRf>	Extended format that includes <NR1>, <NR2> and <NR3>. Example: 273 273. 2.73E2
<NRf+>	Expanded decimal format that includes <NRf> and MIN MAX DEF. Example: 273 273. 2.73E2 MAX. MIN and MAX are the minimum and maximum limit values that are implicit in the range specification for the parameter. DEF is the default values for the parameter.
<Bool>	Boolean Data. Example: 0 | 1 or ON | OFF

Table 2.1 Numeric Data Formats

2.4 Response Data Types

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

Class	Suffix	Unit	Unit with Multiplier
Amplitude	V	volt	MV (millivolt)
Current	A	amps	MA (milliamp)
Power	W	watt	MW (milliwatt)
Resistance	OHM	ohm	MOHM (megohm)
	R	ohm	MR(megohm)
Slew Rate	A/uS	amps/microsecond	
Time	S	second	MS (millisecond)
Common Multipliers			
	1E3	K	kilo
	1E-3	M	milli
	1E-6	U	micro

Table 2.2 Suffixes and Multipliers

<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.

Response Messages

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

Sending a Response Message

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

Multiple Response Messages

If more than one query command is sent in the same program message (see the paragraph entitled, "Multiple Command Messages"), the multiple response messages for all the queries is sent to the computer when the electronic load 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:

1. Always tell the electronic load what to send to the computer. The following two steps must always be performed to send information from the computer to the instrument:
   - Send the appropriate query command(s) in a program message.
   - Address the electronic load to talk.
2. The complete response message must be received by the computer before another program message can be sent to the electronic load.

2.5 SCPI Command Completion

SCPI commands sent to the electronic load are processed either sequentially or in parallel. Sequential commands finish execution before a subsequent command begins. Parallel commands allow other commands to begin executing while the parallel command is still executing. Commands that affect trigger actions are among the parallel commands.

The *WAI, *OPC, and *OPC? common commands provide different ways of indicating when all transmitted commands, including any parallel ones, have completed their operations. The syntax and parameters for these commands are described in Chapter 4. Some practical considerations for using these commands are as follows:

*WAI This prevents the electronic load from processing subsequent commands until all pending operations are completed.

*OPC? This places a 1 in the Output Queue when all pending operations have completed. Since it requires your program to read the returned value before executing the next program statement, *OPC? can be used to cause the controller to wait for commands to complete before proceeding with its program.

Note:

The trigger system must be in the Idle state in order for the status OPC bit to be true. Therefore, as far as triggers are concerned, OPC is false whenever the trigger system is in the Initiated state.

Using Device Clear

You can send a device clear at any time to abort a SCPI command that may be hanging up the GPIB interface. The status registers, error queue, and all configuration states are left unchanged when a device clear message is received. Device clear performs the following actions:

- The input and output buffers of the electronic load are cleared.
- The electronic load is prepared to accept a new command string.

The following statement shows how to send a device clear over the GPIB interface using GW BASIC:

```
CLEAR 705        IEEE-488 Device Clear
```

The following statement shows how to send a device clear over the GPIB interface using the GPIB command library for C or QuickBASIC:

```
IOCLEAR (705)
```

2.6 Language Dictionary Information

This section describes the syntax and parameters for all the IEEE 488.2 SCPI subsystem and common commands used by the electronic loads. Since the SCPI syntax remains the same for all programming languages, the examples given for each command are generic.

Presentation of Commands	
Syntax Forms	Syntax definitions use the long form, but only short form headers (or "keywords") appear in the examples. Use the long form to help make your program self-documenting.
Parameters	Most commands require a parameter and all queries will return a parameter. The range for a parameter may vary according to the model of electronic load. Parameters for all models are listed in the Specifications table in the User's Guide.
Channel	If a command only applies to individual channels of a mainframe, the entry Channel Selectable will appear in the command description.
Related Commands	Where appropriate, related commands or queries are included. These are listed because they are either directly related by function, or because reading about them will clarify or enhance your understanding of the original command or query.
Order of Presentation	The dictionary is organized as follows: ▪ Subsystem commands, arranged by subsystem ▪ IEEE 488.2 common commands

Table 2.3 Language Dictionary Information

2.7 Subsystem Commands

Subsystem commands are specific to functions. They can be a single command or a group of commands. The groups are comprised of commands that extend one or more levels below the root. The description of common commands follows the description of the subsystem commands.

The subsystem command groups are arranged according to function: Calibration, Channel, Input, List, Measurement, Port, Status, System, Transient, and Trigger. Commands under each function are grouped alphabetically under the subsystem. Commands followed by a question mark (?) take only the query form.

When commands take both the command and query form, this is noted in the syntax descriptions

IEEE-488 Command Refernce

Common commands begin with an * and consist of three letters (command) or three letters and a ? (query). They are defined by the IEEE 488.2 standard to perform common interface functions. Common commands and queries are categorized under System, Status, or Trigger functions and are listed at the end of this chapter.

Common Commands

Common commands begin with an * and consist of three letters (command) IEEE 488.2 standard to perform some common interface functions. The electronic loads respond to the required common commands that control status reporting, synchronization, and internal operations. The electronic loads also respond to optional common commands that control triggers, power-on conditions, and stored operating parameters.

Common commands and queries are listed alphabetically. If a command has a corresponding query that simply returns the data or status specified by the command, then both command and query are included under the explanation for the command. If a query does not have a corresponding command or is functionally different from the command, then the query is listed separately. The description for each common command or query specifies any status registers affected. Refer to section Programming the Status Registers, which explains how to read specific register bits and use the information that they return.

Mnemonic	Name	Description
*CLS	Clear Status	Clears all event registers and Error Queue.
*ESE<NRf>	Event Enable Command	Program the Standard Event Enable Register.
*ESE?	Event Enable Query	Read the Standard Event Enable Register.
*ESR?	EVent Status Query	Read the Standard Event Status Register and clear it.
*IDN	Identification Query	Return the manufacturer, model number, serial number, and firmware revision levels of the unit.
*OPC	Operation Complete Command	Set the Operation Complete bit in the Standard Event Status Register after all pending commands have been executed.
*OPC?	Operation Complete Query	Places an ASCII "1" into the output queue when all pending selected device operations have been completed.
*RCL<NRf>	Recall Command	Returns the electronic load to the setup configuration stored in the specified memory location.
*RDT?	Frame Query	Return the type of electronic frame.
*RST	Reset Command	Returns the electronic load to the *RST default conditions.
*SAV<NRf>	Save Command	Saves the current setup to the specified memory location.
*SRE<NRf>	Service Request Enable Command	Programs the Service Request Enable register.
*SRE?	Service Request Enable Query	Reads the Service Request Enable register.
*STB	Read Status Byte Query	Read the Status Byte register.
*TRG	Trigger Command	Send a trigger to the electronic load.
*TST?	Self-test Query	Wait until all previous commands are executed.
*WAI	Wait to Continue Command	Wait until all previous commands are executed.

Table 3.1 Common Commands

3.1 *CLS

Description This command clears the bits of the following registers:

- Standard Event Status
- Operation Status Event
- Questionable Status Event
- Status Byte
- Error Queue

Command Syntax *CLS

Parameters None

3.2 *ESE <NRf>

Description This command programs the Standard Event Status Enable register bits. The programming determines which events of the Standard Event Status Event Register (see *ESR?) are allowed to set the ESB (Event Summary Bit) of the Status Byte register. A "1" in the bit position enables the corresponding event. All of the enabled events of the Standard Event Status Event register are logically ORed to cause the Event Summary Bit (ESB) of the Status Byte Register to be set. See Section **Programming the Status Registers** for descriptions of the Standard Event Status registers. The query reads the Standard Event Status Enable register.

Command Syntax *ESE <NRf>

Parameters 0 to 255

Power-On Value See *PSC

Example *ESE 129

Query Syntax *ESE?

Returned Parameters <NR1>

Related Commands *ESR? *PSC *STB?

3.3 *ESR?

Description This query reads the Standard Event Status register. Reading the register clears it. The bit configuration of this register is the same as the Standard Event Status Enable register (see *ESE). See Section 1.4 Programming the Status Registers for a detailed explanation of this register.

Query Syntax *ESR?

Parameters None

Returned Parameters <NR1> (register value)

Related Commands *CLS *ESE *ESE? *OPC

3.4 *IDN?

Description This query requests the electronic load to identify itself. It returns the data in four fields separated by commas.

Query Syntax *IDN?

Parameters None

Returned Parameters <AARD>

Field	Information
B&K Precision	manufacturer
xxxxxx	model number
xxxxxxxxxxxxxxxxx	serial number or 0
x.xx	firmware revision

Example : BK PRECISION, MDL4U001, 600150010677510002, 1.43

3.5 *RDT?

Description This query requests the types of electronic load module. If channel does not exist, it returns 0. If channel exists, it returns the type.

Query Syntax *RDT?

Parameters None

Returned Parameters <AARD>

Example MDL4U200, 0, MDL4U305, 0, 0, 0, 0, 0

3.6 *OPC

Description This command causes the interface to set the OPC bit (bit 0) of the Standard Event Status register when the electronic load has completed all pending operations. (See *ESE command for the bit configuration of the Standard Event Status registers.) Pending operations are complete when:

- All commands sent before *OPC have been executed. This includes overlapped commands. Most commands are sequential and are completed before the next command is executed. Overlapped commands are executed in parallel with other commands. Commands that affect trigger actions are overlapped with subsequent commands sent to the electronic load. The *OPC command provides notification that all overlapped commands have been completed.
- All triggered actions are completed and the trigger system returns to the Idle state.

*OPC does not prevent processing of subsequent commands but bit 0 will not be set until all pending operations are completed. The query causes the interface to place an ASCII "1" in the Output Queue when all pending operations are completed.

Command Syntax *OPC

Parameters None

Query Syntax *OPC?

Returned Parameters <NR1>

Related Commands *TRIG *WAI

3.7 *RCL

Description This command restores the electronic load to a state that was previously stored in memory with a *SAV command to the specified location. All states are recalled with the following exceptions:

- CAL:STATe is set to OFF
- The trigger system is set to the Idle state by an implied ABORt command (this cancels any incomplete trigger actions)

Note:

The device state stored in location 0 is automatically recalled at power turn-on.

Command Syntax *RCL <NRf>

Parameters 0 to 9

Example *RCL3

Related Commands *PSC *RST *SAV

3.8 *RST

Description This command resets ALL channels of the electronic load to the following factory-defined states:

CURR MIN	RES:TRAN:BLEV MIN
CURR:MODE FIX	RES:TRAN:BWID MIN
CURR:PROT:DEL 3	RES:TRAN:MODE CONT
CURR:PROT:LEV MAX	SENS:AVER:COUN 8
CURR:PROT:STAT OFF	SENS:AVER:STAT 1
CURR:RANG MAX	SENS:FUNC:CURR DC
CURR:SLEW MAX	SENS:NPLC 7
CURR:TRAN:ALEV	SENS:VOLT:RANG:AUTO ON
CURR:TRAN:AWID	TRAC:FEED TWO
CURR:TRAN:BLEV	TRAC:FEED:MODE NEV
CURR:TRAN:BWID	TRAC:POIN 2
CURR:TRAN:MODE	CONT TRAN OFF
FUNC CURR	TRIG:COUN 1
FUNC:MODE FIX	TRIG:DEL 0
INP OFF	VOLT MAX
INP:SHOR OFF	VOLT:ON MIN
POW:CONF MAX	VOLT:ON:LATC ON
POW:PROT:DEL 3	VOLT:RANG MAX
RES MAX	VOLT:TRAN:ALEV MAX
RES:RANG MAX	VOLT:TRAN:AWID MIN
RES:TRAN:ALEV MAX	VOLT:TRAN:BWID MIN
RES:TRAN:AWID MIN	VOLT:TRAN:MODE CONT

Note:

- *RST does not clear any of the status registers or the error queue, and does not affect any interface error conditions.
- *RST sets the trigger system to the Idle state.
- *RST clears the presently active list.

