

# Computer Interfacing Manual

## Table of Contents

<b>INTRODUCTION</b>	<b>3</b>
<b>THE RS232 INTERFACE</b>	<b>3</b>
CABLE CONNECTIONS	3
INTERFACE SETTINGS	3
<b>THE USB INTERFACE</b>	<b>4</b>
USING THE ARROYO INSTRUMENTS USB INTERFACE	4
LOADING THE SOFTWARE DRIVERS	4
CABLE CONNECTIONS AND INTERFACE SETTINGS	4
<b>ARROYO CONTROL</b>	<b>4</b>
<b>USING COMMANDS</b>	<b>5</b>
IEE-488.2 COMMANDS	5
DEVICE-INDEPENDENT COMMANDS	5
DEVICE-DEPENDENT COMMANDS	5
LASER COMMAND LIST	5
TEC COMMAND LIST	6
COMMAND SYNTAX AND CONCATENATION	6
COMMAND PATHS	7
NUMERIC SUBSTITUTIONS	7
<b>WORKING WITH RADIXES AND HEX FLOATS</b>	<b>7</b>
<b>WORKING WITH SCRIPTS</b>	<b>9</b>
<b>EVENT AND CONDITION REPORTING</b>	<b>10</b>
INSTRUMENT STATUS REPORTING STRUCTURE	11
<b>ALPHABETICAL LIST OF COMMANDS</b>	<b>12</b>
<b>ERROR MESSAGES</b>	<b>70</b>

## Introduction

The *Computer Interfacing Manual* provides a complete summary of all commands supported by the Arroyo Instruments line of products. While in remote control mode, the computer interface allows full operation of the instrument, plus advanced features only available via the computer interface.

You will find in reviewing the command set that it is largely compatible with both ILX and Newport laser diode drivers and temperature controllers. In fact, for many applications, you can use an Arroyo Instruments controller in place of a Newport or ILX instrument with little to no change in the commands used to control the instrument.

## The RS232 Interface

Some instruments provide a RS232 serial interface, a common, easy-to-use interface for controlling the instrument. With baud rates up to 115k baud, high-speed control and measurement is possible.

### Cable Connections

The RS232 interface is a male DB9 pinned identically to a standard PC RS232 port, so a female/female cable in NULL modem configuration (transmit and receive pins swapped) is required. The full pin-out is described below:

Pin	Description
2	Receive
3	Transmit
5	Ground
Shell	Earth ground

**RS232 Connector (DB-9 Male)**

Depending on if you are connecting to a DB9 or DB25 on the PC, follow the pin-to-pin assignments in the table below.

Instrument	PC DB9	PC DB25
2	3	2
3	2	3
5	5	7

**Instrument to PC pin assignment**

### Interface Settings

Baud Rate	The instrument will operate at 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200 baud rates. The baud rate can be set through the instrument's menu, with a factory default of 9600, and must match the setting on the PC
Parity	None
Data Bits	8
Stop Bits	1
Flow Control	None

Ensure to disable flow control. Failure to do so will prevent the instrument from sending data back to the PC.

## The USB Interface

Because of its speed, expandability, and commonality, the USB interface has become the interface of choice for newer PC-connected devices. Unlike GPIB, USB uses inexpensive cables and allows up to 127 devices to be connected to a single USB master.

### Using the Arroyo Instruments USB Interface

To keep complexity to a minimum, once you have installed the USB drivers, the instrument will appear as a virtual serial port that you can use just like a normal serial port. In this way, you can communicate to the instrument without requiring special software modifications to your existing applications.

### Loading the Software Drivers

Unlike the RS232 interface, using the USB interface requires software drivers (provided on the CD with your instrument or available over the Microsoft Windows Update service) to be loaded onto your PC. Follow the instructions in your instrument manual for installing the USB drivers.

Once a COM port has been assigned to an instrument, it will continue to use the same COM port unless it is reassigned through the Control Panel.

### Cable Connections and Interface Settings

The Arroyo controller uses a full-size USB Type B socket. You may use any USB 1.1 or 2.0 certified cable. The baud rate is fixed at 38,400.

Baud Rate	38400
Parity	None
Data Bits	8
Stop Bits	1
Flow Control	None

Ensure to disable flow control. Failure to do so will prevent the instrument from sending data back to the PC.

## Arroyo Control

ArroyoControl is a free software program that is included on the disc that comes with your instrument, and provides remote control of all Arroyo Instruments' controllers. If you are having problems communicating with the instrument, it can be a useful diagnostic step to install and use ArroyoControl to see if the program can successfully communicate with the instrument. If it does, then it demonstrates that the connection and drivers are properly installed.

## Using Commands

While the instrument does not have a GPIB interface, the command structure and parsing were developed around the IEEE-488.2 standard. Most IEEE-488.2 commands are supported, and the command/response format is consistent with the standard.

When sending commands to the instrument, you must terminate the command string with a carriage return (ASCII 13) or line feed (ASCII 10, or both). The instrument will not start processing a command, or go into remote mode until it has received a carriage return or feed command terminator.

Commands are grouped into two major categories: device-independent and device-dependent commands. Device-independent commands include all the IEEE-488.2 supported commands plus other commands that are universal across all instruments. Device-dependent commands are specific to a class of instruments, such as TEC commands for the temperature controller. Not all instruments support every command. Review the command for instrument-specific differences.

### IEE-488.2 Commands

The following device-independent commands are supported by most instruments:

*CLS	*IST?	*RCL	*TST?
*ESE	*PRE	*RST	*WAI
*ESE?	*PRE?	*SRE	
*ESR?	*PSC	*SRE?	
*IDN?	*PSC?	*STB?	

### Device-Independent Commands

The following device-independent commands are supported by most instruments:

BEEP	ERRSTR?	RADix?	TERM?
BEEP?	LOCAL	REMERR	TERMINAL
BRIGHT	MESsage	REMERR?	TERMINAL?
BRIGHT?	MESsage?	SCRIPT:GET	TIME?
DELAY	ONDELAY	SCRIPT:GO	TIMER?
EQUIPment?	ONDELAY?	SCRIPT:PUT	
ERRors?	RADix	TERM	

### Device-Dependent Commands

Device-dependent commands are specific to a class of instruments, such as the temperature controller.

Many of the device dependent commands have two commands that do the same thing. For example, LAS:I? and LAS:LDI? perform the same operation. The “I”, “IPD”, and “Ppd” forms of commands are considered obsolete but are included for compatibility reasons.

### Laser Command List

LASer:CABLER	LASer:DELAYIN?	LASer:ENABLE:OUTOFF
LASer:CABLER?	LASer:DELAYOUT	LASer:ENABLE:OUTOFF?
LASer:CALMD (CALPD)	LASer:DELAYOUT?	LASer:EVEnt?
LASer:CALMD? (CALPD?)	LASer:DISplay	LASer:F
LASer:COND?	LASer:DISplay?	LASer:F?
LASer:DEC	LASer:ENABLE:COND	LASer:INC
LASer:DC	LASer:ENABLE:COND?	LASer:INTCONT
LASer:DC?	LASer:ENABLE:EVEnt	LASer:INTCONT?
LASer:DELAYIN	LASer:ENABLE:EVEnt?	LASer:LDI (I)

LASer:LDI? (!?)	LASer:MODE:ICW	LASer:QCWCOUNT?
LASer:LDIRES	LASer:MODE:ILBW	LASer:RANGE
LASer:LDIRES?	LASer:MODE:I	LASer:RANGE?
LASer:LDIRESAVAIL?	LASer:MODE:IHBW	LASer:SET:LDI? (!?)
LASer:LDV	LASer:MODE:PULSE	LASer:SET:LDV?
LASer:LDV?	LASer:MODE:BURST	LASer:SET:MDI? (IPD?)
LASer:LIMit:LDI (!)	LASer:MODE:TRIG	LASer:SET:MDP? (Ppd?)
LASer:LIMit:LDI? (!?)	LASer:MODE:LDV	LASer:STB?
LASer:LIMit:LDV	LASer:MODE:MDI (IPD)	LASer:STEP
LASer:LIMit:LDV?	LASer:MODE:MDP (Ppd)	LASer:STEP?
LASer:LIMit:MDI (IPD)	LASer:OUTput	LASer:TOLerance
LASer:LIMit:MDI? (IPD?)	LASer:OUTput?	LASer:TOLerance?
LASer:LIMit:MDP (Ppd)	LASer:PDBias	LASer:VSENSE
LASer:LIMit:MDP? (Ppd?)	LASer:PDBias?	LASer:VSENSE?
LASer:MDI (IPD)	LASer:PW?	LASer:USERCAL:EDIT
LASer:MDI? (IPD?)	LASer:PWF	LASer:USERCAL:EDIT?
LASer:MDP (Ppd)	LASer:PWP	LASer:USERCAL:GET?
LASer:MDP? (Ppd?)	LASer:QCWHOLD	LASer:USERCAL:PUT
LASer:MODE?	LASer:QCWHOLD?	LASer:USERCAL:RESET
LASer:MODE:CW	LASer:QCWCOUNT	

### TEC Command List

TEC:AUTOTUNE	TEC:HEATCOOL	TEC:PID
TEC:AUTOTUNE?	TEC:HEATCOOL?	TEC:PID?
TEC:AUTOTUNESTATE?	TEC:INC	TEC:R
TEC:CABLER	TEC:ITE	TEC:R?
TEC:CABLER?	TEC:ITE?	TEC:SENsor
TEC:CABLETYPE?	TEC:INVERTITE	TEC:SENsor?
TEC:COND?	TEC:INVERTITE?	TEC:SET:ITE?
TEC:CONST	TEC:LIMit:ITE	TEC:SET:R?
TEC:CONST?	TEC:LIMit:ITE?	TEC:SET:T?
TEC:DEC	TEC:LIMit:RHI	TEC:STB?
TEC:DISplay	TEC:LIMit:RHI?	TEC:STEP
TEC:DISplay?	TEC:LIMit:RLO	TEC:STEP?
TEC:ENABLE:COND	TEC:LIMit:RLO?	TEC:T
TEC:ENABLE:COND?	TEC:LIMit:THI	TEC:T?
TEC:ENABLE:EVEnt	TEC:LIMit:THI?	TEC:TOLerance
TEC:ENABLE:EVEnt?	TEC:LIMit:TLO	TEC:TOLerance?
TEC:ENABLE:OUTOFF	TEC:LIMit:TLO?	TEC:USERCAL:EDIT
TEC:ENABLE:OUTOFF?	TEC:MODE?	TEC:USERCAL:EDIT?
TEC:EVEnt?	TEC:MODE:ITE	TEC:USERCAL:GET?
TEC:FAN	TEC:MODE:R	TEC:USERCAL:PUT
TEC:FAN?	TEC:MODE:T	TEC:USERCAL:RESET
TEC:GAIN	TEC:OUTput	TEC:V?
TEC:GAIN?	TEC:OUTput?	

### Command Syntax and Concatenation

All commands use a common syntax for constructing the command and parameter passing. Commands are case-insensitive, and some commands have optional characters, which are denoted as lower-case characters in the command definition. For example, LASer:EVEnt? can be shortened to LAS:EVE? This allows for command abbreviation and therefore reduced communication times.

The input buffer of the instrument allows for up to 128 characters to be received before a command overflow error occurs. If a command overflow does occur, the entire buffer is discarded and an E-102 (Message too long) error will occur.

If a command requires one or more parameters, place a space between the command and the first parameter, then use a comma to separate additional parameters. For example, the laser tolerance command might look like:

LASER:TOLERANCE 10,25

It is also possible to concatenate multiple commands together as a single communication to the instrument. By using a semicolon, multiple commands can be sent to the instrument at once, but command processing will not start until all commands and the command terminator have been received. For example, to set the laser set point to 10mA and turn the output on, the command would look like:

LASER:LDI 10; LASER:OUTPUT 1

### Command Paths

The command set is structured in a tree-like organization, where commands for a common device or function are grouped together. For example, all commands specific to the laser driver start with a "LASER:" prefix. This is called the command path.

When issuing multiple concatenated commands to the instrument, it "remembers" the command path of the previous command, and allows you to omit the common path from the second command. Using the example above where the set point was changed to 10mA and the output turned on, it could also be shortened by omitting the second "LASER:" portion because when the OUTPUT command is processed, the command parser "remembers" its position in the command tree and starts looking for the command at that level. The command could therefore be shortened to this:

LASER:LDI 10;OUTPUT 1

Removing the optional characters from the commands, you can further shorten the command to:

LAS:LDI 10;OUT 1

In the case where the same command exists at the last command level and in one or more levels above, you can start the command with a colon (':') to force the command parser to start at the root. For example, take the following command:

LASER:MODE:LDV;LDV 0.5

At first glance, it appears to set the laser into voltage control mode and then set the set point to 0.5V. However, after processing the first command, the parser is at the "LASER:MODE:" level in the command tree and the second "LDV" command will match the "LASER:MODE:LDV" command, basically changing modes twice. To force the parser to interpret the command the way we intended, compose the second command after the semicolon like this:

LASER:MODE:LDV; :LASER:LDV 0.5

### Numeric Substitutions

For command readability, you can also substitute alternate values for zero (0) and one (1). "0" can be replaced by "OFF", "NEW", or "FALSE". "1" can be replaced by "ON", "OLD", or "TRUE".

## Working with Radixes and Hex Floats

It is possible to change the base coding of decimal values to binary, octal, or hexadecimal (hex) by using the RADIX command. Depending on your programming interface, it may be easier to send or receive data in one of these alternate notations.

The RADIX command only affects data sent by the instrument to the PC; the instrument will always accept values in alternate base values.

Hex values are prefixed with a '#H', octal with a '#O', and binary with a '#B'. For example, the decimal value 47635 would be represented as following:

Binary:           #B1011101000010011  
Octal:            #O135023  
Hexadecimal:   #HBA13

In addition to the binary, octal, and hex, floating point numbers can be transmitted and received in a 'ASCII-hex' notation. Transmission of floating-point values over plain text interfaces (such as is used on Arroyo controllers) invariably leads to rounding of the floating point number simply because the plain text interface does not (typically) transmit the full resolution a floating point number is capable of. This does not normally affect operation, and commands to the Arroyo controllers can always include additional resolution to mitigate this.

However, in some applications, it is useful to have the *exact* floating point value, and the hex float mode enables this. When operating in hex float mode, floating point numbers are transmitted in IEEE 754 notation, starting with a '#E' prefix.

