Appendix C
Remote Operation (Command Mode)

Rev 1.01

C.1 Command Interface

All communications with the XL™ are through a command interface using the RS-232 Serial Port. Commands are used to setup, monitor and control the XL™. To access the command interface, the XL™ must be connected to a computer that is running some terminal program. The terminal program must use the following terminal and communication settings:

BAUD RATE:	9600
DATA BITS:	8
STOP BITS:	1
PARITY:	N
DUPLEX:	FULL
TERMINAL EMULATION:	VT52

When the computer is connected and ready to communicate, pressing the ENTER key will cause the command interface prompt to be displayed. This prompt is an arrow **"->"** on the left side of the screen. It may take a couple of key presses to "wake up" the XL™. Once the prompt is displayed the XL™ is ready to receive commands. If no commands are received in a two-minute time period the XL™ will turn off automatically, returning to a low power mode.

The command interface is best suited for communications with other computer equipment. The command interface allows a user to easily develop his/her own custom interface, or for RS-232 devices to access the XL™ inputs and outputs. For human interaction with the XL™ the menu interface is normally used as discussed in chapter 3. This chapter will give a detailed description of the commands used by the command interface.

All commands must be terminated with a carriage return and a line feed character. The "↵" character in the examples is used to represent the ENTER key and signifies a carriage return and a line feed being sent to the XL™. Any parameters must follow the command as described with the proper spacing between the command and the parameter. The command may be in upper or lower case. When entering the command mode the characters typed on the host system will be received by the XL™ and echoed back to the host system. This allows the user to verify that what was typed was also received by the XL™. The BACKSPACE key may be used to edit characters on the command line. Echoing the command may be turned off using the 'ECHO=OFF' command. The commands listed below are grouped into categories based on the function of the command.

The XL™ has three basic types of commands:

Read Commands: The command to read each value consists of the value name, followed by "?". For example, to read the time, the command is "TIME?". The XL™ response will be similar to "TIME = 12:34:56".

Write commands: The command to write each value consists of the value name, followed by "=", and then the new value. For example, to set the date, the command is "DATE=05/05/99". There is no response to a write command.

Action Commands: These simply consist of the command. For example, "RESETSTATUSVALUES", or "PURGENOW".

Below is a table of commands. In the table, commands followed by "?" are values that can be read, commands followed by "=" are values that can be written, and commands followed by "?/=" are values that can be read and written. Action commands have no special character following.

SYSTEM COMMANDS	
COMMAND	**DESCRIPTION**
TIME?/=	Current time (normally set to local time)
DATE?/=	Current date (normally set to local date)
SITEID?/=	Site ID
SDIADDRESS?/=	SDI-12 address
BAUDRATE?/=	Baud rate of the communications port
FLOWCONTROL?/=	Hardware or software flow control (or none)
CHARACTERDELAY?/=	Delay between characters out the comm port
LINEDELAY?/=	Delay after line-feeds out the comm port
RESETDEFAULTS	Reset settings to system defaults

STATUS COMMANDS	
FIRMWAREVERSION?	XL™ firmware version
CHECKSUM?	Diagnostic to verify firmware integrity
BATTERY?	Current battery voltage
BATTMIN?	Lowest measured battery voltage
BATTMAX?	Highest measured battery voltage
SYSTEMRESETS?	Number of system resets
RESETSTATUSVALUES	Resets BattMin, BattMax, and SystemResets
STATUS1-5SOURCE?/=	Source for the status value
STATUS1-5LABEL?/=	Label for the status value
STATUS1-5VALUE?	Value of the corresponding status source

DATA FILE COMMANDS	
ERASEDATACARD	Erase the data card in the external slot
ERASEINTERNALDATA	Erase data from internal memory
DIRDATACARD	Directory of data card
DIRINTERNALDATA	Directory of internal data
DUMPDATACARDFILE <*filename*>	Transmit the file out the serial port
DUMPINTERNALDATAFILE <*name*>	Transmit the file out the serial port
COPYDATATOCARD	Copy the internal data files to the data card
DATACARDBYTESUSED?	Number of bytes used on the data card
DATACARDBYTESFREE?	Number of bytes free on the data card
INTERNALDATABYTESUSED?	Number of bytes used in internal memory
INTERNALDATABYTESFREE?	Number of bytes free in internal memory

SCANNING COMMANDS	
SCANNING?/=	Scanning on or off
SCANRATE?/=	Amount of time between scans
NEXTSCAN?	Amount of time until next scan
NEXTSCANAT?/=	Time of next scan

SENSOR INPUT COMMANDS	
STAGE?/=	Current stage level
LASTSTAGE?	Stage level at time of last scan
STAGEUNITS?/=	Unit of measurement for stage (feet, meters, ...)
STAGEOFFSET?/=	Offset for stage
TEMPERATURE?	Current temperature value
TEMPERATUREUNITS?/=	Unit of measurement for temperature (C, F)
COUNTERTOTAL?/=	Total counts since last reset
COUNTERSUM?/=	Counts since last interval
COUNTERSLOPE?/=	Counter slope
SUMINTERVAL?/=	Interval to reset COUNTERSUM
COUNTERMODE?/=	Counter mode enabled/disabled
RESETCOUNTERS	Reset all of the counters
ANALOG1-4?	Analog sensor value
ANALOG1-4SLOPE?/=	Analog sensor slope
ANALOG1-4OFFSET?/=	Analog sensor offset

DIGITAL I/O COMMANDS	
DIGITAL1-2MODE?/=	Digital mode (input, pulse, continuous, ...)
DIGITAL1-2TRIGGERSOURCE?/=	Source of value to evaluate for alarm conditions
DIGITAL1-2TRIGGERTYPE?/=	Trigger type (always, > set point, ...)
DIGITAL1-2TRIGGERPOINT?/=	Point with which the source value is compared
DIGITAL1-2PULSELENGTH?/=	Length in ms of output pulse
DIGITAL1-2WARMUPDELAY?/=	Length in ms of warm-up delay

SDI-12 COMMANDS	
SDITESTADDRESS?/=	Address used when sending SDI-12 commands
SDIACKNOWLEDGE	Send an acknowledge command
SDIIDENTIFY	Send an identify command
SDIVERIFY	Send a verify command
SDIMEASURE	Send a measure command
SDIDATAn	Send a collect data command
SDIEXTENDED=<*command*>	Send a custom extended command

OUTPUT COMMANDS	
AUTOPRINT?/=	Data being logged also gets sent out serial port
REMOTEDISPLAYLABEL?/=	Label for the data going to the remote display
REMOTEDISPLAYSOURCE?/=	Source of data going to the remote display
EMAIL?/=	E-mail modem on/off
EMAILSENDRATE?/=	E-mail modem send rate
EMAILSENDHEADERS?/=	Send headers with e-mail data (yes/no)
EMAILSENDSITEID?/=	Send Site ID with e-mail data (yes/no)
EMAILALARMS?/=	Send e-mail based on alarm conditions (yes/no)
EMAILTRIGGERSOURCE?/=	Source of value to evaluate for alarm conditions
EMAILTRIGGERMODE?/=	Trigger mode (always, delta, ...)
EMAILTRIGGERPOINT?/=	Point with which the source value is compared
ALERT?/=	ALERT radio on/off
ALERTRADIOTYPE?/=	Type of ALERT radio
ALERT1-5SOURCE?/=	Source for the ALERT radio
ALERT1-5ID?/=	ALERT radio ID
ALERT1-5MAXRANGE?/=	Maximum range for this datum
ALERT1-5MINRANGE?/=	Minimum range for this datum
ALERT1-5TRIGGERMODE?/=	Trigger mode (always, delta, ...)
ALERT1-5TRIGGERPOINT?/=	Trigger point
ALERT1-5HOLDOFF?/=	Delay after transmitting
ALERT1-5OVERRIDE?/=	Override
ALERT1-5FIXEDTXDRATE?/=	Fixed Txd Rate for the ALERT radio
ALERT1-5FORCETRANSMISSION	Force the ALERT radio to transmit

LOGGING COMMANDS	
LOGGING?/=	Log data to internal memory
LOGCOLUMN1-25SOURCE?/=	Source of flash card output column
LOGCOLUMN1-25HEADER?/=	Column header for flash card output