Command Syntax *RST

Parameters None

Related Commands *PSC *SAV

3.9 *SAV

Description This command stores the present state of the electronic load to a specified location in memory. Up to 101 states can be stored.

If a particular state is desired at power-on, it should be stored in location 0. It will then be recalled at power-on if the power-on state is set to RCL0. Use *RCL to retrieve instrument states.

Command Syntax *SAV <NRf>

Parameters 0 to 100

Example *SAV 3

Related Commands *PSC *RST *RCL

3.10 *SRE

Description This command sets the condition of the Service Request Enable register. This register determines which bits from the Status Byte Register (see *STB for its bit configuration) are allowed to set the Master Status Summary (MSS) bit and the Request for Service (RQS) summary bit. A 1 in any Service Request Enable register bit position enables the corresponding Status Byte register bit and all such enabled bits are then logically ORed to cause Bit 6 of the Status Byte Register to be set.

When the controller conducts a serial poll in response to SRQ, the RQS bit is cleared, but the MSS bit is not. When *SRE is cleared (by programming it with 0), the electronic load cannot generate an SRQ to the controller. The query returns the current state of *SRE.

Command Syntax *SRE <NRf>

Parameters 0 to 255

Default Value See *PSC

Example *SRE 128

Query Syntax *SRE?

Returned Parameters <NR1> (register binary value)

Related Commands *ESE *ESR *PSC

3.11 *STB?

Description This query reads the Status Byte register, which contains the status summary bits and the Output Queue MAV bit. Reading the Status Byte register does not clear it. The input summary bits are cleared when the appropriate event registers are read (see Section 1.4 Programming the Status Registers for more information). A serial poll also returns the value of the Status Byte register, except that bit 6 returns Request for Service (RQS) instead of Master Status Summary (MSS). A serial poll clears RQS, but not MSS. When MSS is set, it indicates that the electronic load has one or more reasons for requesting service.

Query Syntax *STB?

Parameters None

Returned Parameters <NR1> (register value)

Related Commands *SRE *ESR *ESE

3.12 *TRG

Description This command generates a trigger to any system that has BUS selected as its source (for example, TRIG:SOUR BUS). The command has the same effect as the Group Execute Trigger (<GET>) command.

Command Syntax *TRG

Parameters None

Related Commands ABOR INIT TRIG:IMM

3.13 *TST?

Description This query causes the electronic load to do a self-test and report any errors.

Query Syntax TST?

Parameters None

Related Commands <NR1> 0 indicates the electronic load has passed self-test. Non-zero indicates an error code.

3.14 *WAI

Description This command instructs the electronic load not to process any further commands until all pending operations are completed. Pending operations are complete when:

- All commands sent before *WAI have been executed. This includes overlapped commands. Most commands are sequential and are completed before the next command is executed. Overlapped commands are executed in parallel with other commands. Commands that affect input voltage or state, relays, and trigger actions are overlapped with subsequent commands sent to the electronic load. The *WAI command prevents subsequent commands from being executed before any overlapped commands have been completed.
- All triggered actions are completed and the trigger system returns to the Idle state.

*WAI can be aborted only by sending the electronic load a GPIB DCL (Device Clear) command.

Command Syntax WAI?

Parameters None

Related Commands *OPC

Channel Subsystem

These commands program the channel selection capability of the electronic load. The CHANnel and INSTrument commands are equivalent.

4.1 CHANnel

Description This command selects the multiple electronic load channels to which all subsequent channel-specific commands will be directed. If the specified channel number does not exist or is outside the MIN/MAX range, an error code is generated (see Chapter 5). Refer to the installation section of the User's Guide for more information about channel number assignments. Channel 11-18 refers to channel 1-8 of MDL4U002 mainframe extension.

Command Syntax CHANnel <NR1>

Parameters 1-8, 11-18

***RST Value** MINimum

Examples CHAN 3

Query Syntax CHANnel?

Returned Parameters <NR1>

4.2 CHANnel:ID?

Channel Specific

Description This query reads the model numbers of the modules installed in the mainframe. It returns the data in comma-separated fields.

Query Syntax CHANnel:ID?

Parameters None

Returned Parameters <AARD>

Field	Information
xxxxxx	model number
xxxxxxxxxxxxxxxxx	serial number or 0
Verx.xx-x.xx	firmware revision

Example : MDL4U600, 600156010677530001, Ver1.35-1.20

Trigger Subsystem

The trigger subsystem is made up of a series of commands and subsystems to configure the trigger model.

5.1 TRIGer:SOURce

Description This command selects the trigger source. This command is not channel specific and applies to the entire mainframe.

Source	Description
BUS	Accepts a GPIB <GET> signal or a *TRG command as the trigger source. This selection guarantees that all previous commands are complete before the trigger occurs.
EXTernal	Selects the electronic load's trigger input as the trigger source. This trigger is processed as soon as it is received.
HOLD	Only the TRIG:IMM command will generate a trigger in HOLD mode. All other trigger commands are ignored.
MANUal	The event occurs when the Trig key is pressed.
TIMer	This generates triggers that are in synchronization with the electronic load's internal oscillator as the trigger source. The internal oscillator begins running as soon as this command is executed. Use TRIG:TIM to program the oscillator period.

Table 5.1

Command Syntax TRIGger:SOURce <CRD>

Parameters BUS | EXTernal | HOLD | MANUal | TIMer

***RST Value** MANUal

Examples TRIG:SOUR BUS TRIG:SOUR EXT

Query Syntax TRIGger:SOURce?

Returned Parameters <CRD>

Related Commands ABOR TRIG TRIG:DEL

5.2 TRIGger:TIMer

Description This command specifies the period of the triggers generated by the internal trigger generator. This command is not channel specific and applies to the entire mainframe.

Command Syntax TRIGger:TIMer <NRf+>

Parameters 1 to 999.99s | MINimum | MAXimum | DEFault

Unit seconds

***RST Value** 0.001

Examples TRIG:TIM 0.25 TRIG:TIM MAX

Query Syntax TRIGger:TIMer? [MINimum | MAXimum | DEFault]

Returned Parameters <NR3>

Related Commands ABOR TRIG TRIG:SOUR TRIG:DEL

System Subsystem

System commands control the system-level functions of the electronic load that are not directly related to input control or measurement functions.

6.1 SYSTem:PRESet

Description This command returns the instrument to states optimized for front panel operation.

Command Syntax SYSTem:PRESet

Parameters None

6.2 SYSTem:POSetup

Description This command is used to 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.

Command Syntax SYSTem:POSetup <CRD>

Parameters RST | SAV0

***RST Value** RST

Examples SYST:POS RST

Query Syntax SYSTem:POSetup?

Returned Parameters <CRD>

Related Commands *RST *SAV

6.3 SYSTem:VERSion?

Description This query returns the SCPI version number to which the electronic load complies. The value is of the form YYYY.V, where YYYY is the year and V is the revision number for that year.

Query Syntax SYSTem:VERSion?

Parameters None

Examples SYST:VERS?

Returned Parameters <NR2>

6.4 SYSTem:ERRor?

Description This query returns the next error number followed by its corresponding error message string from the remote programming error queue. The queue is a FIFO (first-in, first-out) buffer that stores errors as they occur. As it is read, each error is removed from the queue. When all errors have

been read, the query returns "0, No Error". If more errors are accumulated than the queue can hold, the last error in the queue is "-350, Too Many Errors".

Query Syntax SYSTem:ERRor?

Parameters None

Examples SYST:ERR?

Returned Parameters <NR1>, <SRD>

6.5 SYSTem:CLEar

Description This action command is used to clear the Error Queue of messages.

Command Syntax SYSTem:CLEar

Parameters None

Examples SYST:CLE

Related Commands SYST:ERR?

6.6 SYSTem:LOCal

Description This command places the electronic load in local mode during RS-232 operation. The front panel keys are functional.

Command Syntax SYSTem:LOCal

Parameters None

Examples SYST:LOC

Related Commands SYST:REM SYST:RWL

6.7 SYSTem:REMote

Description This command places the electronic load in remote mode during RS-232 operation. This disables all front panel keys except the Local key. Pressing the Local key while in the remote state returns the front panel to the local state.

Command Syntax SYSTem:REMote

Parameters None

Examples SYST:REM

Related Commands SYST:LOC SYST:RWL

6.8 SYSTem:RWLock

Description This command places the electronic load in remote mode during RS-232 operation. All front panel keys including the Local key are disabled. Use SYSTem:LOCal to return the front panel to the local state.

Command Syntax SYSTem:RWLock

Parameters None

Examples SYST:RWL

Related Commands SYST:REM SYST:LOC

Status Subsystem

These commands program the electronic load's status registers. The electronic load has five groups of status registers:

1. Channel Status
2. Channel Summary
3. Questionable Status
4. Standard Event Status
5. Operation Status

Refer to Section **1.4 Programming the Status Registers** for more information.

7.1 STATus:CHANnel?

Channel Specific

Description This query returns the value of the Channel Event register. The Event register is a read-only register, which holds (latches) all events that are passed into it. Reading the Channel Event register clears it.

Query Syntax STATus:CHANnel[:EVENt]?

Parameters None

Examples STAT:CHAN:EVEN?

Returned Parameters <NR1> (register value)

Related Commands *CLS

7.2 STATus:CHANnel:CONDition?

Channel Specific

Description This query returns the value of the Channel Condition register. The particular channel must first be selected by the CHAN command.

Query Syntax STATus:CHANnel:CONDition?

Parameters None

Examples STAT:CHAN:COND?

Returned Parameters <NR1> (register value)

Related Commands STAT:CHAN?

7.3 STATus:CHANnel:ENABle

Channel Specific

Description This command sets or reads the value of the Channel Enable register for a specific channel. The particular channel must first be selected by the CHAN command.

Command Syntax STATus:CHANnel:ENABle <NR1>

Parameters 0 to 65535

Examples STAT:CHAN:ENAB 3

Query Syntax STATus:CHANnel:ENABle?

Returned Parameters <NR1> (register value)

Related Commands *CLS

7.4 STATus:CSUM?

Description This query returns the value of the Channel Event summary register. The bits in this register correspond to a summary of the channel register for each input channel. Reading the Channel Event summary register clears it. This command is not channel specific and applies to the entire mainframe.

Query Syntax STATus:CSUMmary[:EVENt]?

Parameters None

Examples STAT:CSUM:EVEN?

Returned Parameters <NR1> (register value)

Related Commands *CLS

7.5 STATus:CSUMmary:ENABle

Description This command sets or reads the value of the Channel Enable summary register. This command is not channel specific and applies to the entire mainframe.

Command Syntax STATus:CSUMmary:ENABle <NR1>

Parameters 0 to 255

Examples STAT:CSUM:ENAB 3

Query Syntax STATus:CSUMmary:ENABle?

Returned Parameters <NR1> (register value)

Related Commands *CLS

7.6 STATus:OPERation?

Description This query returns the value of the Operation Event register. The Event register is a read-only register that holds (latches) 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 and applies to the entire mainframe.

Query Syntax STATus:OPERation[:EVENt]?

Parameters None

Examples STAT:OPER:EVEN?

Returned Parameters <NR1> (register value)

Related Commands *CLS

7.7 STATus:OPERation:CONDition?

Description This query returns the value of the Operation Condition register. This is a read-only register that holds the real-time (unlatched) operational status of the electronic load. This command is not channel specific and applies to the entire mainframe.

Query Syntax STATus:OPERation:CONDition?

Parameters None

Examples STAT:OPER:COND?

Returned Parameters <NR1> (register value)

Related Commands STAT:QUES:COND?

7.8 STATus:OPERation:ENABle

Description This command and its query can be used to set 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 (OPER) 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 and applies to the entire mainframe.

Command Syntax STATus:OPERation:ENABle <NR1>

Parameters 0 to 65535

Default Value 0

Examples STAT:OPER:ENAB 32 STAT:OPER:ENAB 1

Query Syntax STATus:OPERation:ENABle?

Returned Parameters <NR1> (register value)

Related Commands STAT:OPER?