For example, the single-precision float point value 123.45 would be encoded as #E42F6E666. For double-precision values, the hex data would be twice the length, and 123.45 in double-precision hex data would be #E405EDCCCCCCCCCD.

While the instrument will always affect floating point numbers encoded in hex float, the HEXFLOAT command enabled the transmission of hex float values from the instrument to the PC for those values that are represented in floating point notation (such as measurements and set points). Non-floating point numbers will continue to transmit normally (or in alternate base if selected by the RADIX command).

See [http://en.wikipedia.org/wiki/IEEE\\_754-1985](http://en.wikipedia.org/wiki/IEEE_754-1985) for additional information on the IEEE 754 format. See <http://babbage.cs.qc.edu/IEEE-754/> for an online tool to convert to/from hexadecimal values.

## Working with Scripts

Script support was added with version 2 of the firmware. It is normally used to load scripts for execution by the instrument's function keys, but scripts can also be executed remotely.

A script is a series of commands, anything you can normally send to the instrument, but stored in memory for execution at a later time. A script can contain any valid command except:

- DELAY
- \*WAI
- SCRIPT:GO
- SCRIPT:PUT

A script can be a maximum of 200 characters and up to four scripts can be stored.

A script can contain more than one command, but there is a slight difference when concatenating commands as compared to normal communications: instead of using a semicolon (;) to separate commands, you must use a carat (^). For example, a script to change the set point to 50mA and turn the output on, stored in position 1, would be stored with the following command:

```
SCRIPT:PUT 1, LAS:I 50 ^ LAS:OUT ON
```

Notice that a carat separates the two commands, not a semicolon. You can then execute the script remotely with the SCRIPT:GO command:

```
SCRIPT:GO 1
```

If your instrument has function keys, you can also assign the script to the function key.

## Event and Condition Reporting

In order to allow for quick summary reporting of device status, the instruments contain a powerful status reporting structure, which can be configured to report status at several levels of the instrument's operation. Using the various enable registers, a summary of the instrument's operation is made into the status byte register, allowing the control application to query only the status byte register, and if no pertinent events or conditions are being reported, no other status checking is needed.

In some cases, condition and event registers may appear to report the same thing, but it is how the registers work that make them unique from each other. A condition register will report the status of the instrument at the moment in time when the condition register is queried. For fast conditions (such as laser over-voltage), it's not possible to detect the condition no matter how fast you query the instrument. This is where event registers come in: they "remember" their state until you read them. For example, if a laser goes into current limit, then out of current limit, its condition register will indicate current limit only while it is in that state. The event register, however, will set its current limit bit, and that bit will remain set until the event register is read. All event registers, once read, will automatically reset themselves to zero.

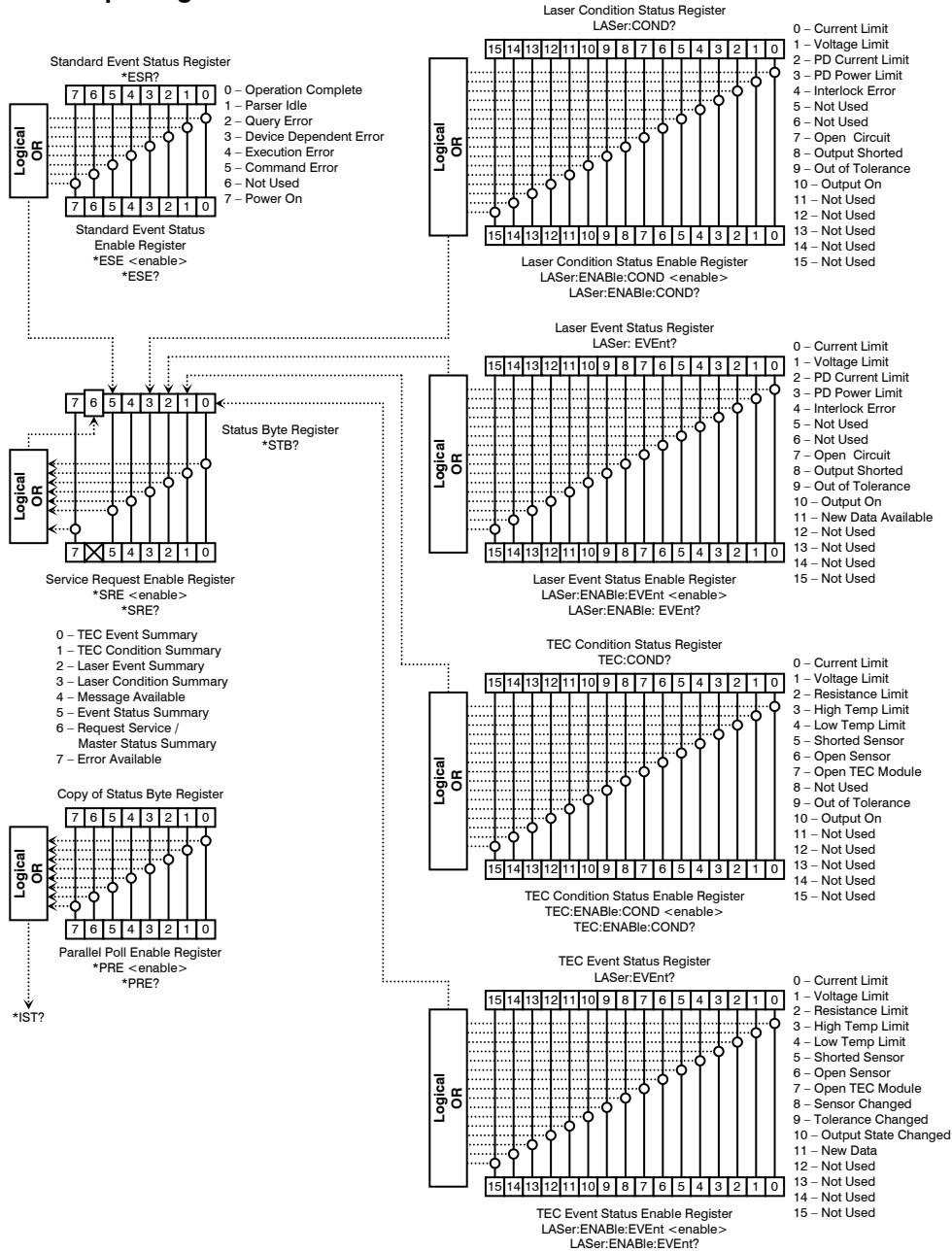
For example, if you wanted to monitor if the laser was in current or power limit, if the output turned off, or if there were any error in the error queue, you could enable reporting of current or power limit conditions in the LASer:COND register, and enable the output status change event in the LASer:EVEnt register. By querying the status byte register (\*STB?), no further action is necessary if the laser event, laser condition, and error available bits were all zero. If you did not use the status reporting system, you would need to issue at least three commands to do the same thing.

At first glance, the reporting structure pictured below can seem complex, but it is actually quite simple. For each register, there is a corresponding enable register. For example, the laser condition register (LASer:COND) has a laser condition enable register (LASer:ENABle:COND). The two registers are ANDed together, and if the resulting value is non-zero, then the corresponding bit in the status byte register is turned on (in this case, bit 3, the laser condition summary bit).

By reviewing your application needs and the reporting capabilities of the system, you can determine what conditions or event you want to enable and appropriately set each of the enable registers.

In USB and RS232 applications, the Service Request Enable Register, the Parallel Poll Enable Register, and the IST bit have little value because the Status Byte Register will contain more detail, but these registers have been implemented for greater compatibility with existing software applications.

### Instrument Status Reporting Structure



## Alphabetical List of Commands

### \*CLS

<b>Synopsis</b>	Clear status registers command
<b>Syntax</b>	*CLS
<b>Details</b>	Clears the standard event status register, all event registers, and the error queue.
<b>See Also</b>	ESR?, ERR?, *IST?, *STB?

### \*ESE

<b>Synopsis</b>	Set the Event Status Register Enable
<b>Syntax</b>	*ESE <i>mask</i>
<b>Details</b>	See the ESR? query for a definition of the individual bits with the <i>mask</i> parameter. The <i>mask</i> is logically ANDed with the ESR register, and if any of the resulting bits are high, bit 5 of the STB register (Event Status Summary) is set.
<b>See Also</b>	*ESE?, ESR?, *STB?

### \*ESE?

<b>Synopsis</b>	Query the Event Status Register Enable
<b>Syntax</b>	*ESE?
<b>Details</b>	Returns the value of the Event Status Register Enable. See the ESR? query for a definition of the individual bits with the response value.
<b>See Also</b>	*ESE, ESR?, *STB?

### \*ESR?

<b>Synopsis</b>	Query the Event Status Register
<b>Syntax</b>	*ESR?
<b>Details</b>	Returns the value of the Event Status Register. As with all event registers, after the value has been read, the register will be set to zero.

Response	Bit	Value	Description
<i>ESR</i>	0	1	Operation complete
	1	2	Parser idle: the parser buffer is empty
	2	4	Query error (errors 300 through 399)
	3	8	Device dependent error (errors 400 through 599)
	4	16	Execution error (errors 200 through 299)
	5	32	Command error (errors 100 through 199)
	6	64	Not Used
	7	128	Power on: the unit has been powered on since the last time this register was read or cleared.

Operation complete indicates when the controller has completed all pending operations and the outputs are stable.

**See Also** \*ESE, ESE?, \*STB?

### \*IDN?

**Synopsis** Query the instrument identification

**Syntax** \*IDN?

**Details** Returns the identification string for the instrument in the following format:

*Arroyo Model SN Ver Build*

Response	Description
<i>Model</i>	The model number of the product, such as '4205'.
<i>SN</i>	The serial number.
<i>Ver</i>	The firmware version.
<i>Build</i>	Internal build number.

### \*IST?

**Synopsis** Query Individual Status

**Syntax** \*IST?

**Details** Returns the value of the 'ist' bit within the reporting structure.

Response	Value	Description
<i>ist</i>	0	ist false
	1	ist true

**See Also** \*PRE, \*STB

### \*PRE

**Synopsis** Set the Parallel Poll Enable Register

**Syntax** \*PRE *mask*

**Details** See the STB? query for a definition of the individual bits with the *mask* parameter. The *mask* is logically ANDed with the Status Byte Register, and if any of the resulting bits are high, the 'ist' bit is set.

**See Also** \*PRE?, \*STB?

### \*PRE?

**Synopsis** Query the Parallel Poll Enable Register

**Syntax** \*PRE?

**Details** Returns the value of the Parallel Poll Enable register. See the STB? query for a definition of the individual bits with response value.

**See Also** \*PRE, \*STB?

### \*PSC

**Synopsis** Set the Power-On Status Clear

**Syntax** \*PSC *enable*

**Details** Enables or disables the power-on clearing of event and condition status enable registers.

Argument	Value	Description
<i>enable</i>	1	All event and condition status enable registers are cleared at power-up.
	0	All event and condition status enable registers are restored their last power-off state.

**See Also** \*PSC?

### \*PSC?

**Synopsis** Query the Power-n Status Clear

**Syntax** \*PSC?

**Details** Returns the value of the Power-On Status Clear register. See the \*PSC command for a definition of the response value.

**See Also** \*PSC

### \*RCL

**Synopsis** Recall command

**Syntax** \*RCL 0

**Details** The \*RCL command is used to restore the unit to saved configuration. However, only '\*RCL 0' is supported, and behaves identically to a \*RST command.

**See Also** \*RST

### \*RST

**Synopsis** Reset command

**Syntax** \*RST

**Details** Resets the instrument to factory defaults, and the output is shut off. The unit remains in remote mode.

**See Also** \*RCL

### \*SRE

**Synopsis** Set the Service Request Enable

**Syntax** \*SRE *mask*

**Details** The \*SRE command sets the Service Request Enable Register bits, which control the Master Status Summary bit in the Status Byte Register.

Argument	Bit	Value	Description
<i>mask</i>	0	1	TEC Event Summary
	1	2	TEC Condition Summary
	2	4	Laser Event Summary
	3	8	Laser Condition Summary
	4	16	Message Available
	5	32	Event Status Summary
	6	64	Request Service / Master Status Summary
	7	128	Error Message Available

**See Also** \*SRE?

### \*SRE?

**Synopsis** Query the Service Request Enable

**Syntax** \*SRE?

**Details** Returns the value of the Service Request Enable register. See the \*SRE command for a definition of the response value.

**See Also** \*SRE

### \*STB?

**Synopsis** Query the Status Byte Register

**Syntax** \*STB?

**Details** Returns the value of the Status Byte Register.

Response	Bit	Value	Description
<i>STB</i>	0	1	TEC Event Summary
	1	2	TEC Condition Summary
	2	4	Laser Event Summary
	3	8	Laser Condition Summary
	4	16	Message Available
	5	32	Event Status Summary
	6	64	Request Service / Master Status Summary
	7	128	Error Message Available

**See Also** \*SRE, \*PRE

### \*WAI

**Synopsis** Wait for Operation Complete command

**Syntax** \*WAI

**Details** The \*WAI command will pause command processing until the Operation Complete flag is true.

---

**BAUD****Synopsis** Sets the baud rate on RS232 interfaces**Syntax** BAUD *baudrate***Details** Changes the baud rate on RS232 interfaces. After sending the command, the baud rate will immediately be adjusted to the new speed, so if you are communicating over RS232, you must also change the PC-side baud rate to match.

Argument	Value	Description
<i>baudrate</i>	300	300 baud
	1200	1200 baud
	2400	2400 baud
	4800	4800 baud
	9600	9600 baud
	19200	19,200 baud
	38400	38,400 baud
	57600	57,600 baud
	115200	115,200 baud

---

**BEEP****Synopsis** Set the beep enable**Syntax** BEEP [*enable*]**Details** Causes the instrument to beep, or enables or disabled the beep sound for error messages and other events that generate an audible response. If *enable* is omitted, it is the equivalent of doing a 'BEEP 2'.

Argument	Value	Description
<i>enable</i>	0	Disable the beep sound
	1	Enable the beep sound
	2	Generate one beep

**See Also** BEEP?

---

**BEEP?****Synopsis** Query the beep enable**Syntax** BEEP?**Details** Returns the value of the Beep register.

Response	Value	Description
<i>enable</i>	0	Disable the beep sound
	1	Enable the beep sound

**See Also** BEEP

---

**BRIGHT****Synopsis** Set the display brightness

**Syntax**      BRIGHT *brightness*

**Details**      Sets the display brightness level, from 0 to 100 percent.

Argument	Value	Description
<i>brightness</i>	0 to 100	Brightness level of the display

On some displays, such as VFDs, there may only be 4 or 8 brightness levels, and the *brightness* parameter will be used to select an appropriate level.