LOGCOLUMN1-25SKIPRATE?/=	Skip rate for flash output column
LOGCOLUMN1-25DIGITS?/=	# of digits to print for flash column

GOES COMMANDS	
GOESMODE?/=	Off, Timed, Random, Both
GOESADDRESS?/=	8 character address, assigned by NESDIS
GOESTIME?/=	International standard time
GOESSTCHANNEL?/=	Self timed channel number
GOESSTRATE?/=	Self Timed transmit rate
GOESSTOFFSET?/=	Offset from midnight to first transmit
GOESSTFORMAT?/=	Format of Self Timed data (SHEF, binary, etc.)
GOESSTORDER?/=	Order of Self Timed data
GOESSTSENDSHEFCODES?/=	Send SHEF codes with each transmit (yes, no)
GOESSTAPPENDBATTERY?/=	Send 1 battery reading at end of transmission
GOESSTSCANSPERXMIT?/=	# of scans per transmission
GOESSTBUFFERCOUNT?/=	# of bytes in transmit buffer
GOESSTCOLUMN1-20SOURCE?/=	Source of ST output column
GOESSTCOLUMN1-20SHEFCODE?/=	SHEF code of ST output column
GOESSTCOLUMN1-20SKIPRATE?/=	Skip rate of ST output column
GOESRRCHANNEL?/=	Random Report channel number
GOESRRRATE?/=	Random Report transmit rate
GOESRRTRIGGERMODE?/=	Trigger mode (always, delta, ...)
GOESRRTRIGGERSOURCE?/=	Trigger source
GOESRRTRIGGERPOINT?/=	Trigger point
GOESRRFORCEXMIT	Force a random transmission
GOESRRBUFFERCOUNT?	# of bytes in RR transmit buffer
GOESRRCOLUMN1-5SOURCE?/=	Source of RR output column

BUBBLER COMMANDS	
BUBBLERATE?/=	Bubbles per minute
PURGEPSI?/=	Pressure with which the bubbler purges
PURGETHRESHOLD?	Pressure at which the bubbler purges
PURGEDURATION?/=	Duration of purge
TANKPSI?	Pressure in the tank
LINEPSI?	Pressure in the line
PURGENOW	Perform an immediate purge
AUTOPURGE?/=	Enable/Disable auto purge function
TIMEDPURGE?/=	Enable/Disable timed purge function
LASTPURGE?	Time of last purge

C.2 System Commands

Command: TIME?

Description: Read the current time.

Parameters: None.

Response: An ASCII text string is returned with the time formatted as hh:mm:ss. The response is terminated using a carriage return, line feed.

Example:

```
-> TIME?↵                    Command sent to the XL™.
Time = 15:37:21␍␊            Response sent to the host system.
```

Command: TIME=hh:mm:ss

Description: Set the current time in the real time clock. The new time is tested for proper values and is then used to update the real time clock.

Parameters: This command expects an input of the format hh:mm:ss, where h is hours, m is minutes and s is seconds.

Response: None, use the TIME? command to see the new value.

Example: `-> TIME=13:53:45↵` Command sent to the XL™.
This sets the current time to 1PM, 53 minutes, and 45 seconds.

Command: DATE?

Description: Read the current date.

Parameters: None.

Response: An ASCII text string is returned with the date formatted as mm/dd/yy. The response is terminated using a carriage return, line feed.

Example: `-> DATE?↵` Command sent to the XL™.
`Date = 03/21/06ᶜᴿᴸᶠ` Response sent to the host system.

Command: DATE=mm/dd/yy

Description: Set the present date in the real time clock. The input is tested for proper values and is then used to update the real time clock.

Parameters:This command expects an input of the format mm/dd/yy, where m is the month, d m is the day, and y is the year.

Response: None, use the DATE? command to see the new value.

Example: `-> DATE=03/21/06↵` Command sent to the XL™.
This sets the current date to March 21st, 2006.

Command: SITEID?

Description: Read the Site ID. The Site ID is the text description of the site. It is 8 characters in length.

Parameters: None.

Response: An ASCII text string is returned with the Site ID formatted as XXXXXXXX. The response is terminated using a carriage return, line feed.

Example: `-> SITEID?↵` Command sent to the XL™.
`SiteID = RIVER1 ᶜᴿᴸᶠ` Response sent to the host system.

Command: SITEID=XXXXXXXX

Description: Set the site ID. The Site ID is the text description of the site. It is 8 characters in length.

Parameters: The site ID gets assigned to the entered text. If the text is shorter than 8 characters, the site ID will be padded with spaces.

Response: None, use the SITEID? command to see the new value.

Example: `-> SITEID=Site-15↵` Command sent to the XL™.
This sets the site ID to "Site-15 ".

Command: SDIADDRESS?

Description: Read the SDI-12 sensor address. This address is how the XL™ is identified when used as a sensor.

Parameters: None.

Response: An ASCII text string is returned with the SDI-12 address. The address is a single character. The response is terminated using a carriage return, line feed.

Example: `-> SDIADDRESS?↵` Command sent to the XL™.
`SDIAddress = 1`  Response sent to the host system.

Command: SDIADDRESS=X

Description: Set the SDI-12 sensor address. This address is how the XL™ is identified when used as a sensor.

Parameters: The SDI-12 address gets assigned to the entered character. The valid range for addresses is 0-9 and A-Z.

Response: None, use the SDIADDRESS? command to see the new value.

Example: `-> SDIADDRESS=0↵` Command sent to the XL™.
This sets the SDI-12 address to 0.

Command: BAUDRATE?

Description: Read the baud rate for the serial communications port.

Parameters: None.

Response: An ASCII text string is returned with the baud rate. The baud rate can be " 300", "1200", "2400", "4800", "9600", or "19.2". The response is terminated using a carriage return, line feed.

Example:

`-> BAUDRATE?↵`	Command sent to the XL™.
`Baud Rate = 9600␍␊`	Response sent to the host system.

Command: BAUDRATE=XXXX

Description: Set the baud rate for the serial communications port.

Parameters: Any of the following are valid baud rate parameters: " 300", "1200", "2400", "4800", "9600", and "19.2".

Response: None, use the **BAUDRATE**? command to see the new value.

Example:

`-> BAUDRATE=1200↵`	Command sent to the XL™.
`␍␊`	Response from the XL™.

Command: FLOWCONTROL?

Description: Read the flow control setting for the serial communications port.

Parameters: None.

Response: An ASCII text string is returned with the flow control. The flow control can be "None", "HW", "SW", or "Both". The response is terminated using a carriage return, line feed.

Example:

`-> FLOWCONTROL?↵`	Command sent to the XL™.
`Flow Control = None␍␊`	Response sent to the host system.

Command: FLOWCONTROL=XXXX

Description: Set the flow control for the serial communications port.

Parameters: There are 4 valid flow control settings, "None", "HW", "SW", and "Both".

Response: None, use the **FLOWCONTROL**? command to see the new value.

Example: `-> FLOWCONTROL=HW↵` Command sent to the XL™.
`┘R└F` Response from the XL™.

Command: CHARACTERDELAY?

Description: Read the delay that the XL™ puts between sending characters out the serial communications port.

Parameters: None.

Response: An ASCII text string is returned with the character delay. The character delay is and integer between 0 and 999. The response is terminated using a carriage return, line feed.

Example: `-> CHARACTERDELAY?↵` Command sent to the XL™.
`Character Delay = 0┘R└F` Response sent to the host system.

Command: CHARACTERDELAY=XXX

Description: Set the character delay. The character delay is how long the XL™ delays in milliseconds between sending characters out the serial communications port.

Parameters: This command accepts an integer between 0 and 999.

Response: None, use the **CHARACTERDELAY**? command to see the new value.

Example: `-> CHARACTERDELAY=50↵` Command sent to the XL™.
`┘R└F` Response from the XL™.

Command: LINEFEEDELAY?

Description: Read the line feed delay. The XL™ waits this amount of time in milliseconds after each line feed it sends out the serial communications port.

Parameters: None.