7.9 STATus:QUEStionable?

Description This query returns the value of the Questionable Event register. The Event register is a read-only register that holds (latches) all events that pass into it. Reading the Questionable Event register clears it. This command is not channel specific and applies to the entire mainframe.

Query Syntax STATus:QUEStionable[:EVENt]?

Parameters None

Examples STAT:QUES:EVEN?

Returned Parameters <NR1> (register value)

7.10 STATus:QUEStionable:CONDition?

Description This query returns the value of the Questionable Condition register. This is a read-only register that holds the real-time (unlatched) questionable status of the electronic load. This command is not channel specific and applies to the entire mainframe.

Query Syntax STATus:QUEStionable:CONDition?

Parameters None

Examples STAT:QUES:COND?

Returned Parameters <NR1> (register value)

Related Commands STAT:OPER:COND?

7.11 STATus:QUEStionable:ENABle

Description This command 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 (QUES) 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 and applies to the entire mainframe.

Command Syntax STATus:QUEStionable:ENABle <NR1>

Parameters 0 to 65535

Default Value 0

Examples STAT:QUES:ENAB 32 STAT:QUES:ENAB 1

Query Syntax STATus:QUEStionable:ENABle?

Returned Parameters <NR1> (register value)

Related Commands STAT:QUES?

7.12 STATus:PRESet

Description When this command is sent, the SCPI event registers are affected as follows: All bits of the following registers are cleared to zero (0):

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

Note:

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

Command Syntax STATus:PRESet

Parameters None

Examples STAT:PRES

Trace Subsystem

The commands in this subsystem are used to configure and control data storage into the buffer.

8.1 TRACe:CLEar

Channel Specific

Description This action command is used to clear the buffer of readings. If you do not clear the buffer, a subsequent store will overwrite the old readings. If the subsequent store is aborted before the buffer becomes full, you could end up with some "old" readings still in the buffer.

Command Syntax TRACe:CLEar

Parameters None

Example TRAC:CLE

8.2 TRACe:FREE?

Channel Specific

Description This command is used to read the status of storage memory. After sending this command and addressing the electronic load to talk, two values separated by commas are sent to the computer. The first value indicates how many bytes of memory are available, and the second value indicates how many bytes are reserved to store readings.

Query Syntax TRACe:FREE?

Returned Parameters <NR1>, <NR1>

Examples TRAC:FREE?

8.3 TRACe:POINts

Channel Specific

Description This command is used to specify the size of the buffer.

Command Syntax TRACe:POINts <NRf+>

Parameters 0 to 1000 | MINimum | MAXimum | DEFault

***RST Value** 1000

Examples TRAC:POIN 10

Query Syntax TRACe: POINts? { MINimum | MAXimum | DEFault }

Returned Parameters <NR1>

Related Commands TRAC:FEED

8.4 TRACe:FEED

Channel Specific

Description This command is used to select the source of readings to be placed in the buffer. With VOLTage selected, voltage readings are placed in the buffer (TRAC:POIN maximum value is 1000). With CURRent selected, current readings are placed in the buffer (TRAC:POIN maximum value is 1000). With TWO selected, voltage and current are placed in the buffer when storage is performed (TRAC:POIN maximum value is 500).

Command Syntax TRACe:FEED <CRD>

Parameters VOLTage | CURRent | TWO

***RST Value** TWO

Examples TRAC:FEED VOLT

Query Syntax TRACe:FEED?

Returned Parameters <CRD>

Related Commands TRAC:POIN

8.5 TRACe:FEED:CONTrol

Channel Specific

Description This command is used to select the buffer control. With NEVer selected, storage into the buffer is disabled. When NEXT is selected, the storage process starts, fills the buffer and then stops. The buffer size is specified by the :POINts command.

Command Syntax TRACe:FEED:CONTrol <CRD>

Parameters NEVer | NEXT

***RST Value** NEVer

Examples TRAC:FEED:CONT NEXT

Query Syntax TRACe: FEED:CONT?

Returned Parameters <CRD>

Related Commands TRAC:FEED

8.6 TRACe:DATA?

Channel Specific

Description When this command is sent and the electronic load is addressed to talk, all the readings stored in the buffer are sent to the computer.

Query Syntax TRACe:DATA?

Returned Parameters <NR3>, <NR3>…etc, or <NR3> <NR3>, <NR3> <NR3> …etc (if TRACe:FEED TWO is set)

8.7 TRACe:FILTer

Channel Specific

Description This command is used to select whether the data in buffer is filtered data.

Command Syntax TRACe:FILTer:STATe <BOOL>

Parameters 0 | 1 | ON | OFF

***RST Value** OFF

Examples TRAC:FILT 1

Query Syntax TRACe:FILTer:STATe?

Returned Parameters <NR1>

8.8 TRACe:DELay

Channel Specific

Description This command is used to select the delay time for trigger in buffer.

Command Syntax TRACe:DELay <NRf>

Parameters {0 to 3600s | MINimum | MAXimum | DEFault}

Unit S (second)

***RST Value** 0

Examples TRAC:DEL 1

Query Syntax TRACe:DELay? {MINimum | MAXimum | DEFault}

Returned Parameters <NR3>

8.9 TRACe:TIMer

Channel Specific

Description This command is used to select the interval for timer.

Command Syntax TRACe:TIMer <NRf>

Parameters {0.0002 to 3600s | MINimum | MAXimum | DEFault}

Unit S (second)

***RST Value** 1

Examples TRAC:TIM 0.1

Query Syntax TRACe:TIMer? {MINimum | MAXimum | DEFault}

Returned Parameters <NR3>

Source Subsystem

These commands control the input of the electronic load. The INPut and OUTput commands are equivalent. The CURRent, RESistance, and VOLTage commands program the actual input current, resistance, and voltage.

9.1 [SOURce:]INPut:ALL

Description This command enables or disables all the inputs of the module. The state of a disabled input is a high impedance condition.

Command Syntax [SOURce:]INPut:ALL[:STATe] <bool>

Parameters 0 | 1 | OFF | ON

Examples INP:ALL 1

9.2 [SOURce:]INPut

Channel Specific

Description This command enables or disables the electronic load inputs. The state of a disabled input is a high impedance condition.

Command Syntax [SOURce:]INPut[:STATe] <bool>

Parameters 0 | 1 | OFF | ON

***RST Value** OFF

Examples INP 1

Query Syntax INPut[:STATe]?

Returned Parameters 0 | 1

Related Commands *RCL *SAV

9.3 [SOURce:]INPut:SYNCon

Channel Specific

Description This command enables or disables to change the electronic load inputs when INP:ALL command is received.

Command Syntax [SOURce:]INPut:SYNCon[:STATe] <bool>

Parameters 0 | 1 | OFF | ON

***RST Value** ON

Examples INP:SYNC 1

Query Syntax INPut:SYNCon[:STATe]?

Returned Parameters 0 | 1

Related Commands INP:ALL

9.4 [SOURce:]INPut:SHORt

Channel Specific

Description This command programs the specified electronic load module to the maximum current that it can sink in the present operating range.

Command Syntax [SOURce:]INPut:SHORt[:STATe] <bool>

Parameters 0 | 1 | OFF | ON

***RST Value** OFF

Examples INP:SHOR 1

Query Syntax INPut:SHORt:STATe?

Returned Parameters 0 | 1

Related Commands INP

9.5 [SOURce:]REMote:SENSe

Channel Specific

Description This command is used to select the remote measure mode for the load.

Command Syntax [SOURce:]REMote:SENSe[:STATe] <bool>

Parameters 0 | 1 | OFF | ON

***RST Value** 0

Examples REM:SENS 0

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

Returned Parameters <CRD>

9.6 [SOURce:]FUNCtion

Channel Specific

Description This command selects the input regulation mode of the electronic load.

CURRent	constant current mode
RESistance	constant resistance mode
VOLTage	constant voltage mode
POWer	constant power mode
IMPEDANCE	constant impedance mode

Command Syntax [SOURce:] FUNCtion <function>

Parameters CURRent | RESistance | VOLTage | POWer | IMPEDANCE

***RST Value** CURRent

Examples FUNC RES

Query Syntax [SOURce:]FUNCtion?

Returned Parameters <CRD>

9.7 [SOURce:]FUNCtion:MODE

Channel Specific

Description This command determines whether the input regulation mode is controlled by values in a list or by the FUNCtion command setting.

FIXed	The regulation mode is determined by the FUNCtion or MODE command.
LIST	The regulation mode is determined by the active list.

Command Syntax [SOURce:]FUNCtion:MODE <mode>

Parameters FIXed | LIST

***RST Value** FIXed

Examples FUNC:MODE FIX

Query Syntax [SOURce:]FUNCtion:MODE?

Returned Parameters <CRD>

Related Commands FUNC

9.8 [SOURce:]TRANsient

Channel Specific

Description This command turns the transient generator on or off.

Command Syntax [SOURce:]TRANsient[:STATe] <bool>

Parameters 0 | 1 | OFF | ON

***RST Value** OFF

Examples TRAN 1

Query Syntax [SOURce:]TRANsient[:STATe]?

Returned Parameters 0 | 1

Related Commands CURR:TRAN:CURR:MODE CURR:TRAN:ALEV

9.9 [SOURce:]PROTection:CLEar

Channel Specific

Description This command clears the latch that disables the input when a protection condition such as overvoltage (OV) or overcurrent (OC) is detected. All conditions that generated the fault must be removed before the latch can be cleared. The input is then restored to the state it was in before the fault condition occurred.

Command Syntax [SOURce:]INPut:PROTection:CLEar

Parameters None

Examples :PROT:CLE

9.10 [SOURce:]INPut:TIMer

Channel Specific

Description These commands enable or disable the electronic load on timer.

Command Syntax [SOURce:]INPut:TIMer[:STATe] <bool>

Parameters 0 | 1 | OFF | ON

***RST Value** OFF

Examples INP:TIM 1

Query Syntax INPut:TIMer[:STATe]?

Returned Parameters 0 | 1

Related Commands INP:TIM:DEL

9.11 [SOURce:]INPut:TIMer:DELay

Channel Specific

Description This command specifies the timer setting.

Command Syntax [SOURce:]INPut:TIMer <NRf+>

Parameters 1 to 60000s | MINimum | MAXimum | DEFault

Unit seconds

***RST Value** 10

Examples INP:TIM:DEL 5

Query Syntax [SOURce:]INPut:TIMer:DELay? [MINimum | MAXimum | DEFault]

Returned Parameters <NR3>

Related Commands INP:TIM

9.12 [SOURce:]CURRent

Channel Specific

Description This command sets the current that the load will regulate when operating in constant current mode.

Command Syntax [SOURce:]CURRent[:LEVel][:IMMediate] <NRf+>

Parameters 0 through max rated current

Unit A (amps)

***RST Value** MINimum

Examples CURR 5 CURR:LEV 0.5

Query Syntax [SOURce:]CURRent[:LEVel][:IMMediate]?

Returned Parameters <NR3>

Related Commands CURR:RANG

9.13 [SOURce:]CURRent:RANGe

Channel Specific

Description This command sets the current range of the electronic load module. There are two current ranges, high range (model dependent) and low range (model dependent).

When you program a range value, the load automatically selects the range that corresponds to the value that you program. If the value falls in a region where ranges overlap, the electronic load selects the range with the highest resolution.

Note:

When this command is executed, the IMMediate, TRANsient, TRIGgered, and SLEW current settings are adjusted as follows.

- If existing settings are within new range, no adjustment is made.
- If existing settings are outside new range, the levels are set to the maximum value of the new range.

Command Syntax [SOURce:]CURRent:RANGe <NRf+>

Parameters 0 through max. Rated current

Unit A (amps)

***RST Value** max. current (high range)

Examples SOUR:CURR:RANGE 5

Query Syntax [SOURce:]CURRent:RANGe

Returned Parameters <NR3>

Related Commands CURR CURR:SLEW

9.14 [SOURce:]CURRent:SLEW

Channel Specific

Description This command sets the slew rate for all programmed changes in the input current level of the electronic load. This command programs both positive and negative slew rates. MAXimum sets the slew to the fastest possible rate. MINimum sets the slew to the slowest rate.