**See Also**      BRIGHT?

## **BRIGHT?**

**Synopsis**      Query the display brightness

**Syntax**      BRIGHT?

**Details**      Returns the value of the display brightness. See the BRIGHT command for more details.

**See Also**      BRIGHT

## **CALdate?**

**Synopsis**      Query the calibration date

**Syntax**      CALdate?

**Details**      Returns the date string of the last calibration.

## **DELAY**

**Synopsis**      Causes a delay in command processing

**Syntax**      DELAY *time*

**Details**      Causes command processing to be delayed for the specified number of milliseconds.

Argument	Value	Description
<i>time</i>	1 to 30000	Delay, in milliseconds

## **EQUIPment?**

**Synopsis**      Query for installed equipment

**Syntax**      EQUIPment?

**Details**      The EQUIPment? query returns a comma-separated list of installed modules, which is model number from each module. For modular controllers, the first parameter is the model number of the chassis, followed by the model numbers of each module. For single channel instruments, it returns the model number of the instrument.

## **ERRors?**

**Synopsis**      Query for errors

**Syntax**      ERRors?

**Details** Returns a comma-delimited list of error codes. If no error has occurred, a 0 is returned.

A typical response might look like:

```
201,124
```

**See Also** ERRSTR?

## ERRSTR?

**Synopsis** Query for errors with string descriptions

**Syntax** ERRSTR?

**Details** Similar to the ERR? query, but a string description is included with the error code. A typical response might look like:

```
201,"Out of range",124,"Data mismatch"
```

**See Also** ERR?

## HEXFLOAT

**Synopsis** Enables the hex float mode

**Syntax** HEXFLOAT *enable*

**Details** Enables transmission of hex floats from the instrument. See the section *Working with Radixes and Hex Floats* for more information on this mode.

Argument	Value	Description
<i>enable</i>	0	Hex float mode disabled
	1	Hex float mode enabled

**See Also** HEXFLOAT?

## HEXFLOAT?

**Synopsis** Query the state of the hex float mode

**Syntax** HEXFLOAT?

**Details** Returns the enable state of hex mode. See the HEXFLOAT command for more details.

**See Also** HEXFLOAT

## LASer:AUX?

**Synopsis** Query the auxiliary voltage inputs

**Syntax** LASer:AUX? *index*

**Details** Returns the voltage (in volts) measured by the auxiliary voltage inputs. Only certain instruments (such as the LaserPak) support these inputs, see your instrument manual for

details. For instruments that feature multiple auxiliary inputs, the *index* argument specifies which input to return.

Argument	Value	Description
<i>index</i>	0	Returns the first auxiliary input voltage
	1	Returns the second auxiliary input voltage
	Etc...	

If the auxiliary input is configured to be used as a temperature input, use the LAS:R? and LAS:T? queries to return the resistance or temperature measurement, respectively.

**See Also** LASer:R?, LASer:T?

## LASer:CABLER

**Synopsis** Set cable resistance compensation

**Syntax** LASer:CABLER *resistance*

**Details** Set the cable resistance to adjust the voltage measurement to display the voltage at the laser, compensating for the voltage losses in the cable and connectors. See your instrument manual for additional information on using this feature.

Argument	Value	Description
<i>resistance</i>	0.001 to 10	Cable resistance, in ohms

**See Also** LASer:CABLER?

## LASer:CABLER?

**Synopsis** Query the cable resistance compensation

**Syntax** LASer:CABLER?

**Details** Returns the value of the cable resistance compensation value. See the LASer:CABLER command for more details.

**See Also** LASer:CABLER

## LASer:CALMD (CALPD)

**Synopsis** Set photodiode optical response

**Syntax** LASer:CALMD *PDresp*

**Details** Set the photodiode optical response value for converting photodiode current into optical power. This is the PD Response value from the instrument menu.

Argument	Value	Description
<i>PDresp</i>	0.0001 to 1000	Sensitivity in $\mu\text{A}/\text{mW}$

If *PDresp* is zero, then power mode will be unavailable, and the instrument will display photodiode current. If *PDresp* is non-zero, then power mode will be allowed, and the instrument will display optical power.

**See Also** LASer:CALMD?

---

**LASer:CALMD? (CALPD?)****Synopsis** Query the photodiode optical response**Syntax** LASer:CALMD? (CALPD?)**Details** Returns the value of the photodiode optical response. See the LASer:CALMD command for more details.**See Also** LASer:CALMD

---

**LASer:CAL:CANCEL****Synopsis** Cancel calibration in progress**Syntax** LASer:CAL:CANCEL**Details** Cancels any calibration in progress.

---

**LASer:CAL:LDX, LDX?, LVX, LVX?, MDX, MDX?****Synopsis** Laser calibration functions**Details** See the calibration section for details on how to use these commands.

---

**LASer:COND?****Synopsis** Query laser condition**Syntax** LASer:COND?**Details** Returns the laser condition register.

Response	Bit	Value	Description
<i>conditions</i>	0	1	Current limit
	1	2	Voltage limit
	2	4	Photodiode current limit
	3	8	Photodiode power limit
	4	16	Interlock disabled
	5	32	Unused
	6	64	Unused
	7	128	Laser open circuit
	8	256	Laser shorted
	9	512	Out of tolerance
	10	1024	Output on
	11	2048	Unused
	12	4096	Unused
	13	8192	R limit
	14	16384	T limit
	15	32768	Unused

**See Also** LASer:ENABLE:COND, \*STB?

---

**LASer:CONST****Synopsis** Set the Sheinhart-Hart temperature conversion constants

**Syntax** LASer:CONST A, B, C

**Details** Sets the constants used in converting the resistance of auxiliary temperature inputs to °C using the Steinhart-Hart formula. By default, the thermistor constants are set to those for a BetaTHERM 10K3A1 thermistor.

Argument	Value	Description
A	± 9.9999	First Steinhart-Hart constant ( $\times 10^{-3}$ )
B	± 9.9999	Second Steinhart-Hart constant ( $\times 10^{-4}$ )
C	± 9.9999	Third Steinhart-Hart constant ( $\times 10^{-7}$ )

Some vendors may refer to A, B, and C as C1, C2, & C3.

See your user's manual for more information on using this function.

**See Also** LASer:CONST?

### LASer:CONST?

**Synopsis** Query the Steinhart-Hart temperature conversion constants

**Syntax** LASer:CONST?

**Details** Returns the sensor temperature conversion constants. See the LASer:CONST command for a complete definition of the A, B, and C response values.

**See Also** LASer:CONST

### LASer:DC

**Synopsis** Set the duty cycle in QCW mode

**Syntax** LASer:DC *dutycycle*

**Details** This changes the duty cycle percentage when in QCW mode. A new frequency value will be computed based on the duty cycle and pulse width (the pulse width is not changed). If the resulting frequency is outside the allowable range, an E-201 error will be generated and the command ignored.

Argument	Description
<i>dutycycle</i>	Duty cycle, in percent

**See Also** LASer:DC?, LASer:F, LASer:PWF, LASer:PWP

### LASer:DC?

**Synopsis** Query the QCW duty cycle

**Syntax** LASer:DC?

**Details** Returns the value of the duty cycle. See LASer:DC for a definition of the *dutycycle* response value.

**See Also** LASer:DC

---

**LASer:DEC****Synopsis**      Decrement the laser set point**Syntax**        LASer:DEC *steps* [, *time*]**Details**        The LASer:DEC command uses the step size defined with the LASer:STEP command to decrement the laser set point. If the *time* parameter is omitted, then the set point is immediately decremented *steps* times the step size. If the *time* parameter is included, then the instrument will still decrement *steps* times the step size, but will pause at each step for *time* milliseconds.

Argument	Value	Description
<i>steps</i>	1 to 65000	Number of steps to decrement
<i>time</i>	0 to 65000	Number of milliseconds to pause between each step

**See Also**      LASer:INC, LASer:STEP

---

**LASer:DELAYIN****Synopsis**      Set the pulse delay, in seconds, between the trigger input and output pulse**Syntax**        LASer:DELAYIN *delay***Details**        Sets the delay from the rising edge of TRIGIN to the generation of the output pulse in QCW mode.

Argument	Value	Description
<i>delay</i>	0.000015 to 1	Delay, in seconds

**See Also**      LASer:DELAYIN?, LASer:DELAYOUT

---

**LASer:DELAYIN?****Synopsis**      Query the trigger input pulse delay**Syntax**        LASer:DELAYIN?**Details**        Returns the value of the trigger input delay. See LASer:DELAYIN for a definition of the *delay* response value.**See Also**      LASer:DELAYIN

---

**LASer:DELAYOUT****Synopsis**      Set the trigger output delay, in seconds, between the output pulse and the trigger output**Syntax**        LASer:DELAYOUT *delay***Details**        Sets the delay from the rising edge of the pulse to the rising edge of TRIGOUT in QCW mode.

Argument	Value	Description
<i>delay</i>	0.000000 to 1	Delay, in seconds

**See Also**      LASer:DELAYOUT?, LASer:DELAYIN

---

**LASer:DELAYOUT?****Synopsis** Query the trigger output delay**Syntax** LASer:DELAYOUT?**Details** Returns the value of the trigger output delay. See LASer:DELAYOUT for a definition of the *delay* response value.**See Also** LASer:DELAYOUT

---

**LASer:DISplay****Synopsis** Set laser display enable**Syntax** LASer:DISplay *enable***Details** The LASer:DISplay command can be used to completely lock out local operation of the instrument and display “Display Disabled” instead of the normal display.

Response	Value	Description
<i>enable</i>	0	Disables the display and front panel
	1	Enables the display and front panel

Once the display is disabled, the front panel is completely locked out. The only way to restore functionality to the front panel is to issue a “LASer:DISplay 1” or cycle power on the unit.

**See Also** LASer:DISplay?

---

**LASer:DISplay?****Synopsis** Query the display enable state**Syntax** LASer:DISplay?**Details** Returns the value of the laser display enable state. See LASer:DISplay for a definition of the *enable* response value.**See Also** LASer:DISplay

---

**LASer:ENABLE:COND****Synopsis** Set Laser Condition Enable register**Syntax** LASer:ENABLE:COND *conditions***Details** Enables reporting of selected conditions to the Status Byte Register. See the LASer:COND command for a definition of the *conditions* parameter.**See Also** LASer:ENABLE:COND?, LASer:COND?

---

**LASer:ENABLE:COND?****Synopsis** Query Laser Condition Enable register

**Syntax** LASer:ENABLE:COND?

**Details** Returns the value of the Laser Condition Enable register. See the LASer:COND command for a definition of the *conditions* response.

**See Also** LASer:ENABLE:COND, LASer:COND?

---

### **LASer:ENABLE:EVENT**

**Synopsis** Set Laser Event Enable register

**Syntax** LASer:ENABLE:EVENT *events*

**Details** Enables reporting of selected events to the Status Byte Register. See the LASer:EVENT command for a definition of the *events* parameter.

**See Also** LASer:ENABLE:EVENT?, LASer:EVENT?

---

### **LASer:ENABLE:EVENT?**

**Synopsis** Query Laser Event Enable register

**Syntax** LASer:ENABLE:EVENT?

**Details** Returns the value of the Laser Event Enable register. See the LASer:EVENT command for a definition of the *events* response.

**See Also** LASer:ENABLE:EVENT, LASer:EVENT?

---

### **LASer:ENABLE:OUTOFF**

**Synopsis** Set the Output Off Enable register

**Syntax** LASer:ENABLE:OUTOFF *outoff*

**Details** The Output Off register controls what conditions will cause the laser output to be turned off. Some conditions are always enabled, as indicated below. The factory default conditions are shown in **bold**.

Argument	Bit	Value	Description
<i>outoff</i>	0	1	Current limit
	<b>1</b>	<b>2</b>	<b>Voltage limit (<i>always enabled</i>)</b>
	<b>2</b>	<b>4</b>	<b>Photodiode current limit</b>
	<b>3</b>	<b>8</b>	<b>Photodiode power limit</b>
	<b>4</b>	<b>16</b>	<b>Interlock disabled (<i>always enabled</i>)</b>
	5	32	Unused
	6	64	Unused
	<b>7</b>	<b>128</b>	<b>Laser open circuit (<i>always enabled</i>)</b>
	<b>8</b>	<b>256</b>	<b>Laser shorted (<i>always enabled</i>)</b>
	9	512	Out of tolerance
	10	1024	TEC off
	11	2048	TEC temperature limit
	<b>12</b>	<b>4096</b>	<b>Hardware error (<i>always enabled</i>)</b>
	13	8192	R limit
	14	16384	T limit
	<b>15</b>	<b>32768</b>	<b>Thermal limit exceeded (<i>always enabled</i>)</b>

The default value for this register is 37278.

**See Also** LASer:ENABLE:OUTOFF?

---

### LASer:ENABLE:OUTOFF?

**Synopsis** Query the Output Off Enable register

**Syntax** LASer:ENABLE:OUTOFF?

**Details** Returns the value of the Output Off register. See the LASer:ENABLE:OUTOFF command for definition of *outoff* response value.

**See Also** LASer:ENABLE:OUTOFF

---

### LASer:EVENT?

**Synopsis** Query the value of the laser event register

**Syntax** LASer:EVENT?

**Details** Returns the laser event register.

Response	Bit	Value	Description
<i>events</i>	0	1	Current limit
	1	2	Voltage limit
	2	4	Photodiode current limit
	3	8	Photodiode power limit
	4	16	Interlock state changed
	5	32	Unused
	6	64	Unused
	7	128	Laser open circuit
	8	256	Laser short circuit
	9	512	Out of tolerance changed state
	10	1024	Output changed state
	11	2048	New data
	12	4096	Unused
	13	8192	R limit
	14	16384	T limit
15	32768	Unused	

After reading the event register, the event register is set to zero.

**See Also** LASer:ENABLE:EVENT

## LASer:F

**Synopsis** Set the frequency of pulses in QCW mode

**Syntax** LASer:F *frequency*

**Details** Changes the frequency when in QCW mode. A new duty cycle value will be computed based on the frequency and pulse width (the pulse width is not changed). If the resulting duty cycle is outside the allowable range, an E-201 error will be generated and the command ignored.