Response: An ASCII text string is returned with the line feed delay. The line feed delay is an integer value. The response is terminated using a carriage return, line feed.

Example: `-> LINEFEEDDELAY?↵` Command sent to the XL™.
`Line Feed Delay = 0┘R└F` Response sent to the host system.

Command: LINEFEEDDELAY=XXX

Description: Set the character delay. The character delay is how long the XL™ delays in milliseconds between sending characters out the serial communications port.

Parameters: There are 4 valid flow control settings, “None”, “HW”, “SW”, and “Both”.

Response: None, use the **LINEFEEDDELAY**? command to see the new value.

Example:

```
-> LINEFEEDDELAY=50↵                Command sent to the XL™.
CRLF                                Response from the XL™.
```

Command: RESETDEFAULTS

Description: This command is used to force the unit to run through the power up sequence. This will cause the unit to reinitialize some of the system variables, initialize the real time clock and then turn off. The ENTER key must be pressed a few times again to wake up the unit. This has the same effect as if the power is removed and then restored.

Parameters: None.

Response: None.

Example:

```
-> RESETDEFAULTS↵                   Command sent to the XL™
```

C.3 Status Commands

Command: FIRMWAREVERSION?

Description: Read the XL™’s firmware version number.

Parameters: None.

Response: The response is an ASCII text string showing the version number, followed with a carriage return, line feed. See the example below:

Example:

```
-> FIRMWAREVERSION?↵                Command sent to the XL™.
Firmware Version = 1.00CRLF         Response sent to the host system.
```

Command: CHECKSUM?

Description: When this command is issued, the XL™ performs a diagnostic checksum test.

Parameters: None.

Response: After the test is done, the XL™ will respond either “PASS”, or “FAIL”. If the test fails, the “FAIL” response will be followed by a number indicating how far the checksum failed by. This number is useful for diagnosing problems.

Example: `->CHECKSUM?↵` Command sent to the XL™

Command: BATTERY?

Description: Read the internal battery voltage input channel.

Parameters: None.

Response: An ASCII string representing the battery voltage in volts is returned. The response is terminated with a carriage return, line feed.

Example:

`-> BATTERY?↵`	Command sent to XL™.
`12.5CRLF`	Response sent to the host system.

Command: BATTMIN?

Description: The XL™ keeps track of the lowest battery voltage reading. This command reads that value.

Parameters: None.

Response: An ASCII string representing the battery voltage in volts is returned. The response is terminated with a carriage return, line feed.

Example:

`-> BATTMIN?↵`	Command sent to XL™.
`10.8CRLF`	Response sent to the host system.

Command: BATTMAX?

Description: The XL™ keeps track of the highest battery voltage reading. This command reads that value.

Parameters: None.

Response: An ASCII string representing the battery voltage in volts is returned. The response is terminated with a carriage return, line feed.

Example:

`-> BATTMAX?↵`	Command sent to XL™.
`13.3CRLF`	Response sent to the host system.

Command: SYSTEMRESETS?

Description: The XL™ keeps track of the number of times it has been reset. These resets could have been caused by power failures or watchdog resets. This command reads that value.

Parameters: None.

Response: An ASCII string representing the number of system resets is returned. The response is terminated with a carriage return, line feed.

Example:

`-> SYSTEMRESETS?↵`	Command sent to XL™.
`3CRLF`	Response sent to the host system.

Command: RESETSTATUSVALUES

Description: This command causes the XL™ to reset the status values. This includes the reset counter, the max battery, and the min battery.

Parameters: None.

Response: None.

Example:

`->RESETSTATUSVALUES↵`	Command sent to XL™.
`CRLF`	Response sent to the host system.

Command: STATUSnSOURCE?

Description: The status values are a way to get quick access to certain values. This reads the source of an individual status value.

Parameters: There are 5 status values (1-5). The specific status value is determined by ‘n’.

Response: The source of the status value corresponding to ‘n’ (e.g. Stage, PSI, Temp).

Example: `-> STATUS1SOURCE?↵` Command sent to XL™.
`Status 1 Source = Stage␍␊` Response sent to the host system.

Command: STATUSnSOURCE=XXXXXXX

Description: Set the source of the n[th] status value.

Parameters: 'n' is from 1-5. The XL™ expects the new source of the status value corresponding to 'n' (e.g. Stage, PSI, Temp).

Response: None.

Example: `-> STATUS1SOURCE=PSI↵` Command sent to XL™.
`␍␊` Response sent to the host system.

Command: STATUSnLABEL?

Description: Read the label of the n[th] status value.

Parameters: There are 5 status values (1-5). The specific status value is determined by 'n'.

Response: The label of the n[th] status value (e.g. Stage =, WaterLvl, Batt =). The label is 8 characters long.

Example: `-> STATUS3LABEL?↵` Command sent to XL™.
`Status 3 Label = BattV = ␍␊` Response sent to the host system.

Command: STATUSnLABEL=XXXXXXXX

Description: Set the label of the n[th] status value.

Parameters: The label for the n[th] status value is set to the entered text. The text can be up to 8 characters. Entries shorter than 8 chars are padded with spaces to make 8.

Response: None.

Example: `-> STATUS5LABEL=WatLvl =↵` Command sent to XL™.
`␍␊` Response sent to the host system.

Command: STATUSnVALUE?

Description: Read the actual value of the n^{th} status value. Based on the source of the n^{th} status value, the XL™ scans the appropriate value.

Parameters: There are 5 status values (1-5). The specific status value is determined by 'n'.

Response: The XL™ scans the n^{th} status value and returns it.

Example:

```
-> STATUS2VALUE?↵                    Command sent to XL™.
Status 2 Value = 23.41␍␊             Response sent to the host system.
```

C.4 Data File Commands

Command: ERASEDATACARD

Description: Format the flash card in the card slot. This is an irreversible process.

Parameters: None.

Response: None.

Example:

```
->ERASEDATACARD↵                     Command sent to XL™.
␍␊                                   Response sent to the host system.
```

Command: ERASEINTERNALDATA

Description: Format the internal flash memory. This is an irreversible process.

Parameters: None.

Response: None.

Example:

```
->ERASEINTERNALDATA↵                 Command sent to XL™.
␍␊                                   Response sent to the host system.
```

Command: DIRDATACARD

Description: Get a list of the files on the data card.

Parameters: None.

Response: The response is a list of files in the following format:
ffffffff.eee sssssssss mm/dd/yy hh:mm
Where ffffffff.eee is the filename in an 8.3 format, sssssssss is the filesize, mm/dd/yy is the file date, and hh:mm is the file time.

Example:
```
->DIRDATACARD↵                          Command sent to XL™.
Site01.001    26,103  04/04/01  18:01
Site01.002    45,228  04/04/03  04:15
Site01.003     6,846  04/04/04  21:33
CRLF                                    Response sent to the host system.
```

Command: DIRINTERNALDATA

Description: Get a list of the files in the internal data memory.

Parameters: None.

Response: See DIRDATACARD above.

Example: See DIRDATACARD above.

Command: DUMPDATACARDFILE <*filename*>

Description: Transmits the file, specified by *filename*, through the serial port. If *filename* is not found, nothing is transmitted. Use DIRDATACARD to get the names of the files on the data card.

Parameters: The filename is specified by *filename*.

Response: The XL™ responds by transmitting the specified file.

Example:
```
->DUMPDATACARDFILE SITE1.001↵       Command sent to XL™.
CRLF                                Data file sent from the XL™.
MM/DD/YY HH:MM:SS Stage    Temp
03/01/01 12:00:00 24.83475 12.58436CRLF
03/01/01 12:01:00 24.72364 12.54357CRLF
03/01/01 12:02:00 24.63845 12.56457CRLF
                  |
                  |
03/01/01 13:45:00 24.38375 12.52645CRLF
03/01/01 13:46:00 24.83375 12.55673CRLF
03/01/01 13:47:00 24.83745 12.56535CRLF
CRLF
Done, Close FileCRLF
```

Command: DUMPINTERNALDATAFILE <*filename*>

Description: Transmits the file, specified by *filename*, through the serial port. If *filename* is not found, nothing is transmitted. Use DIRINTERNALDATA to get the names of the files in internal data memory.

Parameters: None.