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

Parameters See specifications

Unit A (amps per micro second)

Examples CURR:SLEW MAX

Related Commands CURR CURR:NEG CURR:SLEW:POS

9.15 [SOURce:]CURRent:SLEW:POSitive

Channel Specific

Description This command sets the slew rate of the current for positive transitions. MAXimum sets the slew to the fastest possible rate. MINimum sets the slew to the slowest rate.

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

Parameters See specifications

Unit A (amps per micro second)

Examples CURR:SLEW:POS 0.0001

Query Syntax [SOURce:]CURRent:SLEW:POSitive?

Returned Parameters <NR3>

Related Commands CURR:SLEW

9.16 [SOURce:]CURRent:SLEW:NEGative

Channel Specific

Description This command sets the slew rate of the current for negative transitions. MAXimum sets the slew to the fastest possible rate. MINimum sets the slew to the slowest rate.

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

Parameters see specifications

Unit A (amps per micro second)

Examples CURR:SLEW:NEG 0.0001

Query Syntax [SOURce:]CURRent:SLEW:NEGative?

Returned Parameters <NR3>

Related Commands CURR:SLEW

9.17 [SOURce:]CURRent:PROTection:STATe

Channel Specific

Description This command enables or disables the overcurrent protection feature.

Command Syntax [SOURce:]CURRent:PROTection:STATe <Bool>

Parameters 0 | 1 | OFF | ON

***RST Value** OFF

Examples CURR:PROT:STAT 1

Query Syntax [SOURce:]CURRent:PROTection:STATe?

Returned Parameters <NR3>

Related Commands CURR:PROT

9.18 [SOURce:]CURRent:PROTection

Channel Specific

Description This command sets the soft current protection level. If the input current exceeds the soft current protection level for the time specified by CURR:PROT:DEL, the input is turned off.

Note:

Use CURR:PROT:DEL to prevent momentary current limit conditions caused by programmed changes from tripping the overcurrent protection.

Command Syntax [SOURce:]CURRent:PROTection:LEVel <NRf+>

Parameters 0 through max. Rated current

Unit A (amps)

Examples CURR:PROT 2

Query Syntax [SOURce:]CURRent:PROTection:LEVel?

Returned Parameters <NR3>

Related Commands CURR:PROT:DEL CURR:PROT:STAT

9.19 [SOURce:]CURRent:PROTection:DELay

Channel Specific

Description This command specifies the time that the input current can exceed the protection level before the input is turned off.

Command Syntax [SOURce:]CURRent:PROTection[:DELay] <NRf+>

Parameters 0 to 60 seconds

Unit seconds

***RST Value** 0

Examples CURR:PROT:DEL 5

Query Syntax [SOURce:]CURRent:PROTection:DELay?

Returned Parameters <NR1>

Related Commands CURR:PROT CURR:PROT:STAT

9.20 [SOURce:]CURRent:TRANsient:MODE

Channel Specific

Description This command selects the operating mode of the transient generator as follows in constant current mode.

CONTinuous	The transient generator puts out a continuous pulse stream after receipt of a trigger.
PULSe	The transient generator puts out a single pulse upon receipt of a trigger.
TOGGle	The transient generator toggles between two levels upon receipt of a trigger.

Command Syntax [SOURce:]CURRent:TRANsient:MODE <mode>

Parameters CONTinuous | PULSe | TOGGle

***RST Value** CONTinuous

Examples CURR:TRAN:MODE TOGG

Query Syntax [SOURce:]CURRent:TRANsient:MODE?

Returned Parameters <CRD>

Related Commands CURR:TRAN:ALEV TRAN

9.21 [SOURce:]CURRent:TRANsient:ALEVel [SOURce:]CURRent:TRANsient:BLEVel

Channel Specific

Description These commands specify the transient level of the input current. The transient function switches between level A and level B.

Command Syntax [SOURce:]CURRent:TRANsient:ALEVel <NRf+> [SOURce:]CURRent:TRANsient:BLEVel <NRf+>

Parameters 0 through max. Rated current

Unit A (amps)

Examples CURR:TRAN:ALEV 5 CURR:TRAN:BLEV 0.5

Query Syntax [SOURce:]CURRent:TRANsient:ALEVel? [SOURce:]CURRent:TRANsient:BLEVel

Returned Parameters <NR3>

Related Commands CURR:

9.22 [SOURce:]CURRent:TRANsient:AWIDth [SOURce:]CURRent:TRANsient:BWIDth

Channel Specific

Description These commands specify the transient pulse width of the input current.

Command Syntax [SOURce:]CURRent:TRANsient:AWIDth <NRf+> [SOURce:]CURRent:TRANsient:BWIDth <NRf+>

Parameters 1 to 65535 us

Unit S (seconds)

***RST Value** 500uS

Examples CURR:TRAN:AWID 0.001 CURR:TRAN:BLEV 0.02

Query Syntax [SOURce:]CURRent:TRANsient:AWIDth? [SOURce:]CURRent:TRANsient:BWIDth

Returned Parameters <NR3>

Related Commands CURR:

9.23 [SOURce:]CURRent:HIGH [SOURce:]CURRent:LOW

Channel Specific

Description These commands set the high and low voltage level when the load is in constant current mode.

Command Syntax [SOURce:]CURRent:HIGH <NRf+> [SOURce:]CURRent:LOW <NRf+>

Parameters see specifications for voltage range

Unit V (volts)

Examples CURR:HIGH 5

Query Syntax [SOURce:]CURRent:HIGH? [SOURce:]CURRent:LOW?

Returned Parameters <NR3>

9.24 [SOURce:]VOLTage

Channel Specific

Description This command sets the voltage that the electronic load will regulate when operating in constant voltage (CV) mode.

Command Syntax [SOURce:]VOLTage[:LEVel][:IMMediate] <NRf+>

Parameters See specifications for range

Unit V (volts)

Examples VOLT 5

Query Syntax [SOURce:]VOLTage[:LEVel][:IMMediate]?

Returned Parameters <NR3>

Related Commands VOLT:RANG

9.25 [SOURce:]VOLTage:RANGe

Channel Specific

Description This command sets the voltage range of the electronic load module. There is only one voltage range.

Command Syntax [SOURce:] VOLTage:RANGe <NRf+>

Parameters see specifications for range

Unit V (volts)

Examples VOLT:RANG 15

Query Syntax [SOURce:]VOLTage:RANGe?

Returned Parameters <NR3>

Related Commands VOLT

9.26 [SOURce:]VOLTage:RANGe:AUTO

Channel Specific

Description This command sets the voltage auto range state of the electronic load module.

Command Syntax [SOURce:]VOLTage:RANGe:AUTO<bool>

Parameters 0 | 1 | ON | OFF

***RST Value** 1

Examples VOLT:RANG:AUTO 1

Query Syntax [SOURce:]VOLTage:RANGe:AUTO?

Returned Parameters <NR1>

9.27 [SOURce:]VOLTage:ON

Channel Specific

Description This command sets the voltage of sink current on.

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

Parameters 0 through max. rated voltage.

Unit V (volts)

Examples VOLT 5

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

Returned Parameters <NR3>

Related Commands VOLT:LATCh

9.28 [SOURce:]VOLTage:LATCh

Channel Specific

Description This command sets the action type of Von.

Command Syntax [SOURce:]VOLTage:LATCh[:STATe] <b>

Parameters 0 | 1 | ON | OFF

***RST Value** ON

Examples VOLT:LATC 1

Query Syntax [SOURce:] VOLTage:LATCh[:STATe]?

Returned Parameters 0 | 1

Related Commands VOLT:ON

9.29 [SOURce:]VOLTage:HIGH [SOURce:]VOLTage:LOW

Channel Specific

Description These commands set the high and low current limit when the load is in constant voltage mode.

Command Syntax [SOURce:]VOLTage:HIGH <NRf+> [SOURce:]VOLTage:LOW <NRf+>

Parameters 0 to max. rated current.

Unit A (amps)

Examples VOLT:HIGH 5

Query Syntax [SOURce:]VOLTage:HIGH? [SOURce:]VOLTage:LOW?

9.30 [SOURce:]VOLTage:TRANsient:MODE

Channel Specific

Description This command selects the operating mode of the transient generator as follows in constant voltage mode.

CONTinuous PULSe TOGGle

The transient generator puts out a continuous pulse stream after receipt of a trigger.

The transient generator puts out a single pulse upon receipt of a trigger.

The transient generator toggles between two levels upon receipt of a trigger.

Command Syntax [SOURce:]VOLTage:TRANsient:MODE <mode>

Parameters CONTinuous | PULSe | TOGGle

***RST Value** CONTinuous

Examples VOLT:TRAN:MODE PULS

Query Syntax [SOURce:] VOLTage:TRANsient:MODE?

Returned Parameters <CRD>

Related Commands VOLT:TRAN:ALEV TRAN

9.31 [SOURce:]VOLTage:TRANsient:ALEVel [SOURce:]VOLTage:TRANsient:BLEVel

Channel Specific These commands specify the transient level of the input voltage. The transient function switches between level A and level B.

Command Syntax [SOURce:]VOLTage:TRANsient:ALEVel <NRf+> [SOURce:]VOLTage:TRANsient:BLEVel <NRf+>

Parameters 0 to max. rated voltage

Unit V (volts)

Examples VOLT:TRAN:ALEV 5 VOLT:TRAN:BLEV 0.5

Query Syntax [SOURce:]VOLTage:TRANsient:ALEVel? [SOURce:]VOLTage:TRANsient:BLEVel?

Returned Parameters <NR3>

Related Commands VOLT

9.32 [SOURce:]VOLTage:TRANsient:AWIDth [SOURce:]VOLTage:TRANsient:BWIDth

Channel Specific

Description This command specifies the transient pulse width of the input voltage.

Command Syntax [SOURce:]VOLTage:TRANsient:AWIDth <NRf+> [SOURce:]VOLTage:TRANsient:BWIDth <NRf+>

Parameters 1 to 65535 us

Unit S (second)

***RST Value** 500uS

Examples VOLT:TRAN:AWID 0.001 VOLT:TRAN:BWID 0.02

Query Syntax [SOURce:] VOLTage:TRANsient:AWIDth? [SOURce:]VOLTage:TRANsient:BWIDth?

Returned Parameters <NR3>

Related Commands VOLT

9.33 [SOURce:]RESistance

Channel Specific

Description This command sets the resistance of the electronic load when operating in constant resistance mode.

Command Syntax [SOURce:]RESistance[:LEVel][:IMMediate] <NRf+>

Parameters 0 to max. rated resistance

Unit Ω (ohms)

Examples RES 5 RES:LEV 3.5

Query Syntax [SOURce:]RESistance[:LEVel][:IMMediate]?

Returned Parameters <NR3>

Related Commands RES:RANG

9.34 [SOURce:]RESistance: RANGe

Channel Specific

Description This command sets the resistance range of the electronic load module. This limits the value the resistance can be set to.

When you program a range value, the electronic load automatically selects the range that corresponds to the value that you program. If the value falls in a region where ranges overlap, the electronic load selects the range with the highest resolution.

Note:

When this command is executed, the IMMediate, TRANsient, TRIGgered, and SLEW resistance settings are adjusted as follows. If existing settings are within new range, no adjustment is made. If existing settings are outside new range, the levels are set to either the maximum or minimum value of the new range, depending on which they are closest to

---

Command Syntax [SOURce:]RESistance:RANGe <NRf+>

Parameters 0 to max. rated resistance

Unit Ω (ohms)

Examples RES:RANG 15 SOUR:RES:RANGE 20

Query Syntax [SOURce:]RESistance:RANGe?

Returned Parameters <NR3>

Related Commands RES RES:SLEW

9.35 [SOURce:]RESistance:HIGH [SOURce:]RESistance:LOW

Channel Specific

Description This command is used to set the voltage high and low limit determined when the load is in constant resistance mode.