Argument	Description
<i>frequency</i>	Frequency, in hertz

**See Also** LASer:F?

## LASer:F?

**Synopsis** Query the pulse frequency

**Syntax** LASer:F?

**Details** Returns the value of the frequency. See LASer:F for a definition of the *frequency* response value.

**See Also** LASer:F

## LASer:INC

**Synopsis** Increment the laser set point

**Syntax** LASer:INC *steps* [, *time*]

**Details** The LASer:INC command uses the step size defined with the LASer:STEP command to increment the laser set point. If the *time* parameter is omitted, then the set point is

immediately incremented *steps* times the step size. If the *time* parameter is included, then the instrument will still increment *steps* times the step size, but will pause at each step for *time* milliseconds.

Argument	Value	Description
<i>steps</i>	1 to 65000	Number of steps to increment
<i>time</i>	0 to 65000	Number of milliseconds to pause between each step

**See Also** LASer:DEC, LASer:STEP

## LASer:INTCONT

**Synopsis** Enables or disables the intermittent contact detection

**Syntax** LASer:INTCONT *enable*

**Details** The LASer:INTCONT turns the intermittent contact detection of the laser driver on or off. See the user's manual for more details.

Argument	Value	Description
<i>enable</i>	0	Turn intermittent contact detection off
	1	Turn intermittent contact detection on

**See Also** LASer:INTCONT?

## LASer:INTCONT?

**Synopsis** Query the state of the intermittent contact setting

**Syntax** LASer:INTCONT?

**Details** Returns the intermittent contact setting.

Response	Description
<i>enable</i>	The state of the intermittent contact setting

**See Also** LASer:INTCONT

## LASer:LDI (I)

**Synopsis** Set the laser current set point

**Syntax** LASer:LDI *setpoint*

**Details** The LASer:LDI sets the laser current set point. An error will be generated if the value of *setpoint* is greater than the current limit.

Argument	Value	Description
<i>setpoint</i>	0 to $I_{lim}$	Set the current set point in milliamps

**See Also** LASer:LDI?, LASer:LIMit:LDI, LASer:SET:LDI

## LASer:LDI? (I?)

**Synopsis** Query the actual laser current

**Syntax** LASer:LDI?

**Details** Returns the actual (measured) laser current.

Response	Description
<i>current</i>	The actual (measured) current in milliamps

**See Also** LASer:LDI

## LASer:LDIRES

**Synopsis** Set the laser current resolution

**Syntax** LASer:LDIRES *resolution*

**Details** The LASer:LDIRES sets the laser current resolution for both the set point and measurement. An error will be generated if the value of *resolution* is not supported. See the LASer:LDIRESAVAIL? query to get a list of supported values.

Argument	Value	Description
<i>resolution</i>	<i>Model-specific</i>	Sets the resolution in milliamps

This function is only available in firmware version 2.0 and later.

**See Also** LASer:LDIRES?, LASer:LDIRESAVAIL?

## LASer:LDIRES?

**Synopsis** Query the laser current resolution value

**Syntax** LASer:LDIRES?

**Details** Returns the laser current resolution value.

Response	Description
<i>resolution</i>	The resolution value, in milliamps

This function is only available in firmware version 2.0 and later.

**See Also** LASer:LDIRES, LASer:LDIRESAVAIL?

## LASer:LDIRESAVAIL?

**Synopsis** Query the list of available laser current resolution values

**Syntax** LASer:LDIRESAVAIL?

**Details** Returns a list of available current resolutions, which can be used with the LASer:LDIRES command to change the laser current set point and measurement resolution.

Response	Description
<i>resolution</i>	A comma separated list of available resolutions, in milliamps

This function is only available in firmware version 2.0 and later.

**See Also** LASer:LDIRES, LASer:LDIRES?

---

**LASer:LDV****Synopsis** Set the laser voltage set point**Syntax** LASer:LDV *setpoint***Details** The LASer:LDV sets the laser voltage set point.

Argument	Value	Description
<i>setpoint</i>	0 to $V_{\text{limit}}$	Set the voltage set point in volts

**See Also** LASer:LIMit:LDV?, LASer:LDV?, LASer:SET:LDV?

---

**LASer:LDV?****Synopsis** Query the laser actual voltage**Syntax** LASer:LDV?**Details** Returns the actual laser voltage.

Response	Description
<i>voltage</i>	Actual laser voltage, in volts

**See Also** LASer:LIMit:LDV, LASer:LDV

---

**LASer:LIMit:LDI (I)****Synopsis** Set the laser current limit**Syntax** LASer:LIMit:LDI *limit***Details** The LASer:LDI sets the laser current limit. If the laser current set point is greater than the value of *limit*, then laser current set point will be reduced to the value of *limit*.

Argument	Value	Description
<i>limit</i>	0 to $I_{\text{max}}$	Set the laser current limit in milliamps

**See Also** LASer:LDI, LASer:LIMit:LDI?

---

**LASer:LIMit:LDI? (I?)****Synopsis** Query the laser current limit**Syntax** LASer:LIMit:LDI?**Details** Returns the value of the laser current limit. See LASer:LIMit:LDI for a definition of the *limit* response value.**See Also** LASer:LIMit:LDI

---

**LASer:LIMit:LDV****Synopsis** Set the laser voltage limit**Syntax** LASer:LIMit:LDV *limit*

**Details** The LASer:LIMit:LDV sets the laser voltage limit. If the laser voltage set point is greater than the value of *limit*, then laser voltage set point will be reduced to the value of *limit*.

Argument	Value	Description
<i>limit</i>	0 to $V_{max}$	Set the laser voltage limit in volts

**See Also** LASer:LIMit:LDV?, LASer:LDV

### LASer:LIMit:LDV?

**Synopsis** Query the laser voltage limit

**Syntax** LASer:LIMit:LDV?

**Details** Returns the value of the laser voltage limit. See LASer:LIMit:LDV for a definition of the *limit* response value.

**See Also** LASer:LIMit:LDV, LASer:LDV

### LASer:LIMit:MDI (IPD)

**Synopsis** Set the photodiode current limit

**Syntax** LASer:LIMit:MDI *limit*

**Details** The LASer:LIMit:MDI sets the photodiode current limit. If the photodiode current set point is greater than the value of *limit*, then photodiode current set point will be reduced to the value of *limit*.

Argument	Value	Description
<i>limit</i>	0 to $I_{m_{max}}$	Set the photodiode current limit in microamps

**See Also** LASer:LIMit:MDI?, LASer:MDI

### LASer:LIMit:MDI? (IPD?)

**Synopsis** Query the photodiode current limit

**Syntax** LASer:LIMit:MDI?

**Details** Returns the value of the photodiode current limit. See LASer:LIMit:MDI for a definition of the *limit* response value.

**See Also** LASer:LIMit:MDI, LASer:MDI

### LASer:LIMit:MDP (Ppd)

**Synopsis** Set the photodiode power limit

**Syntax** LASer:LIMit:MDP *limit*

**Details** The LASer:LIMit:MDP sets the photodiode power limit. If the photodiode power set point is greater than the value of *limit*, then photodiode power set point will be reduced to the value of *limit*.

Argument	Value	Description
<i>limit</i>	0 to $P_{max}$	Set the photodiode power limit in milliwatts

**See Also** LASer:LIMit:MDP?, LASer:MDP

### LASer:LIMit:MDP? (Ppd?)

**Synopsis** Query the photodiode power limit

**Syntax** LASer:LIMit:MDP?

**Details** Returns the value of the photodiode power limit. See LASer:LIMit:MDP for a definition of the *limit* response value.

**See Also** LASer:LIMit:MDP, LASer:MDP

### LASer:LIMit:RHIGH

**Synopsis** Set the high resistance limit

**Syntax** LASer:LIMit:RHIGH *limit*

**Details** The LASer:LIMit:RHI sets the upper resistance limit for auxiliary temperature sensor measurements. If the measured resistance from the temperature sensor exceeds the high limit, and the corresponding bit is enabled in the OUTOFF register, the laser output will be turned off.

Argument	Value	Description
<i>limit</i>	0 to 50000	Set the high resistance limit, in ohms

**See Also** LASer:ENABLE:OUTOFF, LASer:LIMit:RHIGH?, LASer:RLOW

### LASer:LIMit:RHIGH?

**Synopsis** Query the high resistance limit

**Syntax** LASer:LIMit:RHIGH?

**Details** Returns the value of the high resistance limit. See LASer:LIMit:RHIGH for a definition of the *limit* response value.

**See Also** LASer:LIMit:RHIGH

### LASer:LIMit:RLOW

**Synopsis** Set the low resistance limit

**Syntax** LASer:LIMit:RLOW *limit*

**Details** The LASer:LIMit:RLO sets the lower resistance limit for auxiliary temperature sensor measurements. If the measured resistance from the temperature sensor exceeds the low limit, and the corresponding bit is enabled in the OUTOFF register, the laser output will be turned off.

Argument	Value	Description
<i>limit</i>	0 to 50000	Set the low resistance limit, in ohms

**See Also** LASer:ENABle:OUTOFF, LASer:LIMit:RLOw?, LASer:RLOw

### LASer:LIMit:RLOw?

**Synopsis** Query the low resistance limit

**Syntax** LASer:LIMit:RLOw?

**Details** Returns the value of the low resistance limit. See LASer:LIMit:RLOw for a definition of the *limit* response value.

**See Also** LASer:LIMit:RLOw

### LASer:LIMit:THIgh

**Synopsis** Set the high temperature limit

**Syntax** LASer:LIMit:THIgh *limit*

**Details** The LASer:LIMit:THI sets the upper temperature limit for auxiliary temperature sensor measurements. If the measured temperature from the temperature sensor exceeds the high limit, and the corresponding bit is enabled in the OUTOFF register, the laser output will be turned off.

Argument	Value	Description
<i>limit</i>	0 to 50000	Set the high temperature limit, in °C

**See Also** LASer:ENABle:OUTOFF, LASer:LIMit:THIgh?, LASer:TLOw

### LASer:LIMit:THIgh?

**Synopsis** Query the high temperature limit

**Syntax** LASer:LIMit:THIgh?

**Details** Returns the value of the high temperature limit. See LASer:LIMit:THIgh for a definition of the *limit* response value.

**See Also** LASer:LIMit:THIgh

### LASer:LIMit:TLOw

**Synopsis** Set the low temperature limit

**Syntax** LASer:LIMit:TLOw *limit*

**Details** The LASer:LIMit:TLO sets the lower temperature limit for auxiliary temperature sensor measurements. If the measured temperature from the temperature sensor exceeds the low limit, and the corresponding bit is enabled in the OUTOFF register, the laser output will be turned off.

Argument	Value	Description
<i>limit</i>	0 to 50000	Set the low temperature limit, in °C

**See Also** LASer:ENABLE:OUTOFF, LASer:LIMit:TLOW?, LASer:TLOW

### LASer:LIMit:TLOW?

**Synopsis** Query the low temperature limit

**Syntax** LASer:LIMit:TLOW?

**Details** Returns the value of the low temperature limit. See LASer:LIMit:TLOW for a definition of the *limit* response value.

**See Also** LASer:LIMit:TLOW

### LASer:MDI (IPD)

**Synopsis** Set the photodiode current set point

**Syntax** LASer:MDI *setpoint*

**Details** The LASer:MDI sets the photodiode current set point.

Argument	Value	Description
<i>setpoint</i>	0 to $I_{m_{limit}}$	Set the photodiode current set point in microamps

**See Also** LASer:LIMit:MDI?, LASer:MDI?, LASer:SET:MDI?

### LASer:MDI? (IPD?)

**Synopsis** Query the actual photodiode current

**Syntax** LASer:MDI?

**Details** Returns the actual photodiode current.

Response	Description
<i>PDcurrent</i>	Actual photodiode current, in microamps

**See Also** LASer:LIMit:MDI, LASer:MDI

### LASer:MDP (Ppd)

**Synopsis** Set the photodiode power set point

**Syntax** LASer:MDP *setpoint*

**Details** The LASer:MDP sets the photodiode power set point.

Argument	Value	Description
<i>setpoint</i>	0 to $Po_{limit}$	Set the photodiode power set point in milliwatts

**See Also** LASer:LIMit:MDP?, LASer:MDP?, LASer:SET:MDP?

---

**LASer:MDP? (Ppd?)****Synopsis** Query the actual photodiode power**Syntax** LASer:MDP?**Details** Returns the actual photodiode power.

Response	Description
<i>PDpower</i>	Actual photodiode power, in milliwatts

**See Also** LASer:LIMit:MDP, LASer:MDP

---

**LASer:MODE?****Synopsis** Query operational mode**Syntax** LASer:MODE?**Details** Returns the control mode of the laser driver.

Response	Value	Description
<i>mode</i>	ILBW	Laser current control (Io/ACC) mode, low bandwidth
	IHBW	Laser current control (Io HiBW) mode, high bandwidth
	PULSE	Laser current control (Io Pulse) mode, QCW internal trigger
	TRIG	Laser current control (Io Ext Trig) mode, QCW external trigger
	BURST	Laser current control (Io Burst) mode, QCW burst
	MDI	Photodiode current control (Im/AMC) mode
	MDP	Photodiode power control (Po/APC) mode
	LDV	Laser voltage control (Vf/AVC) mode

**See Also** LASer:MODE:ILBW, LASer:MODE:IHBW, LASer:MODE:PULSE, LASer:MODE:TRIG, LASer:MODE:BURST, LASer:MODE:MDI, LASer:MODE:MDP; LAS:MODE:LDV

---

**LASer:MODE:BURST****Synopsis** Set operational mode to laser current control, QCW burst.**Syntax** LASer:MODE:BURST**Details** Switches the Laser driver to laser current control mode, QCW burst. Pulses are triggered with a LASer:TRIGGER command or via the external trigger. The number of pulses is controlled by the LASer:QCWCOUNT command. If the mode is being changed and the laser output is on, the laser output will be turned off and an error generated.**See Also** LASer:MODE?, LASer:QCWCOUNT, LASer:TRIGGER

---

**LASer:MODE:ICW (CW)****Synopsis** Set operational mode to laser current control, continuous wave**Syntax** LASer:MODE:ICW**Details** This command is identical to LASer:MODE:ILBW, and is included for compatibility reasons. If a LASer:MODE? query is done after this command is issued, "ILBW" will be returned.

**See Also** LASer:MODE:ILBW

---

### **LASer:MODE:ILBW (I)**

**Synopsis** Set operational mode to laser current control, low bandwidth

**Syntax** LASer:MODE:ILBW

**Details** Switches the Laser driver to laser current control mode, low bandwidth. If the mode is being changed and the laser output is on, the laser output will be turned off and an error generated.