Response: See DUMPDATACARDFILE above.

Example: See DUMPDATACARDFILE above.

Command: COPYDATATOCARD

Description: Copies all of the files from internal memory to the data card. Files are renamed as they are copied. The file extension gets changed so that the first letter is alphabetic. For example, 000 gets changed to A00, 001 gets changed to A01. If files already exist with 'A', then 'B' is used, and so on.

Parameters: None.

Response: The XL™ reports each file being copied. It gives the name before and after renaming.

Example:

`->COPYDATATOCARD↵`	Command sent to XL™.
`Copying Site_ID.000 ==> Site_ID.A00␍␊`	Response from XL™.
`Copying Site_ID.001 ==> Site_ID.A01␍␊`	Response from XL™.
`Copying Site_ID.002 ==> Site_ID.A02␍␊`	Response from XL™.
`␍␊`	Response from XL™.

Command: DATACARDBYTESUSED?

Description: Calculates the total number of bytes used on the data card. This number reflects the total space used on the card.

Parameters: None.

Response: The response will be an ASCII string representing the number of bytes used.

Example:

`->DATACARDBYTESUSED?↵`	Command sent to the XL™.
`Data Card Bytes Used = 20311␍␊`	Response from the XL™.

Command: DATACARDBYTESFREE?

Description: Calculates the total number of bytes free on the data card. This number reflects the total space free on the card.

Parameters: None.

Response: The response will be an ASCII string representing the number of bytes free.

Example:

`->DATACARDBYTESFREE?↵`	Command sent to the XL™.
`Data Card Bytes Free = 9710234␍␊`	Response from the XL™.

Command: INTERNALDATABYTESUSED?

Description: Calculates the total number of bytes used in internal memory. This number reflects the total space used in the internal memory.

Parameters: None.

Response: The response will be an ASCII string representing the number of bytes used.

Example:

`->INTERNALDATABYTESUSED?↵`	Command sent to XL™.
`Internal Data Bytes Used = 710234␍␊`	Response from XL™.

Command: INTERNALDATABYTESFREE?

Description: Calculates the total number of bytes free in internal memory. This number reflects the total space free in the internal memory.

Parameters: None.

Response: The response will be an ASCII string representing the number of bytes free.

Example:

`->INTERNALDATABYTESFREE?↵`	Command sent to XL™.
`Internal Data Bytes Free = 9710234␍␊`	Response from XL™.

C.5 Scanning Commands

Command: SCANNING?

Description: Reads the scanning flag. This value determines whether or not the XL™ is set to scan.

Parameters: None.

Response: The response will be either the "ON" or "OFF".

Examples:

`-> SCANNING?↵`	Command sent to the XL™.
`OFF⌐R└F`	Response sent to the host system.

Command: SCANNING=XXX

Description: Set the scanning flag.

Parameters: This flag can be set to either "ON" or "OFF".

Response: None.

Example:

`-> SCANNING=ON↵`	Command sent to the XL™
`⌐R└F`	Response from the XL™.

Command SCANRATE?

Description: Read the current scan rate interval.

Parameters: None.

Response: An ASCII text string is returned with the time formatted as hh:mm:ss. The response is terminated using a carriage return, line feed.

Example:

`-> SCANRATE?↵`	Command sent to the XL™.
`00:00:30⌐R└F`	Response sent to the host system.

Command: SCANRATE=hh:mm:ss

Description: Set the time interval for the scanning operation. Keep in mind that the scan rate should be as fast as, or some multiple faster than the report intervals. See the note at the end of this section.

Parameters: The parameter string is formatted for two digits used for the hours, (hh); two digits used for the minutes, (mm); and two digits used for the seconds, (ss). The hour, minute and second values must be separated with some character as shown above. Note the space between the command and the parameter.

Response: None, use the SCANRATE command to see the new value.

Example:	`-> SCANRATE=00:01:00↵`	Command sent to the XL™. This sets the scan rate to force a scan once every minute.
	`ᶜᴿᴸᶠ`	Response sent to the host system.

Command: NEXTSCAN?

Description: Get the amount of time until the next scan.

Parameters: None.

Response: The response will be an ASCII string in the format of HH:MM:SS indicating the amount of time until the next scan.

Examples:	`-> NEXTSCAN?↵`	Command sent to the XL™.
	`00:14:59ᶜᴿᴸᶠ`	Response sent to the host system.

Command: NEXTSCANAT?

Description: Get the scheduled time of the next scan.

Parameters: None.

Response: The response will be an ASCII string in the format of HH:MM:SS.

Examples:	`-> NEXTSCANAT?↵`	Command sent to the XL™.
	`16:30:00ᶜᴿᴸᶠ`	Response sent to the host system.

Command: NEXTSCANAT=hh:mm:ss

Description: Set the time at which the XL™ will performs its next scan.

Parameters: The format for the time is HH:MM:SS. The time ranges from 00:00:00 to 23:59:59.

Response: None

Examples:

`-> NEXTSCANAT=12:00:00↵`	Command sent to the XL™.
`ᶜᴿᴸᶠ`	Response sent to the host system.

C.6 Sensor Input Commands

Command: STAGE?

Description: Get a reading from the internal pressure sensor.

Parameters: None.

Response: An ASCII string representing the stage is returned. The response is terminated with a carriage return, line feed.

Example:

`-> STAGE?↵`	Command sent to XL™.
`4.25ᶜᴿᴸᶠ`	Response sent to the host system.

Command: STAGE=X.XXX

Description: Calibrates the user offset for stage. The XL™ performs a pressure measurement and compares the measured value to the entered value to get the offset. To perform this calibration, measure the stage with a reference and enter the reference value into this function.

Parameters: This command accepts a real number representing stage level.

Response: None.

Example:

`-> STAGE=8.6↵`	Command sent to XL™.
`ᶜᴿᴸᶠ`	Response sent to the host system.

Command: STAGEUNITS?

Description: Check which units the XL™ is reporting stage in.

Parameters: None.

Response: An ASCII string representing the stage units. The response can be any of the following: “Feet”, “Meters”, “Inches”, or “UserDef”.

Example: `-> STAGEUNITS?↵` Command sent to XL™.
`Stage Units = Feet⌐R└F` Response sent to the host system.

Command: STAGEUNITS=XXXXXXX

Description: Set which units the XL™ is reporting stage in.

Parameters: This command takes an ASCII string representing the units in which stage should be reported. This value tells the XL™ what slope value to use when calculating stage from pressure. For example, the general equation for stage is: Stage = Slope * Pressure + Offset. For stage in feet, the slope is 2.3067. In meters, it's 0.70308. A custom slope value of 1.234 can be entered as "UserDef 1.234".

Response: None.

Example: `-> STAGEUNITS=Feet↵` Command sent to XL™.
`⌐R└F` Response sent to the host system.

Command: STAGEOFFSET?

Description: Directly read the offset for the stage value.

Parameters: None.

Response: An ASCII string representing the offset is returned. The response is terminated with a carriage return, line feed.

Example: `-> STAGEOFFSET?↵` Command sent to XL™.
`Stage Offset = 0.21⌐R└F` Response sent to the host system.

Command: STAGEOFFSET=X.XXX

Description: Set the user offset for stage.

Parameters: This command accepts a real number representing stage offset.

Response: None.

Example: `-> STAGEOFFSET=0↵` Command sent to XL™.
`⌐R└F` Response sent to the host system.

Command: TEMPERATURE?

Description: Get a reading from the internal temperature sensor. This input is expecting a type T thermocouple.

Parameters: None.

Response: An ASCII string representing the temperature is returned. The response is terminated with a carriage return, line feed.

Example:

```
-> TEMPERATURE?↵                    Command sent to XL™.
Temperature = 23.65␍␊F              Response sent to the host system.
```

Command: TEMPERATUREUNITS?

Description: Get the units in which temperature is being reported. This will be either "Deg C" for Celsius, or "Deg F" for Fahrenheit.

Parameters: None.

Response: An ASCII string representing the temperature units is returned. The response is terminated with a carriage return, line feed.