Command Syntax [SOURce:]RESistance:HIGH <NRf+> [SOURce:]RESistance:LOW <NRf+>

Parameters 0 to max. rated voltage

Unit V (volts)

Examples RES:HIGH 5

Query Syntax [SOURce:]RESistance:HIGH? [SOURce:]RESistance:LOW?

Returned Parameters <NR3>

9.36 [SOURce:]RESistance:TRANsient:MODE

Channel Specific

Description This command selects the operating mode of the transient generator as follows in constant resistance mode.

CONTinuous	The transient generator puts out a continuous pulse stream upon receipt of a trigger.
PULSe	The transient generator puts out a single pulse upon receipt of a trigger.
TOGGle	The transient generator toggles between two levels upon receipt of a trigger.

Command Syntax [SOURce:]RESistance:TRANsient:MODE <mode>

Parameters CONTinuous | PULSe | TOGGle

***RST Value** CONTinuous

Examples RES:TRAN:MODE PULS

Query Syntax [SOURce:]RESistance:TRANsient:MODE?

Returned Parameters <CRD>

Related Commands RES:TRAN:ALEV TRAN

9.37 [SOURce:]RESistance:TRANsient:ALEVel [SOURce:]RESistance:TRANsient:BLEVel

Channel Specific

Description This command specifies the transient level of the input resistance. The transient function switches between level A and level B.

Command Syntax [SOURce:]RESistance:TRANsient:ALEVel <NRf+> [SOURce:]RESistance:TRANsient:BLEVel <NRf+>

Parameters 0 to max. rated resistance.

Unit Ω (ohms)

Examples RES:TRAN:ALEV 5 RES:TRAN:BLEV 0.5

Query Syntax [SOURce:]RESistance:TRANsient:ALEVel? [SOURce:]RESistance:TRANsient:BLEVel?

Returned Parameters <NR3>

Related Commands RES

9.38 [SOURce:]RESistance:TRANsient:AWIDth [SOURce:]RESistance:TRANsient:BWIDth

Channel Specific

Description This command specifies the transient pulse width of the input resistance.

Command Syntax [SOURce:]RESistance:TRANsient:AWIDth <NRf+> [SOURce:]RESistance:TRANsient:BWIDth <NRf+>

Parameters 1 to 65535 us

Unit S (second)

***RST Value** 500 uS

Examples RES:TRAN:AWID 0.001 RES:TRAN:BWID 0.02

Query Syntax [SOURce:]RESistance:TRANsient:AWIDth? [SOURce:]RESistance:TRANsient:BWIDth?

Returned Parameters <NR3>

Related Commands RES

9.39 [SOURce:]POWer

Channel Specific

Description This command sets the power of the load when operating in constant power mode.

Command Syntax [SOURce:]POWer[:LEVel][:IMMediate] <NRf+>

Parameters 0 to max. rated power

Unit W (power)

***RST Value** MINimum

Examples POW 5 POW:LEV 3.5

Query Syntax [SOURce:]POWer[:LEVel][:IMMediate]?

Returned Parameters <NR3>

Related Commands POW:RANG

9.40 [SOURce:]POWer:RANGe

Channel Specific

Description This command sets the power range of the load. This limits the settable range for power.

Command Syntax [SOURce:]POWer:RANGe <NRf+>

Parameters 0 to max. rated power.

Unit W (power)

Examples POW:RANG 15 SOUR:POW:RANGE MIN

Query Syntax [SOURce:]POWer:RANGe?

Returned Parameters <NR3>

9.41 [SOURce:]POWer:HIGH [SOURce:]POWer:LOW

Channel Specific

Description This command sets the voltage high and low limit determined when in constant power mode.

Command Syntax [SOURce:]POWer:HIGH <NRf+> [SOURce:]POWer:LOW <NRf+>

Parameters 0 to max. rated voltage

Unit V (volts)

Examples POW:HIGH 5

Query Syntax [SOURce:]POWer:HIGH? [SOURce:]POWer:LOW?

Returned Parameters <NR3>

9.42 [SOURce:]POWer:PROTection

Channel Specific

Description This command sets the soft power protection level. If the input power exceeds the soft power protection level for the time specified by POW:PROT:DEL, the input is turned off.

Note:

Use POW:PROT:DEL to prevent momentary power limit conditions caused by programmed changes from tripping the overpower protection.

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

Parameters 0 to max. rated power

Unit W (power)

***RST Value** MAXimum

Examples POW:PROT 100

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

Returned Parameters <NR3>

Related Commands POW:PROT:DEL

9.43 [SOURce:]POWer:PROTection:DELay

Channel Specific

Description This command specifies the time that the input power can exceed the protection level before the input is turned off.

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

Parameters 0 to 60 seconds

Unit seconds

***RST Value** 0

Examples POW:PROT:DEL 5

Query Syntax [SOURce:]POWer:PROTection:DELay?

Returned Parameters <NR1>

Related Commands POW:PROT

9.44 [SOURce:]POWer:CONF

Channel Specific

Description This command sets the hard power protection level.

Command Syntax [SOURce:]POWer:CONFig[:LEVel] <NRf+>

Parameters 0 through max. Rated power

Unit W (power)

Examples POW:CONFig 100

Query Syntax [SOURce:]POWer:CONFig[:LEVel

Returned Parameters <NR3>

Related Commands POW:PROT

9.45 [SOURce:]IMPedance

Channel Specific

Description This command sets the resistance value when in CZ mode.

Command Syntax [SOURce:]IMPedance[:LEVel][:IMMediate] <NRf+>

Parameters 0 to 7500 ohms.

Unit R (ohms)

***RST Value** MAXimum

Examples IMP 5 IMP:LEV 3.5

Query Syntax [SOURce:]IMPedance[:LEVel][:IMMediate]?

Returned Parameters <NR3>

Related Commands IMP:RANG

9.46 [SOURce:]IMPedance:RANGe

Channel Specific

Description This command is used to set the resistance range when the load is in CZ mode. This limits the settable resistance.

Command Syntax [SOURce:]IMPedance:RANGe <NRf+>

Parameters 0 to 7500 ohms.

Unit R (ohms)

Examples IMP:RANG 15 SOUR:IMP:RANGE MIN

Query Syntax [SOURce:]IMPedance:RANGe?

Returned Parameters <NR3>

9.47 [SOURce:]IMPedance:RESistance

Channel Specific

Description This command is used to set series resistance when the load is in CZ mode.

Command Syntax [SOURce:]IMPedance:RESistance <NRf+>

Parameters 0 through 7500 ohms

Unit R (ohms)

***RST Value** 0

Examples IMP:RES 5

Query Syntax [SOURce:]IMPedance:RESistance?

Returned Parameters <NR3>

9.48 [SOURce:]IMPedance:INDuction

Channel Specific

Description This command is used to set the series inductance when the load is in CZ mode.

Command Syntax [SOURce:]IMPedance:INDuction <NRf+>

Parameters 0 through 65000uH

Unit uH

***RST Value** 0

Examples IMP:IND 5

Query Syntax [SOURce:]IMPedance:INDuction?

Returned Parameters <NR3>

9.49 [SOURce:]IMPedance:CAPacitance

Channel Specific

Description This command is used to set the parallel capacitance value when the load is in CZ mode.

Command Syntax [SOURce:]IMPedance:CAPacitance <NRf+>

Parameters 20 through 65000uF

Unit uF

Examples IMP:CAP 5

Query Syntax [SOURce:]IMPedance:CAPacitance?

Returned Parameters <NR3>

9.50 [SOURce:]IMPedance:HIGH

[SOURce:]IMPedance:LOW

Description This command is used to set the high and low voltage determined when the load is in CZ mode.

Command Syntax [SOURce:]IMPedance:HIGH <NRf+> [SOURce:]IMPedance:LOW <NRf+>

Parameters 0 to max. rated voltage

Unit V (volts)

Examples IMP:HIGH 5

Query Syntax [SOURce:]IMPedance:HIGH? [SOURce:]IMPedance:LOW?

Returned Parameters <NR3>

List Subsystem

List commands let you program complex sequences of input changes with rapid, precise timing, and synchronized with trigger signals. Each function for which lists can be generated has a list of values that specify the input at each list step.

10.1 [SOURce:]LIST:RANGe

Description This command sets the current range for list mode.

Command Syntax [SOURce:]LIST:RANGe <NRf>

Parameters MIN through MAX

Unit None

Examples LIST:RANGE 30

Query Syntax [SOURce:]LIST:RANGe?

Returned Parameters <NR3>

Related Commandsd LIST:LEV

10.2 [SOURce:]LIST:COUNt

Description This command sets the number of times that the list is executed before it is completed. The command accepts parameters in the range 1 through 65536, but 65536 is interpreted as infinity.

Command Syntax [SOURce:]LIST:COUNt <NRf+>

Parameters 1 to 65536 | MINimum | MAXimum

Examples LIST:COUN 3

Query Syntax [SOURce:]LIST:COUNt? [MINimum | MAXimum]

Returned Parameters <NR3>

Related Commandsd LIST:STEP

10.3 [SOURce:]LIST:STEP

Description This command sets the steps of the list.

Command Syntax [SOURce:]LIST:STEP <NRf+>

Parameters 2 to 84 | MINimum | MAXimum

Examples LIST:STEP 5

Query Syntax [SOURce:]LIST:STEP? [MINimum | MAXimum]

Returned Parameters <NR3>

Related Commandsd LIST:LEV

10.4 [SOURce:]LIST:LEVel?

Description This command specifies the setting for each list step.

Command Syntax [SOURce:]LIST:LEVel <NR1>, <NRf>

Parameters 1 to steps, MIN to MAX

Unit NONE, NONE

Examples LIST:LEV 1, 10LIST:LEV 2, 15.2

Query Syntax [SOURce:]LIST:LEVel? <NR1>

Returned Parameters <NR3>

Related Commandsd LIST:RANG

10.5 [SOURce:]LIST:SLEW

Description This command sets the slew rate for each step. This command programs both positive and negative going slew rates. MAXimum sets the slew to its fastest possible rate. MINimum sets the slew to its slowest rate. LIST:SLEW? returns the number of points programmed.

Command Syntax [SOURce:]LIST:SLEW[:BOTH] <NR1> ,<NRf>

Parameters 1 to steps, MIN to MAX

Unit NONE, NONE

Examples LIST:SLEW 1, 1.5

LIST:SLEW 2, MAX

Query Syntax [SOURce:]LIST:SLEW[:BOTH]? <NR1>

Returned Parameters <NR3>

Related Commandsd CURR:SLEW VOLT:SLEW RES:SLEW

10.6 [SOURce:]LIST:WIDth

Description This command sets the sequence of list dwell times. Each value represents the time in seconds that the input will remain at the particular list step point before completing the step. If times exceed 16383S, the input remains at the present level until a trigger sequences the next point in the list. Else At the end of the dwell time, the input automatically changes to the next point in the list.

Command Syntax [SOURce:]LIST:WIDth <NR1>, <NRf>

Parameters 1 to steps, 20uS to max

Unit NONE, s (seconds)

Examples LIST:WID 1, 0.02 LIST:WID 2, 0.5

Query Syntax [SOURce:]LIST:WIDth? <NR1>

Returned Parameters <NR3>

10.7 L

IST:SAV][SOURce:]LIST:SAV

Description This command stores the present list file of the electronic load to a specified location in memory. Up to 5 files can be stored. file in saved in locations 1-5 are volatile, the data are nonvolatile, the data will be saved when power is removed.

Command Syntax [SOURce:]LIST:SAV <NR1>

Parameters 1 to 7

Example LIST:SAV 3

Related Commandsd [SOURce:]LIST:RCL

10.8 [SOURce:]LIST:RCL

Description This command restores a list file that was previously stored in memory with a LIST:SAV command to the specified location.

Command Syntax [SOURce:]LIST:RCL <NR1>

Parameters 1 to 7

Example LIST:RCL 3

Related Commandsd [SOURce:]LIST:SAV

Measurement Subsystem

The signal-oriented measurement commands are used to acquire readings. You can use these high-level instructions to control the measurement process.