**See Also** LASer:MODE?

---

### **LASer:MODE:IHBW**

**Synopsis** Set operational mode to laser current control, high bandwidth

**Syntax** LASer:MODE:IHBW

**Details** Switches the Laser driver to laser current control mode, high bandwidth. If the mode is being changed and the laser output is on, the laser output will be turned off and an error generated.

**See Also** LASer:MODE?

---

### **LASer:MODE:MDI (IPD)**

**Synopsis** Set operational mode to photodiode current control

**Syntax** LASer:MODE:MDI

**Details** Switches the Laser driver to photodiode current control mode. If the mode is being changed and the laser output is on, the laser output will be turned off and an error generated.

**See Also** LASer:MODE?

---

### **LASer:MODE:MDP (Ppd)**

**Synopsis** Set operational mode to photodiode power control

**Syntax** LASer:MODE:MDP

**Details** Switches the Laser driver to photodiode power control mode. If the mode is being changed and the laser output is on, the laser output will be turned off and an error generated.

**See Also** LASer:MODE?

---

### **LASer:MODE:LDV**

**Synopsis** Set operational mode to laser voltage control

**Syntax** LASer:MODE:LDV

**Details** Switches the Laser driver to laser voltage control mode. If the mode is being changed and the laser output is on, the laser output will be turned off and an error generated.

**See Also** LASer:MODE?

### LASer:MODE:PULSE

**Synopsis** Set operational mode to laser current control, QCW mode, internal trigger

**Syntax** LASer:MODE:PULSE

**Details** Switches the Laser driver to laser current control mode, QCW. Pulses are internally generated using the pulse width, duty cycle, and frequency settings. If the mode is being changed and the laser output is on, the laser output will be turned off and an error generated.

**See Also** LASer:MODE?

### LASer:MODE:TRIG

**Synopsis** Set operational mode to laser current control, QCW mode, external trigger

**Syntax** LASer:MODE:TRIG

**Details** Switches the Laser driver to laser current control mode, QCW. Pulses are externally triggered with the TRIGIN signal. Each rising edge of TRIG in generates one pulse. Pulse width is determined by the pulse width setting. Duty cycle and frequency are ignored. If the mode is being changed and the laser output is on, the laser output will be turned off and an error generated.

**See Also** LASer:MODE?

### LASer:OUTput

**Synopsis** Set the laser output state

**Syntax** LASer:OUTput *state*

**Details** Turns the laser output on or off.

Argument	Value	Description
<i>state</i>	0	Turn the output off
	1	Turn the output on

**See Also** LASer:OUTput?

### LASer:OUTput?

**Synopsis** Query the laser output state

**Syntax** LASer:OUTput?

**Details** Returns the value of the laser current limit. See LASer:OUT for a definition of the *state* response value.

**See Also** LASer:OUTput

---

**LASer:PDBias****Synopsis** Set the photodiode bias voltage set point**Syntax** LASer:PDBias *voltage***Details** Sets the photodiode bias voltage set point.

Argument	Value	Description
<i>voltage</i>	0 to 5	Negative voltage bias set point, in volts

**See Also** LASer:PDBias?

---

**LASer:PDBias?****Synopsis** Query the photodiode bias voltage set point**Syntax** LASer:PDBias?**Details** Returns the value of the photodiode bias voltage set point. See LASer:PDBias for a definition of the *voltage* response value.**See Also** LASer:PDBias

---

**LASer:PW?****Synopsis** Query the pulse width**Syntax** LASer:PW?**Details** Returns the pulse width, in milliseconds. See LASer:PWF and LASer:PWP for more information on setting the pulse width.**See Also** LASer:PWF, LASer:PWP

---

**LASer:PWF****Synopsis** Set the pulse width while holding the frequency constant**Syntax** LASer:PWF *pulsewidth***Details** Changes the pulse width when in QCW mode. A new duty cycle value will be computed based on the frequency and pulse width. If the resulting duty cycle is outside the allowable range, an E-201 error will be generated and the command ignored.

Argument	Description
<i>pulsewidth</i>	Pulse width, in milliseconds

**See Also** LASer:PW?, LASer:PWP, LASer:QCWHOLD

---

**LASer:PWP****Synopsis** Set the pulse width while holding the duty cycle constant**Syntax** LASer:PWP *pulsewidth*

**Details** Changes the pulse width when in QCW mode. A new frequency value will be computed based on the duty cycle and pulse width. If the resulting frequency is outside the allowable range, an E-201 error will be generated and the command ignored.

Argument	Description
<i>pulsewidth</i>	Pulse width, in milliseconds

**See Also** LASer:PW?, LASer:PWF, LASer:QCWHOLD

## LASer:QCWHOLD

**Synopsis** Set the QCW (pulse) hold setting for pulse width adjustments in the user interface

**Syntax** LASer:QCWHOLD *hold*

**Details** Sets which value to hold constant (frequency or duty cycle) when making pulse width adjustments in the user interface.

This command has no effect on the LASer:PWF and LASer:PWP commands, as they explicitly define which will be held constant. It only affects the pulse width adjustments done through the front panel.

Argument	Value	Description
<i>hold</i>	0	Hold frequency constant
	1	Hold duty cycle constant

**See Also** LASer:QCWHOLD?

## LASer:QCWHOLD?

**Synopsis** Queries the QCW (pulse) hold setting for pulse width adjustments in the user interface

**Syntax** LASer:QCWHOLD?

**Details** Returns the value of the hold constant. See LASer:QCWHOLD for a definition of the *hold* response value.

**See Also** LASer:QCWHOLD

## LASer:QCWCOUNT

**Synopsis** Set the QCW burst count

**Syntax** LASer:QCWCOUNT *count*

**Details** Sets how many pulses will be generated for each burst when operating in IO (Burst) mode.

Argument	Value	Description
<i>count</i>	1 – 60,000	Number of pulses per burst

**See Also** LASer:QCWCOUNT?

---

**LASer: QCWCOUNT?**

**Synopsis**      Queries the number of pulses generated for each burst when operating in Io (Burst) mode.

**Syntax**        LASer:QCWCOUNT?

**Details**        Returns the pulse count.. See LASer:QCWCOUNT for a definition of the *count* response value.

**See Also**      LASer:QCWCOUNT

---

**LASer: R?**

**Synopsis**      Queries the resistance measurement of the auxiliary temperature sensor

**Syntax**        LASer:R?

**Details**        Returns the resistance measurement, in ohms, from the auxiliary temperature sensor input.

**See Also**      LASer:T?

---

**LASer:RANGE**

**Synopsis**      Set the output current range

**Syntax**        LASer:RANGE *range*

**Details**        For instruments that support dual range operation, this command sets the low or high range.

Argument	Value	Description
<i>range</i>	0	Low range
	1	High range

**See Also**      LASer:RANGE?

---

**LASer: RANGE?**

**Synopsis**      Queries the laser current range.

**Syntax**        LASer:RANGE?

**Details**        Returns the current range. See LASer:RANGE for a definition of the *range* response value.

**See Also**      LASer:RANGE

---

**LASer:SET:LDI? (!?)**

**Synopsis**      Query the laser current set point

**Syntax**        LASer:SET:LDI?

**Details**        Returns the value of the laser current set point. See LASer:LDI for a definition of the *setpoint* response value.

**See Also** LASer:LDI

### LASer:SET:MDI? (IPD?)

**Synopsis** Query the photodiode current set point

**Syntax** LASer:SET:MDI? (IPD?)

**Details** Returns the value of the photodiode current set point. See LASer:MDI for a definition of the *setpoint* response value.

**See Also** LASer:MDI

### LASer:SET:MDP? (Ppd?)

**Synopsis** Query the photodiode power set point

**Syntax** LASer:SET:MDP? (Ppd?)

**Details** Returns the value of the photodiode power set point. See LASer:MDP for a definition of the *setpoint* response value.

**See Also** LASer:MDP

### LASer:SET:LDV?

**Synopsis** Query the laser voltage set point

**Syntax** LASer:SET:LDV?

**Details** Returns the value of the laser voltage set point. See LASer:LDV for a definition of the *setpoint* response value.

**See Also** LASer:LDV

### LASer:STB?

**Synopsis** Query the laser status byte

**Syntax** LASer:STB?

**Details** Returns a summary of the enabled conditions within the laser condition and event registers. These bits mirror the bits in the Status Byte Register.

Argument	Bit	Value	Description
<i>status</i>	2	4	Event status register summary
	3	8	Condition status register summary

The values are additive, so a return value of 0, 4, 8, or 12 is possible.

**See Also** \*STB?, LASer:COND?, LASer:ENAB:COND, LASer:ENABLE:EVENT, LASer:EVENT?

### LASer:STEP

**Synopsis** Set laser step size

**Syntax** LASer:STEP *size*

**Details** The command sets the laser step size used by the LASer:INC or LASer:DEC commands.

Argument	Value	Description
<i>size</i>	1 to 65000	Step size

A step of 1 equates to 0.01mA, 0.01mW, or 1 $\mu$ A, depending on the control mode.

**See Also** LASer:DEC, LASer:INC, LASer:STEP?

## LASer:STEP?

**Synopsis** Query the laser step size

**Syntax** LASer:STEP?

**Details** Returns the value of the laser step size. See LASer:STEP for a definition of the *size* response value.

**See Also** LASer:STEP

## LASer: T?

**Synopsis** Queries the converted temperature measurement of the auxiliary temperature sensor

**Syntax** LASer:T?

**Details** Returns the converted temperature measurement, in  $^{\circ}$ C, of the auxiliary temperature sensor input. The conversion from resistance to temperature is done using the Steinhart-Hart formula and the constants set by the LASer:CONST command.

**See Also** LASer:CONST, LASer:R?

## LASer:TOLerance

**Synopsis** Set the laser tolerance criteria

**Syntax** LASer:TOLerance *tolerance, time*

**Details** The LASer:TOLerance command allows control over when the output of the laser driver is considered in tolerance (or stable), in order to satisfy the tolerance condition of the operation complete definition. When used in conjunction with the \*WAI command, it can control when the next command is processed, delaying processing until the output stabilizes at its set point.

Argument	Value	Description
<i>tolerance</i>	0.01 to $I_{max}$	Current tolerance, in milliamps
<i>time</i>	0.1 to 50	Time window in seconds

To be considered in tolerance, the measured current must be within the set point plus or minus the *tolerance* value (the tolerance window) for *time* seconds. Any time it leaves the tolerance window, the timer will reset to zero and begin counting the next time it enters the tolerance window.

In photodiode current mode, the *tolerance* is fixed at 50 $\mu$ A. In photodiode power mode, *tolerance* is fixed at 50mW. In voltage control mode, *tolerance* is fixed at 50mV.

**See Also** LASer:TOLerance?

### LASer:TOLerance?

**Synopsis** Query the laser tolerance criteria

**Syntax** LASer:TOLerance?

**Details** Returns the value of the laser tolerance criteria. See LASer:TOLerance for a definition of the *tolerance* and *time* response values.

**See Also** LASer:TOLerance

### LASer:TRIGger

**Synopsis** Generates a single trigger

**Syntax** LASer:TRIGger

**Details** The LASer:TRIGger command is used to generate a trigger into the QCW system, and acts exactly like a single rising edge on the Trigger Input BNC. See the manual section on QCW operation for more information on triggering.

### LASer:USERCAL:GET?

**Synopsis** Query a laser USERCAL setting

**Syntax** LASer:USERCAL:GET? *index*

**Details** Returns the slope and offset compensation values for a specific user calibration *index*. See LASer:USERCAL:PUT for a definition of the *index* argument and *slope* and *offset* response values.

**See Also** LASer:USERCAL:PUT

### LASer:USERCAL:EDIT

**Synopsis** Enable or disabled user calibration editing

**Syntax** LASer:USERCAL:EDIT *enable*

**Details** Sets the edit enable state for user calibration. A LASer:USERCAL:PUT command will fail until the editing is enabled with this command.

Argument	Value	Description
<i>enable</i>	0	User calibration editing disabled
	1	User calibration editing enabled

**See Also** LASer:USERCAL:EDIT?, LASer:USERCAL:PUT

### LASer:USERCAL:EDIT?

**Synopsis** Query the state of the user calibration edit enable flag

<b>Syntax</b>	LASer:USERCAL:EDIT?
<b>Details</b>	Returns 0 if user calibration editing is disabled, and 1 if user calibration editing is enabled.
<b>See Also</b>	LASer:USERCAL:EDIT

## LASer:USERCAL:PUT

**Synopsis** Sets a USERCAL value

**Syntax** LASer:USERCAL:PUT *index, slope, offset*

**Details** The command sets the user calibration setting for a specific *index*, allowing for user compensation of measurements or set points. The default value for all slopes is 1, and the default value for all offsets is 0.

Compensation is applied according to the following formula:

$$\text{compensated} = \text{slope} * \text{uncompensated} + \text{offset}$$

Argument	Value	Description
<i>index</i>	1	lo set point, low range
	2	lo set point, high range
	3	Im set point
	4	Vf set point
	5	Vf set point, remote sense
	6	lo measurement, low range
	7	lo measurement, high range
	8	Im measurement
	9	Vf measurement
	10	Vf measurement, remote sense
	11	lo measurement, low range, QCW mode
	12	lo measurement, high range, QCW mode
	13	Im measurement, QCW mode
	14	Vf measurement, QCW mode
	15	Vf measurement, remote sense, QCW mode
	16	Auxiliary 1 voltage measurement
	17	Auxiliary 2 voltage measurement
	18	Auxiliary 1 resistance measurement
	19	Auxiliary 2 resistance measurement
<i>slope</i>	0.9 to 1.1	Slope compensation
<i>offset</i>	±5% of range	Offset compensation

LASer:USERCAL:PUT is only supported on certain instruments. Not all indexes are supported in every instrument. See your user's manual for details on what measurements are supported.

You must enable editing of user calibration values with the LASer:USERCAL:EDIT command.

**See Also** LASer:USERCAL:EDIT, LASer:USERCAL:GET?

---

**LASer:VSENSE****Synopsis** Select local or remote voltage sense**Syntax** LASer:VSENSE *select***Details** For instruments that support selectable local/remote voltage sense, this command selects local or remote sense.