Example:

```
-> TEMPERATUREUNITS?↵               Command sent to XL™.
Temperature Units = Deg F␍␊F        Response sent to the host system.
```

Command: TEMPERATUREUNITS=XXXXX

Description: Set the units in which temperature is being reported. This can be either "Deg C" for Celsius, or "Deg F" for Fahrenheit.

Parameters: This command accepts an ASCII string of either "Deg C" or "Deg F".

Response: None.

Example:

```
-> TEMPERATUREUNITS=Deg F↵          Command sent to XL™.
␍␊F                                 Response sent to the host system.
```

Command COUNTERTOTAL?

Description: Read the total event counter variable. Reading the counter will not clear it.

Parameters: None.

Response: The response is an ASCII text string representing a value between 0 and 65535. The response will be followed by a carriage return, line feed.

Example:

```
-> COUNTERTOTAL?↵          Command sent to the XL™
Counter Total = 124⌐R└F    Response sent to the host system.
```

Command: COUNTERTOTAL=XXXXX

Description: Set the counter total to a specific value.

Parameters: The range for the counter total is 0-65535.

Response: None.

Example:

```
-> COUNTERTOTAL=100↵       Command sent to the XL™.
⌐R└F                       Response sent to the host system.
```

Command COUNTERSUM?

Description: Read the counter sum. Reading the counter will not clear it.

Parameters: None.

Response: The response is an ASCII text string representing a value between 0 and 65535.

Example:

```
-> COUNTERSUM?↵            Command sent to the XL™.
Counter Sum = 124⌐R└F      Response sent to the host system.
```

Command: COUNTERSUM=XXXXX

Description: Set the counter sum to a specific value.

Parameters: The range for the counter sum is 0-65535.

Response: None.

Example:

```
-> COUNTERSUM=4534↵        Command sent to the XL™.
⌐R└F                       Response sent to the host system.
```

Command COUNTERSLOPE?

Description: Read the counter slope. This value scales raw counts to the value given to the user.

Parameters: None.

Response: The response is an ASCII text string representing the slope value. The response will be followed by a carriage return, line feed.

Example:

`-> COUNTERSLOPE?↵`	Command sent to the XL™
`Counter Slope = 0.01␍␊`	Response sent to the host system.

Command: COUNTERSLOPE=XXXXX

Description: Set the counter slope. This value scales raw counts to the value given to the user.

Parameters: This command accepts an ASCII string representing the new slope value.

Response: None.

Example:

`-> COUNTERSLOPE=1.0↵`	Command sent to the XL™.
`␍␊`	Response sent to the host system.

Command SUMINTERVAL?

Description: Read the sum interval. This interval is the amount of time after which the XL™ resets COUNTERSUM.

Parameters: None.

Response: The response is an ASCII text string representing the sum interval. The format of the sum interval is HH:MM:SS.

Example:

`-> SUMINTERVAL?↵`	Command sent to the XL™
`Sum Interval = 04:00:00␍␊`	Response sent to the host system.

Command: SUMINTERVAL=hh:mm:ss

Description: Set the sum interval.

Parameters: This format for the new sum interval is HH:MM:SS.

Response: None.

Example:
```
-> SUMINTERVAL=01:00:00↵
└R└F
```
Command sent to the XL™.
Response sent to the host system.

Command COUNTERMODE?

Description: Read the counter mode.

Parameters: None.

Response: The response is an ASCII text string representing the counter mode. The response will be followed by a carriage return, line feed.

Example:
```
-> COUNTERMODE?↵
Counter Mode = Off    └R└F
```
Command sent to the XL™
Response sent to the host system.

Command: COUNTERMODE=XXXXX

Description: Set the counter slope. This value scales raw counts to the value given to the user.

Parameters: This command accepts an ASCII string representing the new slope value.

Response: None.

Example:
```
-> COUNTERMODE=Count Only↵
└R└F
```
Command sent to the XL™.
Response sent to the host system.

Command: RESETCOUNTERS

Description: Reset both total and summation counters to zero.

Parameter: None.

Response: None.

Example:
```
-> RESETCOUNTERS↵
└R└F
```
Command sent to the XL™
Response sent to the host system.

Command: ANALOGn?

Description: Read the analog input channel n.

Parameters: The parameter 'n' is 1-4 for the four analog channels.

Response: An ASCII string representing a voltage between -5.00 and +5.00 volts. The response is terminated with a carriage return and a line feed.

Example: `-> ANALOG1?↵` Command sent to XL™.
`Analog 1 = 3.23265ᶜᴿᴸᶠ` Response sent to the host system.

Command ANALOGnSLOPE?

Description: Read the slope for the n^{th} analog sensor. This value is used in the equation to calculate the sensor value from the raw voltage reading.

Parameters: 'n' is 1-4.

Response: A real number is returned, represented in ASCII text format.

Example: `-> ANALOG2SLOPE?↵` Command sent to the XL™.
`Analog 2 Slope = -12.441ᶜᴿᴸᶠ` Response sent to the host system.

Command ANALOGnSLOPE=XXX.XX

Description: Set the slope for the n^{th} analog sensor. This value is used in the equation to calculate the sensor value from the raw voltage reading.

Parameters: 'n' is 1-4. 'XXX.XX' is a real number. It is to be in the format of: <+/->XXX<.>XX<E[+/-]XX>. In other words, optional sign, digits, optional decimal point, digits, optional exponent, optional sign, exponent digits.

Response: None.

Example: `-> ANALOG2SLOPE=1.44E-6↵` Command sent to the XL™.
`ᶜᴿᴸᶠ` Response sent to the host system.

Command **ANALOGnOFFSET?**

Description: Read the offset for the n[th] analog sensor. This value is used in the equation to calculate the sensor value from the raw voltage reading.

Parameters: 'n' is either 1 or 2.

Response: A real number is returned, represented in ASCII text format.

Example:

```
-> ANALOG1OFFSET?↵                    Command sent to the XL™.
Analog 1 Offset = 0.0CRLF             Response sent to the host system.
```

Command **ANALOGnOFFSET**=XXX.XX

Description: Set the offset for the n[th] analog sensor. This value is used in the equation to calculate the sensor value from the raw voltage reading.

Parameters: 'n' is either 1 or 2. 'XXX.XX' is a real number. It is to be in the format of: <+/->XXX<.>XX<E[+/-]XX>. In other words, optional sign, digits, optional decimal point, digits, optional exponent, optional sign, exponent digits.

Response: None.

Example:

```
-> ANALOG2OFFSET=1.5↵                 Command sent to the XL™.
CRLF                                  Response sent to the host system.
```

C.7 Digital I/O Commands

Command **DIGITALnMODE?**

Description: Read the operation mode of the n[th] digital channel. The mode determines whether the channel is an input or output, and which type.

Parameters: There are two digital channels, so n can be either 1 or 2.

Response: An ASCII string is returned containing the mode. The mode can be "Digital Input", "Trigger Pulse", "Trigger Cont.", or "On For Scan". See the Digital I/O chapter of the user's manual for more information.

Example:

```
-> DIGITAL1MODE?↵                     Command sent to the XL™.
Digital 1 Mode = Digital InputCRLF    Response from the XL™.
```

Command DIGITALnMODE=

Description: Write the operation mode of the nth digital channel. The mode determines whether the channel is an input or output, and which type.

Parameters: There are two digital channels, so n can be either 1 or 2. For the mode, this command expects an ASCII string containing the mode. The mode can be "Digital Input", "Trigger Pulse", "Trigger Cont.", or "On For Scan". See the Digital I/O chapter of the user's manual for more information.

Response: None.

Example:

```
-> DIGITAL1MODE=Trigger Pulse↵
⌐R⌊F
```

Command sent to the XL™.
Response from the XL™.

Command DIGITALnTRIGGERSOURCE?

Description: Read the trigger source for the nth digital channel. This is the source of the value which is compared against the trigger point.

Parameters: There are two digital channels, so n can be either 1 or 2.

Response: An ASCII string is returned containing the source. The source can be "Stage", "PSI", "Temp", etc.

Example:

```
-> DIGITAL1TRIGGERSOURCE?↵
Digital 1 Trigger Source = Stage⌐R⌊F
```

Command sent to XL™.
Response from XL™.