Making Measurements

The electronic load has the ability to make several types of voltage or current measurements. The measurement capabilities of the electronic load are particularly useful with applications that draw current in pulses.

All measurements are performed by digitizing the instantaneous input voltage or current for a defined number of samples and sample interval, storing the results in a buffer, and then calculating the measured result. Many parameters of the measurement are programmable. These include the number of samples, the time interval between samples, and the method of triggering. Note that there is a tradeoff between these parameters and the speed, accuracy, and stability of the measurement in the presence of noise.

There are two ways to make measurements:

1. Use the MEASure commands to immediately start acquiring new voltage or current data, and return measurement calculations from this data as soon as the buffer is full. This is the easiest way to make measurements, since it requires no explicit trigger programming.

2. Use an acquisition trigger to acquire the data. Then use the FETCh commands to return calculations from the data that was retrieved by the acquisition trigger. This method gives you the flexibility to synchronize the data acquisition with a trigger. FETCh commands do not trigger the acquisition of new measurement data, but they can be used to return many different calculations from the data that was retrieved by the acquisition trigger.
   Making triggered measurements with the acquisition trigger system is discussed in the User Manual.

Note:

For each MEASure form of the query, there is a corresponding query that begins with the header FETCh. FETCh queries perform the same calculation as their MEASure counterparts, but do not cause new data to be acquired. Data acquired by an explicit trigger or a previously programmed MEASure command are used.

11.1 :FETCh:VOLTage? :FETCh:CURRent? :FETCh:POWer[:DC]?

Channel Specific

Description These query commands request the latest post-processed readings stored in the sample buffer. After sending this command and addressing the electronic load to talk, the readings are sent to the computer. This command does not affect the instrument setup. This command does not trigger source-measure operations; it simply requests the last available readings. Note that this command can repeatedly return the same readings. Until there are new readings, this command continues to return the old readings.

For example, assume that the electronic load performed 20 measure operations. The :FETCh:VOLTage? command will request the readings for those 20 measure operations. If :FETCh:VOLTage? is sent while performing measure operations (ARM annunciator on), it will not be executed until the electronic load goes back to the idle state.

11.2 :MEASure:VOLTage[:DC]?:MEASure:CURRent[:DC]?

Channel Specific

Description These commands combine other signal-oriented measurement commands to perform a "one-shot" measurement and acquire the reading. Note that if a function is not specified, the measurement will be done on the function that is presently selected.

11.3 :MEASure:ALLVoltage? :MEASure:ALLCurrent?

Description These commands combine the voltage or current measured at the input of all electronic load modules.

11.4 :FETch :ALLVoltage?:FETch:ALLCurrent?

Description These commands combine the real-time voltage or current measured at the input of all electronic load modules.

Sense Subsystem

Sense Commands

The Sense subsystem is used to configure and control the measurement functions of the electronic load. A function does not have to be selected before you program its various configurations. A function can be selected any time after it has been programmed. Whenever a programmed function is selected, it assumes the programmed states.

12.1 SENSe:AVERage:COUNt

Channel Specific

Description The command is used to specify the filter count. In general, the filter count is the number of readings that are acquired and stored in the filter buffer for the averaging calculation. The larger the filter count, the more filtering that is performed.

Command Syntax SENSe:AVERage:COUNt <NRf+>

Parameters 2 to 16

***RST Value** 8

Examples SENS:AVER:COUN 10

Query Syntax SENSe:AVERage:COUNt?

Returned Parameters <NR1>

Related Commands SENS:AVER

12.2 SENSe:VOLTage[:DC]:RANGe:AUTO

Channel Specific

Description This command is used to control auto ranging for voltage. With auto ranging enabled, the instrument automatically goes to the most sensitive range to perform the measurement.

The auto range command (:RANGe:AUTO) is coupled to the command that manually selects the measurement range (:RANGe <n>). When auto range is enabled, the parameter value for :RANGe <n> changes to the automatically selected range value. Thus, when auto range is disabled, the instrument remains at the automatically selected range. When a valid:RANGe <n> command is sent, auto ranging disables.

Command Syntax SENSe:VOLTage[:DC]:RANGe:AUTO <bool>

Parameters 0 | 1 | OFF | ON

***RST Value** ON

Examples SENS:VOLT:RANG:AUTO 1

Query Syntax SENSe:VOLTage[: DC]:RANGe:AUTO?

Returned Parameters 0 | 1

Related Commands SENS:VOLT:RANG

Programming Subsystem

The program subsystem is used to control the parameters of the built-in automatic test function.

13.1 :PROGram:ACTive:CHANnel

The command is used to specify the active channels for the automatic test function. When a channel is active, some parameters for the program setup and selec the channel for the test.

Command Syntax PROGram:ACTive:CHANnel <NRf+>

Parameters 0 – 65535 The value, in decimal, is based on the binary representation of the active channels which are represented by the bits, according to the following table. Note: Channel 18-11 represents channels 8 to 1 for MDL4U002 extended mainframe.
1 – channel is active.
0 – channel is not active.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Channel	18	17	16	15	14	13	12	11	8	7	6	5	4	3	2	1

Table 13.1 Active Channel

Examples PROG:ACT:CHAN 683
683 = 1000100
Sets channel 3 and 7 active.

Query Syntax PROG:ACT:CHAN?

Returned Parameters <NR1>

13.2 :PROGram:ACTive:SEQuence

The command is used to specify the sequence to enable for the program. The sequence number is linked to the instrument settings saved in the internal storage memory. Each sequence number in each programs are linked to one of the instrument settings storage locations. Sequence 1 to 10 of program 1 is linked (in order) to storage locations 1-10.
Sequence 1 to 10 of program 2 is linked (in order) to storage locations 11-20.
….
Sequence 1 to 10 of program 10 is linked (in order) to storage locations 91-100.

Command Syntax PROGram:ACTive:SEQuence <NRf+>

Parameters 0 – 512
The value, in decimal, is based on the binary representation of the sequence number which are represented by the bits, according to the following table.
1 – Selects that sequence number.
0 – Not selected.

Bit Value in decimal	512	256	128	64	32	16	8	4	2	1
Bit	9	8	7	6	5	4	3	2	1	0
Sequence	10	9	8	7	6	5	4	3	2	1

Table 13.2 Active Sequence

Examples PROG:ACT:SEQ 384
384 = 110000000
Enables sequence 8 and 9.

Query Syntax PROG:ACT:SEQ?

Returned Parameters <NR1>

13.3 :PROGram:SEQuence:SHORt

The command is used to specify the short sequence for the program. When a sequence is selected for short, all channels for that sequence will be shorted for testing.

Command Syntax PROGram:SEQuence:SHORt <n>, <NRf+>

Parameters <n> - Sequence number, 1 to 10.
<NRf+> - 0 – 65535
The value, in decimal, is based on the binary representation of the channels to which are represented by the bits, according to the following table. Note: Channel 18-11 represents channels 8 to 1 for MDL4U002 mainframe extension.
1 – channel will be shorted.
0 – channel will not be shorted.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Channel	18	17	16	15	14	13	12	11	8	7	6	5	4	3	2	1

Table 13.3 Sequence Short

Examples PROG:SEQ:SHOR 683
683 = 1000100
Sets channel 3 and 7 for short testing.

Query Syntax PROG:SEQ:SHOR? <n>

Returned Parameters <NRf+>

13.4 :PROGram:SEQuence:ONTime

The command is used to specify the input ON time for the selected sequence, which controls how long to enable the inputs for the selected sequence.

Command Syntax PROGram:SEQuence:ONTime <n>, <NRf>

Parameters <n> - Sequence number, 1 to 10. <NRf> - On time, 0.01 to 60.0 seconds.

Examples PROG:SEQ:ONT 4, 5
Sets sequence 4 ON time to 5 seconds.

Query Syntax PROG:SEQ:ONT? <n>

Returned Parameters <NR1>

13.5 :PROGram:SEQuence:OFFTime

The command is used to specify the input OFF time for the selected sequence, which controls how long to disable the inputs for the selected sequence.

Command Syntax PROGram:SEQuence:OFFTime <n>, <NRf>

Parameters <n> - Sequence number, 1 to 10.
<NRf> - Off time, 0.01 to 60.0 seconds.

Examples PROG:SEQ:OFFT 4, 5 Sets sequence 4 OFF time to 5 seconds.

Query Syntax PROG:SEQ:OFFT? <n>

Returned Parameters <NR1>

13.6 :PROGram:SEQuence:PFDTime

The command is used to specify the pass/fail delay time for the selected sequence, which controls the delay allowed before checking if the settings passed or failed for a pass/fail test.

Command Syntax PROGram:SEQuence:PFDTime <n>, <NRf>

Parameters <n> - Sequence number, 1 to 10.
<NRf> - Pass/Fail Delay, 0.01 to 60.0 seconds.

Examples PROG:SEQ:PFDT 4, 5
Sets sequence 4 with pass/fail delay time to 5 seconds.

Query Syntax PROG:SEQ:PFDT? <n>

Returned Parameters <NR1>

13.7 :PROGram:SEQuence:PAUSe

The command is used to specify the pause sequence for the program. The selected sequence will be paused during the program. The program will remain paused until it is unpaused.

Command Syntax PROGram:SEQuence:PAUSe <NRf+>

Parameters 0 – 512
The value, in decimal, is based on the binary representation of the sequence number which are represented by the bits, according to the following table.
1 – Selects that sequence number for pausing.
0 – Not selected.

Bit Value in decimal	512	256	128	64	32	16	8	4	2	1
Bit	9	8	7	6	5	4	3	2	1	0
Sequence	10	9	8	7	6	5	4	3	2	1

Table 13.4 Sequence Pause

Examples PROG:SEQ:PAUS 384
384 = 110000000
Sequence 8 and 9 will be paused during the test.

Query Syntax PROG:SEQ:PAUS?

Returned Parameters <NR1>

13.8 :PROGram:STOP:CONDition

The command is used to specify the stop condition of the program. Users can select either the stop after completion of the program or stop when a failure occurs in the test.

Command Syntax PROGram:STOP:CONDition <COMPletion | FAILure>

Parameters COMPlete
FAILure

Examples PROG:STOP:COND COMP
Sets the program to stop on completion.

Query Syntax PROG:STOP:COND?

Returned Parameters COMPletion, FAILure

13.9 :PROGram:CHAin

The command is used to specify the program chain parameter. This specifies the next program to run automatic testing after the current program finishes. If set to 0, no further programs will be tested and test will end.

Command Syntax PROGram:CHAin <n>

Parameters <n> - Program number, 1 to 10.
Set to 0 if no programs will be tested after completion of the current program.

Examples PROG:CHA 3
Configures program 3 to be run following after the current program finishes.

Query Syntax PROG:CHA? <n>

Returned Parameters <n>

13.10 :PROGram:SAVe

The command is used to save the configurations to a program location in memory.

Command Syntax PROGram:SAVe <n>

Parameters <n> - Program number, 1 to 10.

Examples PROG:SAV 2
Saves the configurations to program 2 location.

13.11 :PROGram:RCL

The command is used to recall a specified program from memory.

Command Syntax PROGram:RCL <n>

Parameters <n> - Program number, 1 to 10.

Examples PROG:RCL 2
Recall program 2 from memory.

Other Commands

14.1 FAC:MAC?

Description This query command returns the MAC address for the Ethernet interface.

Command Syntax FAC:MAC?

Return String XX.XX.XX.XX.XX.XX
Where XX are alphanumeric characters.