Argument	Value	Description
<i>select</i>	0	Local voltage sense
	1	Remote voltage sense

**See Also** LASer:VSENSE?

---

**LASer: VSENSE?****Synopsis** Queries the voltage sense selection.**Syntax** LASer:VSENSE?**Details** For instruments that support selectable local/remote voltage sense, this command returns the *select* value. See the LASer:VSENSE command for more information.**See Also** LASer:VSENSE

---

**LOCAL****Synopsis** Returns instrument to local mode**Syntax** LOCAL**Details** After issuing the LOCAL command, the remote light will go out, and front panel controls will be enabled.

---

**MESsage****Synopsis** Set the message buffer**Syntax** MESsage *string***Details** Sets the internal message buffer to the value of *string*, up to a maximum of 16 characters.**See Also** MESsage?

---

**MESsage?****Synopsis** Query the message buffer**Syntax** MESsage?**Details** Returns the value of the message buffer.**See Also** MESsage

---

**ONDELAY****Synopsis** Set the laser output on delay**Syntax** ONDELAY *time***Details** The ONDELAY command controls how long the laser driver will delay between the time the user or command turns on the laser on LED and when the output is actually turned on.

Argument	Value	Description
<i>time</i>	0 to 30000	On delay, in milliseconds

Setting the ONDELAY value to 0 disables the delay feature.

**See Also** ONDELAY?

---

**ONDELAY?****Synopsis** Query the laser output on delay**Syntax** ONDELAY?**Details** Returns the value of the laser output on delay. See the ONDELAY command for a complete definition of possible return values.**See Also** ONDELAY

---

**RADix****Synopsis** Set the radix (number base)**Syntax** RADix *base***Details** By default, the instrument is set to decimal number base. Changing to an alternate number base will cause queries for integer values to be returned in the specified number base.

Argument	Value	Description
<i>base</i>	BIN	Binary format (base 2)
	OCT	Octal format (base 8)
	DEC	Decimal format (base 10)
	HEX	Hexadecimal format (base 16)

See the section *Working with Radixes and Hex Floats* for more information on using this command.**See Also** RADix?

---

**RADix?****Synopsis** Query the radix (number base)**Syntax** RADix?

**Details** Returns the current number base. See the RADix command for a complete definition of possible return values.

**See Also** RADix

## REMERR

**Synopsis** Set display of errors while in remote mode

**Syntax** REMERR *enable*

**Details** This command controls if the instrument will display errors while in remote mode. If set to zero, then errors will not be displayed. If set to one, errors will be displayed. Errors will always accumulate in the error queue.

Argument	Value	Description
<i>enable</i>	0	No not display errors in remote mode
	1	Display errors in remote mode

**See Also** REMERR?, ERR?, ERRSTR?

## REMERR?

**Synopsis** Query the display of errors while in remote mode

**Syntax** REMERR?

**Details** Returns the current REMERR setting. See the REMERR command for a complete definition of possible return values.

**See Also** REMERR

## REMSET

**Synopsis** Enables the changing of the set point from the front panel while in remote mode

**Syntax** REMERR *enable*

**Details** This command enables the changing of the set point from the front panel while in remote mode. By default, when in remote mode, the set point cannot be changed, as most programs do not react well to the set point being changed asynchronously from the program operation. However, it can be quite useful to be able to continue to adjust the instrument even when it is under remote control, and this command enables or disabled this capability.

Alternatively, if your program no longer needs to remotely control the instrument, you can use the LOCAL command to return the instrument to local control.

Argument	Value	Description
<i>enable</i>	0	Prevent set point change while in remote mode
	1	Enable set point change while in remote mode

**See Also** REMSET?, LOCAL

---

**REMSET?**

<b>Synopsis</b>	Query the state of the remote set mode
<b>Syntax</b>	REMSET?
<b>Details</b>	Returns the current REMSET setting. See the REMSET command for more information.
<b>See Also</b>	REMSET

---

**SCRIPT:PUT**

<b>Synopsis</b>	Set a script						
<b>Syntax</b>	SCRIPT:PUT <i>index</i> , <i>script</i>						
<b>Details</b>	Stores a script for execution by a function key, or with the SCRIPT:GO command. See the <i>Using Scripts</i> section above for more information.						
	<table> <thead> <tr> <th><u>Argument</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td><i>index</i></td> <td>Script index, 1 to 4</td> </tr> <tr> <td><i>script</i></td> <td>Script, maximum 200 characters.</td> </tr> </tbody> </table>	<u>Argument</u>	<u>Description</u>	<i>index</i>	Script index, 1 to 4	<i>script</i>	Script, maximum 200 characters.
<u>Argument</u>	<u>Description</u>						
<i>index</i>	Script index, 1 to 4						
<i>script</i>	Script, maximum 200 characters.						
<b>See Also</b>	SCRIPT:GET?, SCRIPT:GO						

---

**SCRIPT:GET?**

<b>Synopsis</b>	Retrieve a script
<b>Syntax</b>	SCRIPT:GET? <i>index</i>
<b>Details</b>	Returns the script stored in position <i>index</i> . See the <i>Using Scripts</i> section above for more information.
<b>See Also</b>	SCRIPT:PUT

---

**SCRIPT:GO**

<b>Synopsis</b>	Remotely executes a script
<b>Syntax</b>	SCRIPT:GO <i>index</i>
<b>Details</b>	Executes the script stored at position <i>index</i> . See the <i>Using Scripts</i> section above for more information.
<b>See Also</b>	SCRIPT:PUT

---

**SN?**

<b>Synopsis</b>	Query the serial number of the instrument
<b>Syntax</b>	SN?
<b>Details</b>	Returns the serial number of the instrument. This is the same information that is part of the *IDN? query.
<b>See Also</b>	*IDN?

## TEC:AUTOTUNE

**Synopsis** Start the AutoTune process

**Syntax** TEC:AUTOTUNE *temperature*

**Details** The TEC:AUTOTUNE command is used to start the AutoTune process, using the *temperature* parameter as the AutoTune point. The current and temperature limits should be properly setup prior to starting AutoTune.

Argument	Description
<i>temperature</i>	AutoTune test point, in °C

See the AutoTune section in the user's manual for further information.

**See Also** TEC:AUTOTUNE?

## TEC:AUTOTUNE?

**Synopsis** Query the AutoTune result

**Syntax** TEC:AUTOTUNE?

**Details** Returns the result of the last AutoTune process.

Response	Description
0	No AutoTune has been performed since last power-up
1	AutoTune in process
2	Last AutoTune failed
3	Last AutoTune successful

The TEC:AUTOTUNESTATE? provides additional details during the AutoTune process.

**See Also** TEC:AUTOTUNE, TEC:AUTOTUNESTATE?

## TEC:AUTOTUNESTATE?

**Synopsis** Query the AutoTune result

**Syntax** TEC:AUTOTUNESTATE?

**Details** Returns the progress of the AutoTune process.

Response	Description
0	No AutoTune in process
100	AutoTune initializing
200	AutoTune in P ramp phase
300	AutoTune in P stabilize phase
400	AutoTune in I ramp phase
500	AutoTune Complete

**See Also** TEC:AUTOTUNE

---

**TEC:CABLER****Synopsis** Set the cable resistance**Syntax** TEC:CABLER *resistance***Details** The TEC:CABLER command can be used to set the cable resistance, which is then used to remove voltage drops from the TEC voltage measurement.

Argument	Description
<i>resistance</i>	Resistance of the cable, in ohms.

See the user's manual for additional information on using this setting.

**See Also** TEC:CABLER?, TEC:VTE?

---

**TEC:CABLER?****Synopsis** Query the cable resistance value**Syntax** TEC:CABLER?**Details** Returns the value of the cable resistance value. See TEC:CABLER for a definition of the *resistance* response value.**See Also** TEC:CABLER

---

**TEC:CABLETYPE?****Synopsis** Returns the cable type (high or low current)**Syntax** TEC:CABLETYPE?**Details** Returns the current rating for the cable plugged into the instrument.

Response	Description
0	8A cable
1	10A cable

---

**TEC:COND?****Synopsis** Query TEC condition**Syntax** TEC:COND?**Details** Returns the TEC condition register.

Response	Bit	Value	Description
<i>conditions</i>	0	1	Current limit
	1	2	Voltage limit
	2	4	Sensor limit
	3	8	Temperature high limit
	4	16	Temperature low limit
	5	32	Sensor shorted
	6	64	Sensor open
	7	128	TEC open circuit
	8	256	Unused
	9	512	Out of tolerance
	10	1024	Output on
	11	2048	Unused
	12	4096	Unused
	13	8192	Unused
	14	16384	Unused
15	32768	Unused	

**See Also** TEC:ENABLE:COND, \*STB?

---

## TEC:CONST

**Synopsis** Set sensor temperature conversion constants

**Syntax** TEC:CONST A, B [, C [, R0]]

**Details** The TEC:CONST command sets the sensor constants for conversion of the sensor value to temperature. The number of parameters are dependent on the active sensor.

For the thermistor sensors, A, B, and C are used as constants for the Steinhart-Hart equation for conversion of resistance to temperature. By default, the thermistor constants are set to those for a BetaTHERM 10K3A1 thermistor.

For LM335 and AD590 sensors, A is used as a slope correction (M) term and B is an offset (B) term. By default, A is set to one and B is set to zero.

For RTD sensors, A, B, C, and R0 are used as constants for the RTD to temperature equation. By default, the RTD constants are set to the standard 100Ω Laboratory values.

Argument	Value	Description
<i>For Thermistors:</i>		
A	± 9.9999	First Steinhart-Hart constant (x 10 <sup>-3</sup> )
B	± 9.9999	Second Steinhart-Hart constant (x 10 <sup>-4</sup> )
C	± 9.9999	Third Steinhart-Hart constant (x 10 <sup>-7</sup> )
<i>For LM335:</i>		
A	± 9.9999	Slope term
B	± 99.9999	Offset term, in °C
<i>For AD590:</i>		
A	± 9.9999	Slope term
B	± 99.9999	Offset term, in °C
<i>For RTD:</i>		
A	± 9.9999	First RTD constant (x 10 <sup>-3</sup> )
B	± 9.9999	Second RTD constant (x 10 <sup>-6</sup> )
C	± 9.9999	Third RTD constant (x 10 <sup>-12</sup> )
R0	10 to 220	Nominal resistance at 0°C (Ω)

Some vendors may refer to A, B, and C as C1, C2, & C3.

For additional information, see the sensor sections in the user's manual.

**See Also** TEC:CONST?

---

## TEC:CONST?

**Synopsis** Query sensor temperature conversion constants

**Syntax** TEC:CONST?

**Details** Returns the sensor temperature conversion constants. See the TEC:CONST command for a complete definition of the A, B, C, and R0 response values.

**See Also** TEC:CONST

---

## TEC:DEC

**Synopsis** Decrement the TEC set point

**Syntax** TEC:DEC *steps*

**Details** The TEC:DEC command uses the step size defined with the TEC:STEP command to decrement the TEC set point. The set point is decremented *steps* times the step size.

Argument	Value	Description
<i>steps</i>	1 to 65000	Number of steps to decrement

**See Also** TEC:INC, TEC:STEP

---

## TEC:DISPlay

**Synopsis** Set the display enable state

**Syntax** TEC:DISPlay *enable*

**Details** The TEC:DISplay command can be used to completely lock out local operation of the instrument and display “Display Disabled” instead of the normal display.

Argument	Value	Description
<i>enable</i>	0	Disables the display and front panel
	1	Enables the display and front panel

Once the display is disabled, the front panel is completely locked out. The only way to restore functionality to the front panel is to issue a “TEC:DISplay 1” or cycle power on the unit.

**See Also** TEC:DISplay?

### TEC:DISplay?

**Synopsis** Query the display enable state

**Syntax** TEC:DISplay?

**Details** Returns the value of the TEC display enable state. See TEC:DISplay for a definition of the *enable* response value.

**See Also** TEC:DISplay

### TEC:ENABle:COND

**Synopsis** Set TEC Condition Enable register

**Syntax** TEC:ENABle:COND *conditions*

**Details** Enables reporting of selected conditions to the Status Byte Register. See the TEC:COND command for a definition of the *conditions* parameter. The default value for this register is 0.

**See Also** TEC:ENABle:COND?, TEC:COND?

### TEC:ENABle:COND?

**Synopsis** Query TEC Condition Enable register

**Syntax** TEC:ENABle:COND?

**Details** Returns the value of the TEC Condition Enable register. See the TEC:COND command for a definition of the *conditions* response.

**See Also** TEC:ENABle:COND, TEC:COND?

### TEC:ENABle:EVEnt

**Synopsis** Set TEC Event Enable register

**Syntax** TEC:ENABle:EVEnt *events*

**Details** Enables reporting of selected events to the Status Byte Register. See the TEC:EVEnt command for a definition of the *events* parameter. The default value for this register is 0.

**See Also** see also TEC:ENABLE:EVEnt?; TEC:EVE?

### TEC:ENABLE:EVEnt?

**Synopsis** Query TEC Event Enable register

**Syntax** TEC:ENABLE:EVEnt?

**Details** Returns the value of the TEC Event Enable register. See the TEC:EVEnt command for a definition of the *events* response.

**See Also** TEC:ENABLE:EVEnt, TEC:EVEnt?

### TEC:ENABLE:OUTOFF

**Synopsis** Set the Output Off Enable register

**Syntax** TEC:ENABLE:OUTOFF *outoff*

**Details** The Output Off register controls what conditions will cause the TEC output to be turned off. Some conditions are always enabled, as indicated below.

Argument	Bit	Value	Description
<i>outoff</i>	0	1	Current limit
	1	2	Voltage limit
	2	4	Sensor limit
	<b>3</b>	<b>8</b>	<b>Temperature high limit</b>
	<b>4</b>	<b>16</b>	<b>Temperature low limit</b>
	5	32	Unused
	<b>6</b>	<b>64</b>	<b>Sensor open</b>
	<b>7</b>	<b>128</b>	<b>TEC open circuit</b>
	<b>8</b>	<b>256</b>	<b>Sensor type change (<i>always enabled</i>)</b>
	9	512	Out of tolerance
	<b>10</b>	<b>1024</b>	<b>Sensor short</b>
	11	2048	Unused
	12	4096	Unused
	13	8192	Unused
	14	16384	Unused
	15	32768	Unused

The default value for this register is 1496, indicated by the bold elements above

**See Also** TEC:ENABLE:OUTOFF?

### TEC:ENABLE:OUTOFF?

**Synopsis** Query the Output Off Enable register

**Syntax** TEC:ENABLE:OUTOFF?

**Details** Returns the value of the Output Off register. See the TEC:ENABLE:OUTOFF command for definition of *outoff* response value.