Command DIGITALnTRIGGERSOURCE=

Description: Write the trigger source of the nth digital channel. The mode determines whether the channel is an input or output, and which type.

Parameters: There are two digital channels, so n can be either 1 or 2. For the trigger source, this command expects an ASCII string. The trigger source can be "Stage", "PSI", "Temp", etc.

Response: None.

Example: `-> DIGITAL1TRIGGERSOURCE=Stage↵` Command sent to XL™.
`⌐R└F` Response from XL™.

Command DIGITALnTRIGGERTYPE?

Description: Read the trigger type of the nth digital channel. The trigger type determines the way in which the value of the trigger source is compared to the trigger point.

Parameters: There are two digital channels, so n can be either 1 or 2.

Response: An ASCII string is returned containing the trigger type. The trigger type can be "Always", "> Set Pt", or "< Set Pt".

Example: `-> DIGITAL1TRIGGERTYPE?↵` Command sent to the XL™.
`Digital 1 Trigger Type = Always⌐R└F` Response from the XL™.

Command DIGITALnTRIGGERTYPE=

Description: Write the trigger type of the nth digital channel. The trigger type determines the way in which the value of the trigger source is compared to the trigger point.

Parameters: There are two digital channels, so n can be either 1 or 2. For the trigger type, this command expects an ASCII string. The trigger type can be "Always", "> Set Pt", or "< Set Pt".

Response: None.

Example: `-> DIGITAL1TRIGGERTYPE=> Set Pt↵` Command sent to the XL™.
`⌐R└F` Response from the XL™.

Command DIGITALnTRIGGERPOINT?

Description: Read the trigger point of the nth digital channel. The trigger point is compared to the value of the trigger source, and if the comparison meets the condition specified by the trigger type, the channel is activated.

Parameters: There are two digital channels, so n can be either 1 or 2.

Response: An ASCII string containing a real number is returned for the trigger point.

Example: `-> DIGITAL1TRIGGERPOINT?↵` Command sent to the XL™.
`Digital 1 Trigger Point = 5.75CRLF` Response from the XL™.

Command DIGITALnTRIGGERPOINT=

Description: Write the trigger point of the nth digital channel. The trigger point is compared to the value of the trigger source, and if the comparison meets the condition specified by the trigger type, the channel is activated.

Parameters: There are two digital channels, so n can be either 1 or 2. The command accepts a real number for the trigger point.

Response: None.

Example: `-> DIGITAL1TRIGGERPOINT=4.25↵` Command sent to the XL™.
`CRLF` Response from the XL™.

Command DIGITALnPULSELENGTH?

Description: Read the pulse length of the nth digital channel. The pulse length is the amount of time in milliseconds that the digital channel outputs a signal.

Parameters: There are two digital channels, so n can be either 1 or 2.

Response: An ASCII string containing an integer number is returned for the pulse length.

Example: `-> DIGITAL1PULSELENGTH?↵` Command sent to the XL™.
`Digital 1 Pulse Length = 200CRLF` Response from the XL™.

Command DIGITALnPULSELENGTH=

Description: Write the pulse length of the nth digital channel. The pulse length is the amount of time in milliseconds that the digital channel outputs a signal.

Parameters: There are two digital channels, so n can be either 1 or 2. The command accepts an integer number for the pulse length.

Response: None.

Example: `-> DIGITAL1PULSELENGTH=150↵` Command sent to the XL™.
`CRLF` Response from the XL™.

Command DIGITALnWARMUPDELAY?

Description: Read the warm up delay of the nth digital channel. The warm up delay is the amount of time in milliseconds that the digital channel waits before beginning an input or output.

Parameters: There are two digital channels, so n can be either 1 or 2.

Response: An ASCII string containing an integer number is returned for the warm up delay.

Example:

```
-> DIGITAL1WARMUPDELAY?↵                Command sent to the XL™.
Digital 1 Warm Up Delay = 200␍␊         Response from the XL™.
```

Command DIGITALnWARMUPDELAY=

Description: Write the warm up delay of the nth digital channel. The warm up delay is the amount of time in milliseconds that the digital channel waits before beginning an input or output.

Parameters: There are two digital channels, so n can be either 1 or 2. The command accepts an integer number for the warm up delay.

Response: None.

Example:

```
-> DIGITAL1WARMUPDELAY=150↵             Command sent to the XL™.
␍␊                                      Response from the XL™.
```

C.8 SDI-12 Commands

Command: SDITESTADDRESS?

Description: Read the address that the XL™ uses when issuing SDI-12 commands.

Parameters: None.

Response: An ASCII text string is returned with the address.

Example:

```
-> SDITESTADDRESS?↵                     Command sent to the XL™.
SDI Test Address = 1␍␊                  Response from the XL™.
```

Command: SDITESTADDRESS=x

Description: Set the address that the XL™ uses when issuing SDI-12 commands.

Parameters: The address can be 0-9 or A-Z.

Response: None.

Example:

```
-> SDITESTADDRESS=1↵
↳R↳F
```

Command sent to the XL™.
Response from the XL™.

Command: SDIACKNOWLEDGE

Description: Issue an acknowledge command on the SDI-12 bus using the address specified by SDITESTADDRESS.

Parameters: None.

Response: The XL™ takes the response it receives on the SDI-12 bus and returns it. In the case of the acknowledge command, an SDI-12 sensor will respond with its address. If no response is received, the XL™ responds with "Communication Timed Out".

Example:

```
-> SDIACKNOWLEDGE↵
1↳R↳F
```

Command sent to the XL™.
Response from the XL™.

Command: SDIIDENTIFY

Description: Issue an identify command on the SDI-12 bus using the address specified by SDITESTADDRESS.

Parameters: None.

Response: The XL™ takes the response it receives on the SDI-12 bus and returns it. If no response is received, the XL™ responds with "Communication Timed Out". The response to the identify command is as follows:

allccccccccmmmmmmvvvxx...xx

a: sensor address ("0-9", "A-Z", "a-z", "*", "?").

ll: SDI-12 version compatibility level, e.g. version 1.2 is represented as "12".

cccccccc: 8 character vendor identification to be specified by the vendor and usually in the form of a company name or its abbreviation.

mmmmmm: 6 character field specifying the sensor model number.

vvv: 3 character field specifying the sensor version number.

xx...xx: variable length field (up to a maximum of 13) to be used for serial number or other specific sensor information not relevant to the operation of the data recorder.

Example:
```
-> SDIIDENTIFY↵                                  Command sent to XL™.
112     DAA H-415vvvS#123456V101<CR><LF>         Response from XL™.
```

Command: SDIVERIFY

Description: Issue a verify command on the SDI-12 bus using the address specified by SDITESTADDRESS.

Parameters: None.

Response: The XL™ takes the response it receives on the SDI-12 bus and returns it. If no response is received, the XL™ responds with "Communication Timed Out". The response to the verify command depends on the manufacturer of the sensor. Consult the documentation for the sensor for further information.

Example:
```
-> SDIVERIFY↵                                    Command sent to XL™.
11234<CR><LF>                                    Response from XL™.
```

Command: SDIMEASURE

Description: Issue a measure command on the SDI-12 bus using the address specified by SDITESTADDRESS.

Parameters: None.

Response: The XL™ takes the response it receives on the SDI-12 bus and returns it. If no response is received, the XL™ responds with "Communication Timed Out". The response to the measure command is as follows:

atttn

a: sensor address (“0-9", “A-Z”, “a-z”, “*”, “?”).

ttt: 3 digit integer (000-999) specifying the maximum time, in seconds, the sensor will take to complete the command and have measurement data available in its buffer.

n: single digit integer (0-9) specifying the number of values that will be placed in the data buffer. If “n” is zero (0), no data will be available using subsequent “D” commands.

Example: `-> SDIMEASURE↵` Command sent to the XL™.
`10022⌐R└F` Response from the XL™.

Command: SDIDATAn

Description: Issue a data command on the SDI-12 bus using the address specified by SDITESTADDRESS.

Parameters: ‘n’ is an integer specifying which data command to issue (DATA0, DATA1, etc).