SCPI Command Tree

Commands	Description	Default	Channel
:TRACe			
:CLEar	Clear readings from buffer.		Y
:FREE?	Query points available and points in use.		Y
:POINts <n>	Specify size of buffer (2 to 1024).	2	Y
:POINts?	Query buffer size.		Y
:FEED <name>	Select source of readings (VOLTage, CURRent, TWO).	Two	
:CONTrol <name>	Select buffer control mode (NEVer or NEXT).	Never	Y
:CONTrol?	Query buffer control mode.		Y
:FEED?	Query source of readings for buffer.		Y
:FILTer			
[:STATe]	Select the type of buffer data	OFF	
[:STATe]?	Query the type of buffer data		
:DELay <n>	Set the trigger delay time	0	
:DELay?	Query the trigger delay time		
:TIMer <n>	Set the timer interval	1	
:TIMer?	Query the timer interval		
:DATA?	Read all data in the buffer.		

Table 15.1 Trace Command Summary

Commands	Description	Default	Channel
:CHANnel <n>	Select the electronic load channel.	1	
:CHANel:ID?	This query reads the model numbers of the modules installed in the mainframe.		Y

Table 15.2 CHANnel Command Summary

Commands	Description	Default	Channel
[:SOURce]			
:INPut			
[:STATe] <b>	Set selected channel's input state.	OFF	Y
[:STATe]?	Query selected channel's input state.		Y
:ALL <b>	Set all channels' input states.		N
:SHORt <b>	Set selected channel's load short state.	OFF	Y
:SHORt?	Query selected channel's load short state.		Y
:SYNCon <b>	Set the ON/OFF state for trace mode.		Y
:SYNCon?	Query the ON/OFF state for trace mode.		Y
:TIMer			
[:STATe]	Set the timer ON/OFF state.		Y
[:STATe]?	Query the timer ON/OFF state.		Y
:DELay	Set the delay time for timer.		Y
:DELay?	Query the delay time for timer.		Y
:REMote			
:SENSe	Set the remote sense ON/OFF state.		Y
:SENSe?	Query the remote sense ON/OFF state.		Y
:FUNCtion <name>	Set electronic load's regulation mode (VOLTage, CURRent, RESistance, POWer, IMPEDANCE.	CURRent	Y
:MODE <name>	Set electronic load's input mode (FIXed or LIST).	FIXed	Y
:MODE?	Query electronic load's input mode.		Y
:FUNCtion?	Query electronic load's regulation mode.		Y
:TRANsient			
[:STATe] <b>	Enable or disable transition mode.	OFF	Y
[:STATe]?	Query electronic load's transition mode.		Y
:CURRent			
[:LEVel]			
:RANGe <NRf>	Set constant current range.	MAX	Y
:RANGe?	Query constant current range.		Y
:SLEW			Y
[:BOTH] <NRf>	Set both positive and negative slew rate for current	MAX	Y
:POSitive <NRf>	Set positive slew rate for current	MAX	Y
:POSitive?	Query positive slew rate for current		Y
:NEGative <NRf>	Set negative slew rate for current		Y
:NEGative?	Query negative slew rate for current		Y
:PROTection			
[:STATe] <b>	Disable or enable current protection	ON	
[:STATe]?	Query current protection state		
:LEVel <NRf>	Set current protection level	MAX	
:LEVel?	Query current protection level		
:DELay <NRf>	Set current protection delay time	3	
:DELay?	Query current protection delay time		

Table 15.3 SOURce Command Summary

Commands	Description	Default	Channel
:TRANsient			
:MODE <name>	Set current transiton mode (CONTinuous, PULSe, or TOGGle)	CON-	
:MODE?	Query current transition mode	Tinuous	
:ALEVel <NRf>	Set current transition level A	MAX	
:ALEVel?	Query current transition level A		
:AWIDth <NRf>	Set current transition width A	500 uS	
:AWIDth?	Query current transiton width A		
:BLEVel <NRf>	Set current transition level B	MIN	
:BLEVel?	Query current transition level B		
:BWIDth <NRf>	Set current transition width B	500 uS	
:BWIDth?	Query current transiton width B		
:HIGH <NRf>	Set the voltage specification of upper limit		
:HIGH?	Query the voltage specification of upper limit		
:LOW <NRf>	Set the voltage specification of lower limit		
:LOW?	Query the voltage specification of lower limit		
:VOLTage			
[:LEVel]			
:ON <Nrf>	Set VON level	MAX	
:ON?	Query VON level		
:RANGe <NRf>	Set constant voltage range	MIN	
:RANGe?	Query constant voltage range		
:AUTO	Set the auto range mode for the voltmeter	MAX	
:AUTO?	Query the auto range mode for the voltmeter		
:LATCh <b>	Enable or disable Von in latch mode	ON	
:LATCh?	Query Von latch mode		
:HIGH <NRf>	Set the current specification of upper limit	ON	
:HIGH?	Query the current specification of upper limit		
:LOW <NRf>	Set the current specification of lower limit		
:LOW?	Query the current specification of lower limit		
:RESistance			
[:LEVel]			
:RANGe <NRf>	Set constant resistance range	MAX	
:RANGe?	Query constant resistance range		
:HIGH <NRf>	Set the voltage specification of upper limit	MAX	
:HIGH?	Query the voltage specification of upper limit		
:LOW <NRf>	Set the voltage specification of lower limit		
:LOW?	Query the voltage specification of lower limit		
:POWer			
[:LEVel]			
:RANGe <NRf>	Set power protection delay time	3	
:RANGe?	Query power protection delay time		
:PROTection			
:LEVel <NRf>	Set hardware power protection level		
:LEVel?	Query hardware power protection level		
:DELay <NRf>	Set the delay time after power protection.		
:DELay?	Query the delay time after power protection.		
:CONFig	Set the hard power protection level.		
:CONFig?	Query the hard power protection level.		

Table 15.4 SOURce Command Summary Continued

Commands	Description	Default	Channel
:IMPedance			
[:LEVel]			
[:IMMediate] <NRf>	Set constant IMPedance level		
[:IMMediate]?	Query constant IMPedance level		
:RANGe <NRf>	Set constant resistance range		
:RANGe?	Query constant resistance range		
:RESistance	Set constant resistance level		
:RESistance	Query constant resistance level		
:CAPacitance	Set constant capacitance level		
:CAPacitance?	Query constant capacitance level		
:INDuction	Set constant induction level		
:INDuction?	Query constant induction level		
:HIGH <NRf>	Set the voltage specification of upper limit		
:HIGH?	Query the voltage specification of upper limit		
:LOW <NRf>	Set the voltage specification of lower limit		
:LOW?	Query the voltage specification of lower limit		
:LIST			
:RANGe <NRf>	Set list range		
:RANGe?	Query list range		
:COUNt <n>	Set list cycle (1 to 65536)		
:COUNt?	Query list cycle		
:STEP <n>	Set total list step (2 to 84)		
:STEP?	Query list step		
:LEVel <NRf>,<NRf>	Set list step level		
:LEVel? <NRf>	Query list step level		
:SLEW			
[:BOTH] <NRf>	Set list step slew rate		
[:BOTH]?	Query list step slew rate		
:WIDth <NRf>,<NRf>	Set list step width		
:WIDth <NRf>	Query list step width		
:SAV <name>	Save list file to specify filename (1 to 5)		
:RCL <name>	Recall list file		
:PROTection:CLEar	Clear protection		

Table 15.5 SOURce Command Summary Continued

Commands	Description	Default	Channel
:SENSe			
:AVERage			
:COUNt <n>	Set filter count (1 to 100).	8	Y
:COUNt?	Query filter count.		Y

Table 15.6 SENSe Command Summary

Commands	Description	Default	Channel
:MEASure			
:VOLTage			
:CURRent			
:FETch			
:VOLTage			
:CURRent			
:POWer?	Fetch the last measured power		Y

Table 15.7 MEASure, FETch Command Summary

Commands	Description	Default	Channel
:TRIGger	Path to program trigger layer.		
[:IMMediate]	Generates a trigger signal.		N
:TIMer <NRf>	Set timer interval (0.01 to 999.99sec).	0.1	N
:TIMer?	Query the programmed timer interval.		N

Table 15.8 TRIGger Command Summary

Commands	Description	Default	Channel
:SYSTem			
:PRESet	Return to :SYST:PRESet defaults.		N
:POSetup <name>	Select power-on setup: (RST or SAV0).	RST	N
:POSetup?	Query power-on setup.		N
:VERsion?	Query rev level of SCPI standard.		N
:ERRor?	Query (read) Error Queue.		N
:CLEar	Clears messages from the Error Queue.		N
:LOCal	Take load out of remote mode and into local mode.		N
:REMote	Place load in remote control.		N
:RWLock	Lockout front panel controls.		N

Table 15.9 SYSTem Command Summary

Commands	Description	Default	Channel
:STATus		(note 1)	
:CHANnel	Path to control channel status registers.		Y
[:EVENt]?	Read the event register.	(note 2)	Y
:ENABle <n>	Program the enable register.	(note 3)	Y
:ENABle?	Read the enable register.		Y
:CONDition?	Read the condition register.		Y
:CSUMary	Path to control channel summary status.		
[:EVENt]?	Read the event register.	(note 2)	N
:ENABle <n>	Program the enable register.	(note 3)	N
:ENABle?	Read the enable register.		N
:OPERation	Path to control operation status register.		
[:EVENt]?	Read the event register.	(note 2)	N
:ENABle <n>	Program the enable register.	(note 3)	N
:ENABle?	Read the enable register.		N
:CONDition?	Read the condition register.		N
:QUEStionable	Path to control questionable status register.		
[:EVENt]?	Read the event register.	(note 2)	N
:ENABle <n>	Program the enable register.	(note 3)	N
:ENABle?	Read the enable register.		N
:CONDition?	Read the condition register.		N
:PRESet	Return status register to default states.		N

Table 15.10 STATus Command Summary

Commands	Description	Default	Channel
:PROGram			
:ACTive	Subsystem to ACTive		
:CHANnel <n>	Set the active channel for test		N
:CHANnel?	Query the active channel for test		N
:SEQuence <n>	Set the active sequence for test		N
:SEQuence?	Query the active channel for test		N
:SEQuence	Subsystem to SEQuence		
:SHORt <n>, <n>	Set the short channel for the selected seq.		N
:SHORt? <n>	Query the short channel for the selected seq.		N
:ONTime <n>,<NRf>	Set the on time for the selected seq.		N
:ONTime?<n>	Query the on times for the selected seq.		N
:OFFTime <n>,<NRf>	Set the off times for the selected seq.		N
:OFFTime? <n> Query	the off times for the selected seq.		N
:PFDTime <n>,<NRf>	Set the pass/fail delay time		N
:PFDTime? <n>	Query the pass/fail delay time		N
:PAUSe <n>	Set the pause sequence.		N
:PAUSe?	Query the pause sequence.		N
:STOP			
:CONDition <cond>	Set the stop condition for test.		N
:CONDition?	Query the stop condition for test.		N
:CHAin <n>	Set the chain file for this program file.		N
:CHAin?	Query the chain for this program file.		N
:SAVe <n>	Save this program file.		N
:RCL <n>	Recall this program file.		N

Table 15.11 PROGram Command Summary

Programming Examples

16.1 Introduction

This chapter contains examples on how to program your electronic load. Simple examples show you how to program:

- Input functions such as voltage, current, and resistance
- Measurement functions
- Status and protection functions
- Transient functions, including lists

Note:

The examples in this chapter show which commands are used to perform a particular function, but do not show the commands being used in any particular programming environment.

16.1.1 Power-on Initialization

When the electronic load is first turned on, it wakes up with the input state set OFF. The following commands are given implicitly at power-on:

- *RST or *RCL 0
- *CLS
- *SRE 0
- *ESE 0

*RST is a convenient way to program all parameters to a known state. Refer to the *RST command in Section **Common Commands** to see how each programmable parameter is set by *RST. Refer to the *PSC command in Section **Common Commands** for more information on the power-on initialization of the *ESE and the *SRE registers.

16.1.2 Enabling the Input

To enable the input, use the command:

```
INPut ON
```

16.1.3 Input Voltage

The input voltage is controlled with the VOLTage command. For example, to set the input voltage to 25 volts, use:

```
VOLTage 25
```

16.1.4 Input Current

All models have a programmable current function. The command to program the current is:

CURRent <n>

where <n> is the input current in amps.