**See Also** TEC:ENABLE:OUTOFF

---

**TEC:EVEnt?****Synopsis** Query the TEC event register**Syntax** TEC:EVEnt?**Details** Returns the TEC event register.

Response	Bit	Value	Description
<i>events</i>	0	1	Current limit
	1	2	Voltage limit
	2	4	Sensor limit
	3	8	Temperature high limit
	4	16	Temperature low limit
	5	32	Sensor shorted
	6	64	Sensor open
	7	128	TEC open circuit
	8	256	Sensor type change
	9	512	Out of tolerance changed state
	10	1024	Output changed state
	11	2048	Unused
	12	4096	Unused
	13	8192	Unused
	14	16384	Unused
15	32768	Unused	

After reading the event register, the event register is set to zero.

**See Also** TEC:ENABLE:EVEnt

---

**TEC:FAN****Synopsis** Set the external fan speed**Syntax** TEC:FAN *speed*[, *mode*[, *delay*]]**Details** Set the external fan speed.

Argument	Value	Description
<i>speed</i>	OFF, SLOW, MEDIUM, FAST, or 4.0 to 12.0	Fan speed
<i>mode</i>	1, 2, or 3	Fan delay mode: 1 = Auto (fan off when TEC off) 2 = On (fan always on) 3 = Delayed off (fan turns off <i>delay</i> minutes after TEC turned off).
<i>delay</i>	1 to 240	Minutes to delay turning fan off (only applies to 'Delayed off' <i>mode</i> ).

The values SLOW, MEDIUM, and FAST correspond to 9V, 10.5V, and 12V, respectively. The fan speed can also be controlled by selecting a specific voltage to drive the fan. When using external fan control, ensure your fan and voltage setting are properly match, or damage to the fan may occur.

Setting the *mode* to 1 will turn the fan off whenever the output is off. Setting the *mode* to 2 will turn the fan on always, regardless of the TEC output state. Setting the *mode* to 3 will delay the turning off of the fan to *delay* minutes after the TEC output is turned off.

The *delay* setting only applies when *mode* is set to 3 (delayed off), and sets the number of minutes to delay the fan from turning off after the output has been turned off.

The 5230 does not support the FAN setting.

**See Also** TEC:FAN?

### TEC:FAN?

**Synopsis** Query the external fan speed setting

**Syntax** TEC:FAN?

**Details** Returns the fan speed setting. See TEC:FAN for a definition of the *speed*, *mode*, and *delay* response values.

**See Also** TEC:FAN

### TEC:GAIN

**Synopsis** Set the control loop gain

**Syntax** TEC:GAIN *gain*

**Details** Set the temperature control loop gain.

Argument	Value	Description
<i>gain</i>	1, 3, 5, 10, 30, 50, 100, 300, PID	TEC gain

The 5230 does not support the PID setting.

**See Also** TEC:GAIN?; TEC:PID

### TEC:GAIN?

**Synopsis** Query the control loop gain

**Syntax** TEC:GAIN?

**Details** Returns the value of control loop gain. See TEC:GAIN for a definition of the *gain* response value.

**See Also** TEC:GAIN; TEC:PID

### TEC:HEATCOOL

**Synopsis** Set the heat/cool mode

**Syntax** TEC:HEATCOOL *mode*

**Details** Sets the heat/cool mode.

Argument	Value	Description
<i>mode</i>	BOTH	Heat and cool
	HEAT	Heat only
	COOL	Cool only

The 5230 does not support the HEATCOOL setting.

**See Also** TEC:HEATCOOL?

## TEC:HEATCOOL?

**Synopsis** Query the heat/cool mode

**Syntax** TEC:HEATCOOL?

**Details** Returns the heat/cool mode setting. See TEC:HEATCOOL for a definition of the *mode* response value.

**See Also** TEC:HEATCOOL

## TEC:INC

**Synopsis** Increment the TEC set point

**Syntax** TEC:INC *steps*

**Details** The TEC: INC command uses the step size defined with the TEC:STEP command to increment the TEC set point. The set point is incremented *steps* times the step size.

Argument	Value	Description
<i>steps</i>	1 to 65000	Number of steps to increment

**See Also** TEC:DEC, TEC:STEP

## TEC:ITE

**Synopsis** Set the TEC current set point

**Syntax** TEC:ITE *setpoint*

**Details** Sets the TEC current set point. It must be within the current limit.

Argument	Description
<i>setpoint</i>	Current set point, in amps

Not available in 5230 TECSOURCE.

**See Also** TEC:SET:ITE?, TEC:ITE?

## TEC:ITE?

**Synopsis** Query the TEC current

**Syntax** TEC:ITE?

**Details** The TEC:ITE? Query returns the measured TEC current.

Response	Value	Description
<i>ite</i>	$\pm\text{ITE}_{\text{im}}$	Measure TEC current, in amps

**See Also** TEC:LIMit:ITE

### TEC:INVERTITE

**Synopsis** Controls the ITE invert setting

**Syntax** TEC:INVERTITE *invert*

**Details** Changes (inverts) the polarity of the ITE current.

Argument	Value	Description
<i>invert</i>	0	Normal polarity
	1	Inverted polarity

**See Also** TEC:INVERTITE?

### TEC:INVERTITE?

**Synopsis** Query the state of the ITE inversion

**Syntax** TEC:INVERTITE?

**Details** Returns the ITE inversion setting. See TEC:INVERTITE for a definition of the *invert* response value.

**See Also** TEC:INVERTITE

### TEC:LIMit:ITE

**Synopsis** Set the TEC current limit

**Syntax** TEC:LIMit:ITE *limit*

**Details** Sets the TEC current limit.

Argument	Value	Description
<i>limit</i>	$\pm\text{ITE}_{\text{max}}$	TEC current limit, in amps

**See Also** TEC:ITE?, TEC:LIMit:ITE?

### TEC:LIMit:ITE?

**Synopsis** Query the TEC current limit

**Syntax** TEC:LIMit:ITE?

**Details** Returns the value of the TEC current limit. See TEC:LIMit:ITE for a definition of the *limit* response value.

**See Also** TEC:LIMit:ITE

---

**TEC:LIMit:RHI****Synopsis** Set the high sensor limit**Syntax** TEC:LIMit:RHI *limit***Details** Sets the high sensor limit.

Argument	Description
<i>limit</i>	High sensor limit: Thermistors in k $\Omega$ AD590s in $\mu$ A LM335s in mV RTDs in $\Omega$

**See Also** TEC:LIMit:RHI?

---

**TEC:LIMit:RHI?****Synopsis** Query the high sensor limit**Syntax** TEC:LIMit:RHI?**Details** Returns the value of the high sensor limit. See TEC:LIMit:RHI for a definition of the *limit* response value.**See Also** TEC:LIMit:RHI

---

**TEC:LIMit:RLO****Synopsis** Set the low sensor limit**Syntax** TEC:LIMit:RLO *limit***Details** Sets the low sensor limit.

Argument	Description
<i>limit</i>	Low sensor limit: Thermistors in k $\Omega$ AD590s in $\mu$ A LM335s in mV RTDs in $\Omega$

**See Also** TEC:LIMit:RLO?

---

**TEC:LIMit:RLO?****Synopsis** Query the low sensor limit**Syntax** TEC:LIMit:RLO?**Details** Returns the value of the low sensor limit. See TEC:LIMit:RLO for a definition of the *limit* response value.**See Also** TEC:LIMit:RLO

---

**TEC:LIMit:THI****Synopsis** Set the high temperature limit**Syntax** TEC:LIMit:THI *limit***Details** Sets the high temperature limit.

Argument	Value	Description
<i>limit</i>	-99 to +250	High temperature limit, in degrees

**See Also** TEC:LIMit:THI?

---

**TEC:LIMit:THI?****Synopsis** Query the high temperature limit**Syntax** TEC:LIMit:THI?**Details** Returns the value of the high temperature limit. See TEC:LIMit:THI for a definition of the *limit* response value.**See Also** TEC:LIMit:THI

---

**TEC:LIMit:TLO****Synopsis** Set the low temperature limit**Syntax** TEC:LIMit:TLO *limit***Details** Sets the low temperature limit.

Argument	Value	Description
<i>limit</i>	-99 to +250	Low temperature limit, in degrees

**See Also** TEC:LIMit:TLO?

---

**TEC:LIMit:TLO?****Synopsis** Query the low temperature limit**Syntax** TEC:LIMit:TLO?**Details** Returns the value of the low temperature limit. See TEC:LIMit:TLO for a definition of the *limit* response value.**See Also** TEC:LIMit:TLO

---

**TEC:MODE?****Synopsis** Query the TEC control mode**Syntax** TEC:MODE?**Details** Returns the TEC control mode.

Response	Value	Description
<i>mode</i>	T	Temperature control mode
	R	Sensor control mode
	ITE	Current control mode

**See Also** TEC:MODE:R, TEC:MODE:T; TEC:MODE:ITE

### TEC:MODE:ITE

**Synopsis** Set current control mode

**Syntax** TEC:MODE:ITE

**Details** Changes the set point to amps and sensor measurement to degrees Celsius.

Not available in 5230 TECSOURCE.

**See Also** TEC:MODE?, TEC:MODE:T;TEC:MODE:R

### TEC:MODE:R

**Synopsis** Set sensor control mode

**Syntax** TEC:MODE:R

**Details** Changes the set point and sensor measurement to ohms.

**See Also** TEC:MODE?, TEC:MODE:T, TEC:MODE:ITE

### TEC:MODE:T

**Synopsis** Set temperature control mode

**Syntax** TEC:MODE:T

**Details** Changes the set point and sensor measurement to degrees Celsius.

**See Also** TEC:MODE?, TEC:MODE:R, TEC:MODE:ITE

### TEC:MOUNT

**Synopsis** Set the mount type

**Syntax** TEC:MOUNT *mount*

**Details** Set the mount type to preset controller to mount's operating parameters.

Argument	Value	Description
<i>mount</i>	204, 214, 224, 242, USER	Mount type

Not available in 5230 TECSOURCE.

**See Also** TEC:MOUNT?

### TEC:MOUNT?

**Synopsis** Query the mount type

**Syntax** TEC:MOUNT?

**Details** Returns the mount type. See TEC:MOUNT for a definition of the *mount* response value.

**See Also** TEC:MOUNT

## TEC:OUTput

**Synopsis** Set the TEC output state

**Syntax** TEC:OUTput *state*

**Details** Turns the TEC output on or off.

Argument	Value	Description
<i>state</i>	0	Turn the output off
	1	Turn the output on

**See Also** TEC:OUTput?

## TEC:OUTput?

**Synopsis** Query the TEC output state

**Syntax** TEC:OUTput?

**Details** Returns the TEC output state. See TEC:OUT for a definition of the *state* response value.

**See Also** TEC:OUTput

## TEC:PID

**Synopsis** Sets the PID parameters

**Syntax** TEC:PID *p* [*i* [*d*]]

**Details** Sets PID parameters of the control loop when the GAIN is set to PID.

Argument	Description
<i>p</i>	The Proportional term, can be from 0 to 10
<i>i</i>	The Integral term, can be from 0 to 10
<i>d</i>	The Derivative term, can be from 0 to 10

The *i* and *d* values can be omitted, such that value commands might be:

TEC:PID 1

TEC:PID 1,2

TEC:PID 1,2,3

But the following is not valid:

TEC:PID 1,,3

**See Also** TEC:PID?, TEC:GAIN

## TEC:PID?

**Synopsis** Queries the TEC PID parameters

**Syntax** TEC:PID?

**Details** Returns the TEC PID parameters used when GAIN is set to PID.

**See Also** TEC:PID, TEC:GAIN

## TEC:R

**Synopsis** Set the sensor set point

**Syntax** TEC:R *setpoint*

**Details** Sets the sensor set point. It must be within the low and high sensor limits.

Argument	Description
<i>setpoint</i>	Sensor set point: Thermistors in k $\Omega$ AD590s in $\mu$ A LM335s in mV RTDs in $\Omega$

**See Also** TEC:SET:R?

## TEC:R?

**Synopsis** Query the actual sensor value

**Syntax** TEC:R?

**Details** Returns the actual (measured) sensor value.

Response	Description
<i>sensor value</i>	Actual sensor value: Thermistors in k $\Omega$ AD590s in $\mu$ A LM335s in mV RTDs in $\Omega$

**See Also** TEC:R

## TEC:SENsOr

**Synopsis** Set the sensor type

**Syntax** TEC:SENsOr *sensor*

**Details** Sets the sensor type.

Argument	Value	Description
<i>sensor</i>	1	100uA Thermistor
	2	10uA Thermistor
	3	LM335
	4	AD590
	5	RTD
	6	RTD (4-wire sense)
	7	1mA Thermistor

The 5230 only supports the 10uA Thermistor setting (TEC:SENS 2). Any other value will generate an error.

**See Also** TEC:SENsor?

### TEC:SENsor?

**Synopsis** Query the sensor type

**Syntax** TEC:SENsor?

**Details** Returns the active sensor. See the TEC:SENsor command for a definition of the *sensor* response.

**See Also** TEC:SENsor

### TEC:SET:ITE?

**Synopsis** Query current set point

**Syntax** TEC:SET:ITE?

**Details** Returns the current set point. See TEC:ITE for a definition of the *setpoint* response value.  
Not available in 5230 TECSOURCE.

**See Also** TEC:ITE

### TEC:SET:R?

**Synopsis** Query sensor set point

**Syntax** TEC:SET:R?

**Details** Returns the sensor set point. See TEC:R for a definition of the *setpoint* response value.

**See Also** TEC:R

### TEC:SET:T?

**Synopsis** Query temperature set point

**Syntax** TEC:SET:T?

**Details** Returns the temperature set point. See TEC:T for a definition of the *setpoint* response value.

**See Also** TEC:T

### TEC:STB?

**Synopsis** Query the TEC status byte

**Syntax** TEC:STB?

**Details** Returns a summary of the enabled conditions within the TEC condition and event registers. These bits mirror the bits in the Status Byte Register.

Response	Bit	Value	Description
<i>status</i>	0	1	Event status register summary
	1	2	Condition status register summary

The values are additive, so a return value of 0, 1, 2, or 3 is possible.