Response: The XL™ takes the response it receives on the SDI-12 bus and returns it. If no response is received, the XL™ responds with “Communication Timed Out”. The response to the data command is as follows:

+xx.xxx+y.yyy

x.xxx: channel A voltage

y.yyy: channel B voltage

Example: `-> SDIDATA0↵` Command sent to the XL™.
`+1.234+9.876⌐R└F` Response from the XL™.

Command: SDIEXTENDED=<*command*>

Description: Issue an extended command on the SDI-12 bus using the address specified by SDITESTADDRESS.

Parameters: “command” is the command that is to be issued on the SDI-12 bus. It needs to be a valid SDI-12 command.

Response: The XL™ takes the response it receives on the SDI-12 bus and returns it. If no response is received, the XL™ responds with "Communication Timed Out". The response to the extended command depends entirely on the command.

Example:

```
-> SDIEXTENDED=*A↵                    Command sent to the XL™.
1<CR><LF>                             Response from the XL™.
```

C.9 Output Commands

Command: AUTOPRINT?

Description: Read the auto print flag. If this flag is set, all data being logged also gets sent out the serial port.

Parameters: None.

Response: An ASCII text string is returned with either "On " or "Off".

Example:

```
-> AUTOPRINT?↵                        Command sent to the XL™.
Auto Print = Off<CR><LF>              Response from the XL™.
```

Command: AUTOPRINT=XXX

Description: Set the auto print flag.

Parameters: The flag can be set to either "On " or "Off".

Response: None.

Example:

```
-> AUTOPRINT=ON↵                      Command sent to the XL™.
<CR><LF>                              Response from the XL™.
```

Command: REMOTEDISPLAYLABEL?

Description: Read the remote display label. This label will be printed with the data value on the display.

Parameters: None.

Response: An 8 character ASCII text string with the label.

Example: `-> REMOTEDISPLAYLABEL?↵` Command sent to XL™.
`Remote Display Label = "WtrLvl= "`CR LF Response from XL™.

Command: REMOTEDISPLAYLABEL=XXXXXX

Description: Set the remote display label. This label will be printed with the data value on the display.

Parameters: The XL™ expects an ASCII string for the new label. This label is 8 characters in length.

Response: None.

Example: `-> REMOTEDISPLAYLABLE=Stage = ↵` Command sent to the XL™.
CR LF Response from the XL™.

Command: REMOTEDISPLAYSOURCE?

Description: Read the source of the data to be sent to the remote display.

Parameters: None.

Response: The source assigned to the remote display (e.g. Stage, PSI, Temp).

Example: `-> REMOTEDISPLAYSOURCE?↵` Command sent to XL™.
`Remote Display Source = Stage`CR LF Response from XL™.

Command: REMOTEDISPLAYSOURCE=XXXXXX

Description: Set the source assigned to the remote display.

Parameters: The XL™ expects an ASCII string for the new source (e.g. Stage, PSI, Temp).

Response: None.

Example: `-> REMOTEDISPLAYSOURCE=PSI↵` Command sent to the XL™.
CR LF Response from the XL™.

Command: EMAIL?

Description: Read the e-mail modem enable flag. Setting this flag enables support for the H-260 e-mail modem.

Parameters: None.

Response: An ASCII text string is returned with either “On ” or “Off“.

Example:

```
-> EMAIL?↵                          Command sent to the XL™.
Email = Off␍␊                       Response from the XL™.
```

Command: EMAIL=XXX

Description: Set the e-mail modem enable flag. Setting this flag enables support for the H-260 e-mail modem..

Parameters: The flag can be set to either “On ” or “Off“.

Response: None.

Example:

```
-> EMAIL=ON↵                        Command sent to the XL™.
␍␊                                  Response from the XL™.
```

Command: EMAILSENDRATE?

Description: Read the e-mail send rate. This rate determines how often e-mail messages are to be sent.

Parameters: None.

Response: An ASCII text string is returned with the send rate. The send rate can be “Daily”, “Hourly”, or “Each Scn”.

Example:

```
-> EMAILSENDRATE?↵                  Command sent to the XL™.
Email Send Rate = Daily␍␊           Response sent to the host system.
```

Command: EMAILSENDRATE=xxxxxxxx

Description: Set the transmission rate for the e-mail modem.

Parameters: The new rate can be “Daily”, “Hourly”, or “Each Scn”.

Response: None.

Example:
```
-> EMAILSENDRATE=Each Scn↵
└R└F
```
Command sent to the XL™.
Response sent to the host system.

Command: EMAILSENDHEADERS?

Description: Read the e-mail modem send headers flag. If this flag is set, headers will be included with each e-mail transmission.

Parameters: None.

Response: An ASCII text string is returned with either "On " or "Off".

Example:
```
-> EMAILSENDHEADERS?↵
Email Send Headers = Off└R└F
```
Command sent to the XL™.
Response from the XL™.

Command: EMAILSENDHEADERS=XXX

Description: Set the e-mail modem send headers flag. If this flag is set, headers will be included with each e-mail transmission.

Parameters: The flag can be set to either "On " or "Off".

Response: None.

Example:
```
-> EMAILSENDHEADERS=ON↵
└R└F
```
Command sent to the XL™.
Response from the XL™.

Command: EMAILSENDSITEID?

Description: Read the e-mail modem send site ID flag. If this flag is set, the site ID will be included with each e-mail transmission.

Parameters: None.

Response: An ASCII text string is returned with either "On " or "Off".

Example:
```
-> EMAILSENDSITEID?↵
Email Send Site ID = Off└R└F
```
Command sent to the XL™.
Response from the XL™.

Command: EMAILSENDSITEID=XXX

Description: Set the e-mail modem send site ID flag. If this flag is set, the site ID will be included with each e-mail transmission.

Parameters: The flag can be set to either "On " or "Off".

Response: None.

Example:

```
-> EMAILSENDHSITEID=ON↵                    Command sent to the XL™.
└R└F                                       Response from the XL™.
```

Command: EMAILALARMS?

Description: Read the e-mail modem alarms flag. If this flag is set, the alarm conditions will be evaluated during each scan to determine whether or not to send an e-mail message.

Parameters: None.

Response: An ASCII text string is returned with either "On " or "Off".

Example:

```
-> EMAILALARMS?↵                           Command sent to the XL™.
Email Alarms = Off└R└F                     Response from the XL™.
```

Command: EMAILALARMS=XXX

Description: Set the e-mail modem alarms flag. If this flag is set, the alarm conditions will be evaluated during each scan to determine whether or not to send an e-mail message.

Parameters: The flag can be set to either "On " or "Off".

Response: None.

Example:

```
-> EMAILALARMS=ON↵                         Command sent to the XL™.
└R└F                                       Response from the XL™.
```

Command: EMAILTRIGGERSOURCE?

Description: Read the source that will be evaluated when determining whether or not to trigger an e-mail transmission.

Parameters: None

Response: The response will be an ASCII text string with the current e-mail trigger source (e.g. Stage, PSI, Temp).

Example:
```
-> EMAILTRIGGERSOURCE?↵                    Command sent to XL™.
 Email Trigger Source = Stage ᶜᴿᴸᶠ          Response from XL™.
```

Command: EMAILTRIGGERSOURCE=XXXXXX

Description: Set the e-mail trigger source.

Parameters: The new trigger source value for the e-mail (e.g. Stage, PSI, Temp).

Response: None.

Example:
```
-> EMAILTRIGGERSOURCE=PSI↵     Command sent to the XL™.
ᶜᴿᴸᶠ                           Response sent to the host system.
```

Command: EMAILTRIGGERMODE?

Description: Read the current mode of operation of the H-260 e-mail modem. This value, along with trigger source, determines what conditions are checked when evaluating whether or not to send an e-mail message.

Parameters: None.

Response: An ASCII text string is returned. The response will be "Always ", "> Set Pt", "< Set Pt", or "Delta ".

Example:
```
-> EMAILTRIGGERMODE?↵                  Command sent to XL™.
Email Trigger Mode = Always  ᶜᴿᴸᶠ       Response from XL™.
```

Command: EMAILTRIGGERMODE=XXXXXX

Description: Set the current mode of email operation.

Parameters: The flag can be set to "Always ", "> Set Pt", "< Set Pt", or "Delta ".