16.1.5 Overcurrent Protection

The electronic load can also be programmed to turn off its input if the current protection level is reached.

As explained in Section **Subsystem Commands**, this protection feature is implemented by the following command:

CURRent:PROTection:STATe ON | OFF

Note:

Use CURRent:PROTection:DELay to prevent momentary current limit conditions caused by programmed input changes from tripping the overcurrent protection.

16.1.6 Generating Triggers

You can generate a single trigger by sending the following command over the GPIB:

TRIGger:IMMediate

Note that this command will always generate a trigger. Use the TRIGger:SOURce command to select other trigger sources such as the mainframe's external trigger input.

16.1.7 Programming Transients

Transient operation is used to synchronize input changes with internal or external trigger signals, and simulate loading conditions with precise control of timing, duration, and slew. The following transient modes can be generated:

Continuous	Generates a repetitive pulse stream that toggles between two load levels.
Pulse	Generates a load change that returns to its original state after some time period.
Toggled	Generates a repetitive pulse stream that toggles between two load levels. Similar to Continuous mode except that the transient points are controlled by explicit triggers instead of an internal transient generator.

Note:

Before turning on transient operation, set the desired mode of operation as well as all of the parameters associated with transient operation. At *RST all transient functions are set to OFF.

16.1.8 Continuous Transients

In continuous operation, a repetitive pulse train switches between two load levels. The rate at which the level changes is determined by the slew rate. In addition, use the following commands to program continuous transients:

```
CURRent:TRANsient:MODE CONTinuous
CURRent:TRANsient:ALEVel 5
CURRent:TRANsient:AWIDth 0.4 mS
CURRent:TRANsient:BLEVel 10
CURRent:TRANsient:BWIDth 0.6 mS
TRANsient ON
TRIGger:IMMediate
```

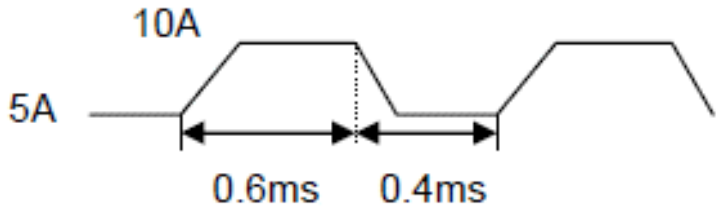

Figure 16.1 Continuous Transients

This example assumes that the CC mode is active and the slew rate is at the default setting (maximum rate). The electronic load module starts conduction at the main level (in this case 5 amps). When transient operation is turned on, the module's input current will slew to and remain at 10 amps for 60% of the period (600 us). The input current will then slew to and remain at 5 amps for the remaining 40% (400 us) of that cycle.

16.1.9 Pulse Transients

Pulsed transient operation generates a load change that returns to level B state after some time period. It is similar to continuous operation with the following exceptions:

a. To get a pulse, an explicit trigger is required. To specify the trigger source, use TRIGger:SOURce.

b. One pulse results from each trigger. Therefore, frequency cannot be programmed. Use the following commands to program pulsed transients:

```
CURRent:TRANsient:MODE PULSe
CURRent:TRANsient:ALEVel 5
CURRent:TRANsient:BLEVel 10
CURRent:TRANsient:BWIDth 10 mS
TRANsient ON
TRIGger:IMMediate
```

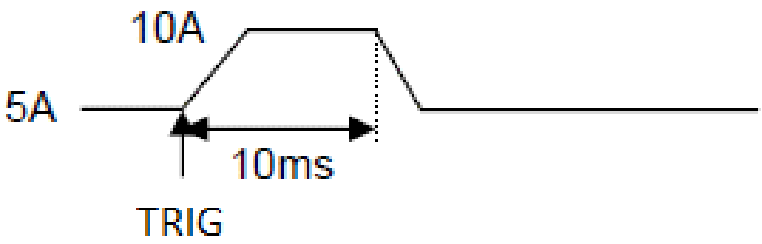

Figure 16.2 Pulse Transients

This example assumes that the CC mode is active, the slew rate is at the factory default setting (maximum rate), and a trigger signal is connected to the mainframe's external trigger input. The electronic load module starts conduction at the transient level A setting (5 amps). When the transient mode is turned on and an external trigger signal is received (or TRIGger:IMMediate is received), the input level starts increasing at a rate determined by the slew rate. When the value specified by the transient level setting (5 amps) is reached, it stays there for the remainder of the time determined by the pulse width setting (10 ms). After this time has elapsed, the input level decreases to the main level again at the rate specified by the slew setting and remains there until another trigger is received. Any triggers that occur during the time the transient level is in effect will re-trigger the pulse, extending the pulse by another pulse-width value.

16.1.10 Toggled Transients

Toggled transient operation causes the module input to alternate between two pre-defined levels as in continuous operation except that the transient transitions are controlled by explicit triggers instead of the internal transient generator. Use the following commands to program toggled transients:

```
TRIGger:SOURce EXTernal
CURRent:TRANsient:MODE TOGGle
CURRent:TRANsient:ALEVel 5
CURRent:TRANsition:BLEVel 10
TRANsient ON
```

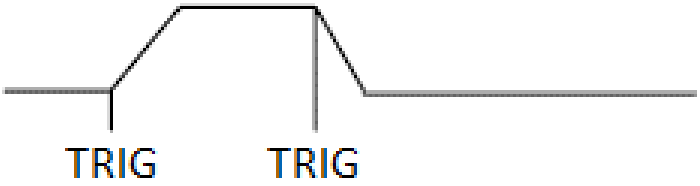

Figure 16.3 Toggled Transients

This example assumes that CC mode is active, the slew rate is at the factory default setting (maximum rate) and a trigger signal is connected to the mainframe's external trigger input. Toggled transient operation is similar to that described for continuous and pulse operation, except that each time a trigger is received the input alternates between the level A and level B current levels.

16.2 Programming Lists

List mode lets you generate complex sequences of input changes with rapid, precise timing, which may be synchronized with internal or external signals. This is useful when running test sequences with a minimum amount of programming overhead. You can program up to 84 steps in the list, the time interval that each setting is maintained, the number of times that the list will be executed, and how the settings change in response to triggers. All list data can be stored in nonvolatile memory using the LIST:SAV command. This means that the programmed data for any list will be retained when the electronic load is turned off. Use the LIST:RCL command to recall the saved state.

16.2.1 4-Step Current Change List Example

The following example procedure shows how to generate and save a simple 4-step list of current changes. When programming, be sure to verify all parameters are within the module's specifications.

1. Set the current range of the list.
   LIST:RANGe 40
2. Set the list cycle.
   LIST:COUNt 10000
3. Specify the number of steps in the list.
   LIST:STEP 4
4. Specify the current setting for step 1.
   LIST:LEVel 1,5
5. Set the current slew rate for step 1.
   LIST:SLEW 1,1
6. Set the width for step 1.
   LIST:WIDth 1,10ms
7. Specify the current setting for step 2.
   LIST:LEVel 2,10

8. Set the current slew rate for step 2.
   LIST:SLEW 2,1

9. Set the width for step 2.
   LIST:WIDth 2,10ms

10. Specify the current setting for step 3.
    LIST:LEVel 3,20

11. Set the current slew rate for step 3.
    LIST:SLEW 3,1

12. Set the width for step 3.
    LIST:WIDth 3,10ms

13. Specify the current setting for step 4.
    LIST:LEVel 4,15

14. Set the current slew rate for step 4.
    LIST:SLEW 4,1

15. Set the width for step 4.
    LIST:WIDth 4,10ms

16. Save the list to memory location 2.
    LIST:SAV 2

17. Set the input regulation mode to be controlled by values in a list.
    FUNCtion:MODE LIST

18. Set trigger source to BUS
    TRIG:SOUR BUS

19. Run the list file
    *TRG

Error Messages

17.1 Error Number List

This appendix gives the error numbers and descriptions that are returned by the electronic load. Error numbers are returned in two ways:

1. Error numbers are displayed on the front panel.

2. Error numbers and messages are read back with the SYSTem:ERRor? query. SYSTem:ERRor? returns the error number into a variable and returns two parameters, an NR1 and a string.

The following table lists the errors that are associated with SCPI syntax errors and interface problems. It also lists the device dependent errors. Information inside the brackets is not part of the standard error message, but is included for clarification. When errors occur, the Standard Event Status register records them in bit 2, 3, 4, or 5:

17.2 Command Errors

100 through 199 (sets Standard Event Status Register bit #5 CME)

Error	Error String [Description/Explanation/Examples]
101	DESIGN ERROR: Too many numeric suffices in Command Spec
110	No Input Command to parse
114	Numeric suffix is invalid value
116	Invalid value in numeric or channel list, e.g. out of range
117	Invalid number of dimensions in a channel list
120	Parameter of type Numeric Value overflowed its storage
130	Wrong units for parameter
140	Wrong type of parameter(s)
150	Wrong number of parameters
160	Unmatched quotation mark (single/double) in parameters
165	Unmatched bracket
170	Command keywords were not recognized
180	No entry in list to retrieve (number list or channel list)
190	Too many dimensions in entry to be returned in parameters
191	Too many char

Table 17.1 Command Errors

17.3 Execution Errors

–200 through –299 (sets Standard Event Status Register bit #4 EXE)

Error	Error String [Description/Explanation/Examples]
-200	Execution error [generic]
-221	Settings conflict [check current device state]
-222	Data out of range [e.g., too large for this device]
-223	Too much data [out of memory; block, string, or expression too long]
-224	Illegal parameter value [device-specific]
-225	Out of memory
-270	Macro error
-272	Macro execution error
-273	Illegal macro label
-276	Macro recursion error
-277	Macro redefinition not allowed

Table 17.2 Execution Errors

17.4 System Errors

–300 through –399 (sets Standard Event Status Register bit #3 DDE)

Error	Error String [Description/Explanation/Examples]
-310	System error [generic]
-350	Too many errors [errors beyond 9 lost due to queue overflow]
Query	Errors -400 through -499 (sets Standard Event Status Register bit #2)
-400	Query error [generic]
-410	Query INTERRUPTED [query followed by DAB or GET before response complete]
-420	Query UNTERMINATED [addressed to talk, incomplete programming message received]
-430	Query DEADLOCKED [too many queries in command string]
-440	Query UNTERMINATED [after indefinite response]

Table 17.3 System Errors

17.5 Self-test Errors

0 through 99 (sets Standard Event Status Register bit #3)

Error	Error String [Description/Explanation/Examples]
0	No error
1	Module Initialization Lost
2	Mainframe Initialization Lost
3	Module Calibration Lost
4	Non-volatile RAM STATE section checksum failed
5	Non-volatile RAM RST section checksum failed
10	RAM selftest
11	CVDAC selftest 1
12	CVDAC selftest 2
13	CCDAC selftest 1
14	CCDAC selftest 2
15	CRDAC selftest 1
16	CRDAC selftest 2
20	Input Downv
40	Flash write failed
41	Flash erase failed
80	Digital I/O selftest error

Table 17.4 Self-test Errors

17.6 Device-Dependent Errors

100 through 32767 (sets Standard Event Status Register bit #3)

Error	Error String [Description/Explanation/Examples]
213	RS-232 buffer overrun error
216	RS-232 receiver framing error
217	RS-232 receiver parity error
218	RS-232 receiver overrun error
220	Front panel UART overrun
221	Front panel UART framing
222	Front panel UART parity
223	Front panel buffer overrun
224	Front panel timeout
225	Front CRC check error
226	Front cmd Error
401	CAL switch prevents calibration
402	CAL password is incorrect
403	CAL not enabled
404	Computed readback cal constants are incorrect
405	Computed programming cal constants are incorrect
406	Incorrect sequence of calibration commands
407	CV or CC status is incorrect for this command
603	FETCH of data that was not acquired
604	Measurement overrange

Table 17.5 Device-Dependent Errors

22820 Savi Ranch Parkway
Yorba Linda, CA 92887
www.bkprecision.com

Version: August 5, 2021