**See Also** \*STB?, TEC:COND?, TEC:ENAB:COND, TEC:ENABLE:EVENT, TEC:EVENT?

### TEC:STEP

**Synopsis** Set TEC step size

**Syntax** TEC:STEP *size*

**Details** The command sets the TEC step size used by the TEC:DEC or TEC:INC commands.

Argument	Value	Description
<i>size</i>	1 to 65000	Step size

The *size* value corresponds to the resolution of the set point, regardless of control mode or sensor. For example, if the temperature resolution is 0.01°C, then a step of 1 would mean a change of 0.01°C. Likewise, for example, RTDs, which typically have a display resolution of 0.01Ω, a step of one would mean a change of 0.01Ω.

**See Also** TEC:DEC, TEC:INC, TEC:STEP?

### TEC:STEP?

**Synopsis** Query TEC step size

**Syntax** TEC:STEP?

**Details** Returns the TEC step size. See TEC:STEP for a definition of the *size* response value.

**See Also** TEC:STEP

### TEC:T

**Synopsis** Set the temperature set point

**Syntax** TEC:T *setpoint*

**Details** Sets the temperature set point. It must be within the low and high temperature limits.

Argument	Description
<i>setpoint</i>	Temperature set point, in degrees Celsius

**See Also** TEC:SET:T?

## TEC:T?

**Synopsis** Query the actual temperature

**Syntax** TEC:T?

**Details** Returns the actual (measured) sensor temperature.

Response	Description
<i>temperature</i>	Actual sensor temperature, in degrees Celsius

**See Also** TEC:T

## TEC:TOLerance

**Synopsis** Set the TEC tolerance criteria

**Syntax** TEC:TOLerance *tolerance, time*

**Details** The TEC:TOLerance command allows control over when the output of the temperature controller is considered in tolerance (or stable), in order to satisfy the tolerance condition of the operation complete definition. When used in conjunction with the \*WAI command, it can control when the next command is processed, delaying processing until the output stabilizes at its set point.

Argument	Value	Description
<i>tolerance</i>	0.01 to 10	Tolerance, in °C
<i>time</i>	0.1 to 50	Time window in seconds

To be considered in tolerance, the measured temperature must be within the set point plus or minus the *tolerance* value (the tolerance window) for *time* seconds. Any time it leaves the tolerance window, the timer will reset to zero and begin counting the next time it enters the tolerance window.

In sensor control mode, the *tolerance* is fixed at 0.01kΩ (100μA thermistor), 0.1kΩ (10μA thermistor), 0.1μA (AD590), 1mV (LM335), and 0.1Ω (RTD).

**See Also** TEC:TOLerance?

## TEC:TOLerance?

**Synopsis** Query the TEC tolerance criteria

**Syntax** TEC:TOLerance?

**Details** Returns the value of the TEC tolerance criteria. See TEC:TOLerance for a definition of the *tolerance* and *time* response values.

**See Also** TEC:TOLerance

---

**TEC:USERCAL:GET?****Synopsis** Query a TEC USERCAL setting**Syntax** TEC:USERCAL:GET? *index***Details** Returns the slope and offset compensation values for a specific user calibration *index*. See TEC:USERCAL:PUT for a definition of the *index* argument and *slope* and *offset* response values.**See Also** TEC:USERCAL:PUT

---

**TEC:USERCAL:EDIT****Synopsis** Enable or disabled user calibration editing**Syntax** TEC:USERCAL:EDIT *enable***Details** Sets the edit enable state for user calibration. A TEC:USERCAL:PUT command will fail until the editing is enabled with this command.

Argument	Value	Description
<i>enable</i>	0	User calibration editing disabled
	1	User calibration editing enabled

**See Also** TEC:USERCAL:EDIT?, TEC:USERCAL:PUT

---

**TEC:USERCAL:EDIT?****Synopsis** Query the state of the user calibration edit enable flag**Syntax** TEC:USERCAL:EDIT?**Details** Returns 0 if user calibration editing is disabled, and 1 if user calibration editing is enabled.**See Also** TEC:USERCAL:EDIT

---

**TEC:USERCAL:PUT****Synopsis** Sets a USERCAL value**Syntax** TEC:USERCAL:PUT *index, slope, offset***Details** The command sets the user calibration setting for a specific *index*, allowing for user compensation of measurements or set points. The default value for all slopes is 1, and the default value for all offsets is 0.

Compensation is applied according to the following formula:

$$\text{compensated} = \text{slope} * \text{uncompensated} + \text{offset}$$

Argument	Value	Description
<i>index</i>	1	ITE set point
	2	ITE measurement
	3	VTE measurement
	4	Th 10 $\mu$ A measurement
	5	Th 100 $\mu$ A measurement
	7	LM335 measurement
	8	AD950 measurement
	9	RTD measurement
	10	RTD 4-wire measurement
	<i>slope</i>	0.9 to 1.1
<i>offset</i>	$\pm 5\%$ of range	Offset compensation

TEC:USERCAL:PUT is only supported on certain instruments. Not all indexes are supported in every instrument. See your user's manual for details on what measurements are supported.

You must enable editing of user calibration values with the TEC:USERCAL:EDIT command.

**See Also** TEC:USERCAL:EDIT, TEC:USERCAL:GET?

## TEC:V?

**Synopsis** Query the actual TEC voltage

**Syntax** TEC:V?

**Details** Returns the actual (measured) TEC voltage.

Response	Description
<i>voltage</i>	Actual TEC voltage, in volts

**See Also** TEC:T

## TERM

**Synopsis** Set response terminator

**Syntax** TERM *terminator*

**Details** This command controls the termination characters used for responses to queries.

Argument	Value	Description
<i>terminator</i>	0 or 1	<CR> <LF>
	2 or 3	<CR>
	4 or 5	<LF>
	6 or 7	no terminator

**See Also** TERM?

## TERM?

**Synopsis** Query response terminator

**Syntax** TERM?

**Details** Returns the current response terminator setting. See the TERM command for a complete definition of possible return values.

**See Also** TERM

## TERMINAL

**Synopsis** Set terminal mode

**Syntax** TERMINAL *enable*

**Details** The command controls the echo of characters back to the PC, typically used when manually controlling the instrument via a terminal software package such as Hyperterminal.

Argument	Value	Description
<i>enable</i>	0	Echo disabled
	1	Echo enabled

**See Also** TERMINAL?

## TERMINAL?

**Synopsis** Query terminal mode

**Syntax** TERMINAL?

**Details** Returns the current response terminal mode setting. See the TERMINAL command for a complete definition of possible return values.

**See Also** TERMINAL

## TIME?

**Synopsis** Query run time

**Syntax** TIME?

**Details** Returns the elapsed time since the unit has been turned on. Format is in HH:MM:SS.ss, where HH is hours, MM is minutes, SS is seconds, and ss is hundredths of a second.

**See Also** TIMER?

## TIMER?

**Synopsis** Query time since last TIMER?

**Syntax** TIMER?

**Details** Returns the elapsed time since the last TIMER? query was received, or, if this is the first TIMER? query, the time since unit has been turned on. Format is in HH:MM:SS.ss, where HH is hours, MM is minutes, SS is seconds, and ss is hundredths of a second.

**See Also** TIME?

---

**VER?**

**Synopsis** Query the firmware version

**Syntax** VER?

**Details** Returns the firmware version. This is the same information that is part of the \*IDN? query.

**See Also** \*IDN?

## Error Messages

Error Code	Description	Cause
E-003	Factory EEPROM Error	The factory section of the EEPROM, which contains the calibration data and system configuration systems, is corrupted. The unit must be returned to the factory for repair.
E-004	User EEPROM Error	The user settings section of the EEPROM, which contains user editable set points, modes, etc., is corrupted. To repair the corruption, the unit was reset to factory defaults.
E-005	User Reset EEPROM	The user reset the unit to factory defaults. This is a notification message only.
E-006	User EEPROM Failed	The user settings section of the EEPROM, which contains user editable set points, modes, etc., is corrupted. The corruption cannot be repaired automatically, please contact the factory for assistance.
E-007	Preset EEPROM Failed	The preset settings section of the EEPROM, which contains user section defaults, is corrupted. The corruption cannot be repaired automatically, please contact the factory for assistance.
E-008	User EEPROM Failed	The user settings section of the EEPROM, which contains user editable set points, modes, etc., is corrupted, and the preset section cannot restore it. The corruption cannot be repaired automatically, please contact the factory for assistance.
E-009	Output Disabled	Added to every ERR? or ERRSTR? query when the EEPROM is corrupted. The output cannot be turned on until the corruption has been repaired.
E-100	General Error	The error code is non-specific, and is generally used when no other error code is suitable.
E-102	Message too long	The message is too long to process (USB/Serial only).
E-104	Type not allowed	The RADix type was invalid
E-123	Path not found	The message used an invalid path command (USB/Serial only).
E-124	Data mismatch	The message contained data that did not match the expected format (USB/Serial only).
E-126	Too few or too many elements	The command requires more or less than the number of parameters actually supplied.
E-127	Change not allowed	An attempt was made to change a parameter that cannot be changed, or is currently read-only.
E-201	Data out of range	The message attempted to set a value that was outside the allowable range (USB/Serial only).
E-202	Invalid data type	When trying to parse the message, the data was in an invalid format (USB/Serial only).
E-204	Suffix not valid	An invalid number base suffix (radix) was encountered when parsing a number (USB/Serial only).
E-217	Configuration Recall failed	An attempt recall a configuration failed. This can be caused if no configuration exists in the selected slot, the slot number is out of range, or if the configuration is corrupted.
E-218	Configuration Save failed	At attempt to save a configuration failed. This can be caused if the slot number is out of range, or the configuration memory is corrupted.
E-220	Script Save Failed	At attempt to save a script failed. This can be caused if the script number is out of range, or the script memory is corrupted.
E-221	Cannot embed script	A script was executed that contained a reference to another script.
E-222	Cannot execute script	At attempt to save a script failed. This can be caused if the script number is out of range, no script exists for the selected index, or the script memory is corrupted.
E-303	Input buffer overrun	The command input buffer overran, which can be caused by excessively long command strings or improperly terminated commands.
E-402	Sensor open, output turned off	A sensor open circuit was detected and the output was turned off.

Error Code	Description	Cause
E-403	Module open, output turned off	A Peltier module open circuit was detected and the output was turned off.
E-404	I limit, output turned off	A current limit was detected and the output was turned off.
E-405	V limit, output turned off	A voltage limit was detected and the output was turned off.
E-406	Thermistor resistance limit, output turned off	The thermistor resistance limit (high or low) was exceeded and the output was turned off.
E-407	Temperature limit, output turned off	The temperature limit (high or low) was exceeded and the output was turned off.
E-409	Sensor change, output off	The sensor was changed while the output was on, forcing the output off.
E-410	Temperature was out of tolerance, output turned off	The temperature went out of tolerance and the output was turned off.
E-415	Sensor short, output turned off	A sensor short circuit was detected and the output was turned off.
E-419	TEC not stable	The TEC is considered stable if the temperature has changed less than 0.02°C for more than 20 seconds.
E-433	Not a TEC	The TEC:CHAN command attempted to select a non-TEC channel
E-434	Ite limit exceeds cable rating	The cable plugged into the unit cannot carry the current as limited by the Ite Limit setting. Lower the Ite limit to the cables capacity, or use a higher capacity cable.
E-435	Mode change	A mode change occurred when the output was on, forcing the output off.
E-436	AutoTune Failed	The AutoTune process failed.
E-437	AutoTune Required T Mode	The AutoTune process was cancelled because the instrument was not in T mode.
E-450	TECPak analog set disconnected	An attempt to turn on the TEC output was prevented because the instrument was configured to use the analog set point input and no analog set point was detected.
E-501	Interlock shutdown output	The interlock was open when the output was on (or was attempting to turn on).
E-504	Laser current limit disabled output.	The laser output was turned off because a current limit was detected and the corresponding bit in the OUTOFF register was set.
E-505	Laser voltage limit disabled output	The laser voltage exceeded the voltage limit and the output was turned off.
E-506	Laser photodiode current limit disabled output	The laser output was turned off because a photodiode current limit was detected and the corresponding bit in the OUTOFF register was set.
E-507	Laser photodiode power limit disabled output	The laser output was turned off because a photodiode power limit was detected and the corresponding bit in the OUTOFF register was set.
E-508	TEC off disabled output	The laser output was turned off because the TEC was off and the corresponding bit in the OUTOFF register was set.
E-509	Laser short circuit disabled output	The laser output was turned off because a short condition was detected and the corresponding bit in the OUTOFF register was set.
E-510	Laser out of tolerance disabled output	The laser output was turned off because an out-of-tolerance condition was detected and the corresponding bit in the OUTOFF register was set.
E-511	Laser control error disabled output	A hardware control error was detected which forced a shutdown of the laser output.
E-512	Power failure	A power failure was detected.
E-514	Laser mode change disabled output	A change in the operating mode of the Laser driver while the output was on shutdown the output.
E-516	Incorrect configuration for calibration to start	The Laser driver was not configured properly, including the mode and output on state, to be able to start the desired calibration process.
E-517	Calibration must have the output on to start	The laser output must be on for the calibration process to start.
E-521	TEC temperature limit disabled	The laser output was turned off because the TEC temperature limit

Error Code	Description	Cause
	output	was exceeded and the corresponding bit in the OUTOFF register was set.
E-534	Po mode selected with PD Response set to zero	Attempted to select Po mode and PD Response was zero, or Laser driver was in Po mode and PD Response was set to zero.
E-535	Calibration cancelled	The active calibration process was cancelled.
E-536	Intermittent contact detected	The instrument detected an intermittent contact and shut down the laser output. If this is triggering falsely (such as in a noisy environment), the intermittent contact detection can be disabled in the main menu.
E-537	Thermal Limit Exceeded	The thermal load inside the instrument is too high, and the output was shutdown to protect the instrument.
W-800	Remote Voltage Sense Low	On instruments with remote voltage sense, the remote voltage measurement is much lower than the voltage at the output connector. This is only a warning, and does not indicate an actual problem.
W-801	Burst Mode, Hold Output	When in lo (Burst) mode, to turn the output on, the Output button must be held down for at least one second. If it is held down for less than one second, this warning message informs the user than the Output button press did not turn the output on.
E-997	Control error, cycle power	A hardware control error occurred, cycle power to resolve. If error continues to occur, contact factory.
E-998	Command not supported	A command was recognized but not supported by the Laser driver.
E-999	Non-specific error	A non-specific error was encountered.



624 Clarion Court, San Luis Obispo, CA 93401

Tel: (805) 543-1302 Fax: (805) 543-1303

**[sales@arroyoinstruments.com](mailto:sales@arroyoinstruments.com)**

**[www.arroyoinstruments.com](http://www.arroyoinstruments.com)**