Response: None.

Example: `-> EMAILTRIGGERMODE=> Set Pt↵` Command sent to XL™.
`ᶜᴿᴸᶠ` Response from XL™.

Command: EMAILTRIGGERPOINT?

Description: Read the trigger point for the e-mail condition check. The value indicated by the e-mail trigger source is compared to this value. Depending on the e-mail trigger mode, different comparisons will be made. If the trigger conditions are met, an e-mail message is generated.

Parameters: None.

Response: An ASCII text string is returned with a real number indicating the trigger point.

Example: `-> EMAILTRIGGERPOINT?↵` Command sent to XL™.
`Email Trigger Point = 2.5ᶜᴿᴸᶠ` Response from XL™.

Command: EMAILTRIGGERPOINT=XXXXX

Description: Set the e-mail trigger point.

Parameters: The XL™ accepts a real number represented in ASCII text format.

Response: None.

Example: `-> EMAILTRIGGERPOINT=3.3↵` Command sent to XL™.
`ᶜᴿᴸᶠ` Response from XL™.

Command: ALERT?

Description: Read the flag enabling the ALERT radio.

Parameters: None.

Response: The ALERT flag can be either “On” or “Off”.

Example: `-> ALERT?↵` Command sent to XL™.
`ALERT = Offᶜᴿᴸᶠ` Response from XL™.

Command: **ALERT=XXX**

Description: Set the flag enabling or disabling the ALERT radio.

Parameters: The XL™ accepts a value of either "On" or "Off".

Response: None.

Example:

`-> ALERT=On↵`	Command sent to XL™.
`CRLF`	Response from XL™.

Command: **ALERTRADIOTYPE?**

Description: Read the ALERT radio type.

Parameters: None.

Response: The ALERT radio type can be either "HS-3210" or "H-3400".

Example:

`-> ALERTRADIOTYPE?↵`	Command sent to XL™.
`ALERT Radio Type = HS-3210CRLF`	Response from XL™.

Command: **ALERTRADIOTYPE=XXXXXXX**

Description: Set the ALERT radio type.

Parameters: The ALERT radio type can be either "HS-3210" or "H-3400".

Response: None.

Example:

`-> ALERTRADIOTYPE=H-3400↵`	Command sent to XL™.
`CRLF`	Response from XL™.

Command: **ALERTnSOURCE?**

Description: Read the source for the n^{th} ALERT alarm.

Parameters: 'n' ranges from 1-5.

Response: The source can be any input source, such as "Stage", "Batt", "Temp", etc.

Example: `-> ALERT1SOURCE?↵` Command sent to XL™.
`ALERT 1 Source = Stage␍␊` Response from XL™.

Command: ALERTnSOURCE=XXXXX

Description: Set the source for the nth ALERT alarm.

Parameters: The source can be any input source, such as “Stage”, “Batt”, “Temp”, etc.

Response: None.

Example: `-> ALERT2SOURCE=PSI↵` Command sent to XL™.
`␍␊` Response from XL™.

Command: ALERTnID?

Description: Read the ID for the nth ALERT alarm.

Parameters: ‘n’ ranges from 1-5.

Response: The ID is a four digit integer.

Example: `-> ALERT1ID?↵` Command sent to XL™.
`ALERT 1 ID = 0123␍␊` Response from XL™.

Command: ALERTnID=XXXX

Description: Set the ID for the nth ALERT alarm.

Parameters: ‘n’ ranges from 1-5. The ID is a four digit integer.

Response: None.

Example: `-> ALERT2ID=4321↵` Command sent to XL™.
`␍␊` Response from XL™.

Command: ALERTnMAXRANGE?

Description: Read the maximum range for the nth ALERT alarm.

Parameters: ‘n’ ranges from 1-5.

Response: Max range is a floating point number.

Example:
```
-> ALERT4MAXRANGE?↵                    Command sent to XL™.
ALERT 4 Max Range = 12.5CRLF           Response from XL™.
```

Command: ALERTnMAXRANGE=XXXXXX

Description: Set the maximum range for the nth ALERT alarm.

Parameters: 'n' ranges from 1-5. Max range is a floating point number.

Response: None.

Example:
```
-> ALERT3MAXRANGE=10.75↵               Command sent to XL™.
CRLF                                   Response from XL™.
```

Command: ALERTnMINRANGE?

Description: Read the minimum range for the nth ALERT alarm.

Parameters: 'n' ranges from 1-5.

Response: Min range is a floating point number.

Example:
```
-> ALERT4MINRANGE?↵                    Command sent to XL™.
ALERT 4 Min Range = 0.0CRLF            Response from XL™.
```

Command: ALERTnMINRANGE=XXXXXX

Description: Set the minimum range for the nth ALERT alarm.

Parameters: 'n' ranges from 1-5. Min range is a floating point number.

Response: None.

Example:
```
-> ALERT3MINRANGE=1.5↵                 Command sent to XL™.
CRLF                                   Response from XL™.
```

Command: ALERTnTRIGGERMODE?

Description: Read the current mode of operation of the nth ALERT alarm. This value, along with the source, determines what conditions are checked when evaluating whether or not to send an ALERT transmission.

Parameters: None.

Response: An ASCII text string is returned. The response will be "Always ", "> Set Pt", "< Set Pt", or "Delta ".

Example:

```
-> ALERT1TRIGGERMODE?↵                    Command sent to XL™.
ALERT 1 Trigger Mode = Always  ␍␊         Response from XL™.
```

Command: ALERTnTRIGGERMODE=XXXXXX

Description: Set the current mode of operation for the nth ALERT alarm.

Parameters: 'n' can range from 1-5. The flag can be set to "Always ", "> Set Pt", "< Set Pt", or "Delta ".

Response: None.

Example:

```
-> ALERT2TRIGGERMODE=> Set Pt↵            Command sent to XL™.
␍␊                                        Response from XL™.
```

Command: ALERTnTRIGGERPOINT?

Description: Read the trigger point for the ALERT condition check. The value indicated by the ALERT source is compared to this value. Depending on the ALERT trigger mode, different comparisons will be made. If the trigger conditions are met, an ALERT radio transmission is generated.

Parameters: None.

Response: An ASCII text string is returned with a real number indicating the trigger point.

Example:

```
-> ALERT5TRIGGERPOINT?↵                   Command sent to XL™.
ALERT 5 Trigger Point = 2.5␍␊             Response from XL™.
```

Command: ALERTnTRIGGERPOINT=XXXXX

Description: Set the trigger point.

Parameters: The XL™ accepts a real number represented in ASCII text format.

Response: None.

Example:
```
-> ALERT4TRIGGERPOINT=3.3↵
└R└F
```
Command sent to XL™.
Response from XL™.

Command: ALERTnHOLDOFF?

Description: Read the hold off value for the nth ALERT alarm. Hold off is a delay, in milliseconds, which comes after a transmission.

Parameters: 'n' ranges from 1-5.

Response: Hold off is an integer, representing milliseconds.

Example:
```
-> ALERT4HOLDOFF?↵
ALERT 4 Hold Off = 100└R└F
```
Command sent to XL™.
Response from XL™.

Command: ALERTnHOLDOFF=XXX

Description: Set the hold off value for the nth ALERT alarm.

Parameters: 'n' ranges from 1-5. Hold off is an integer, representing milliseconds.

Response: None.

Example:
```
-> ALERT3HOLDOFF=300↵
└R└F
```
Command sent to XL™.
Response from XL™.

C.10 Logging Commands

Command: LOGGING?

Description: Read the logging status flag. If this flag is set, then the XL™ will log data to the internal data memory.

Parameters: None.

Response: An ASCII text string is returned with either "Yes" or "No ".

Example:

```
-> LOGGING?↵                    Command sent to the XL™.
Logging = Yes␍␊                 Response from the XL™.
```

Command: LOGGING=XXX

Description: Set the logging status flag.

Parameters: The flag can be set to either "Yes" or "No ".

Response: None.

Example:

```
-> LOGGING=NO↵                  Command sent to the XL™.
␍␊                              Response from the XL™.
```

Command: LOGCOLUMNnSOURCE?

Description: Read the source of the n^{th} column of data to be logged to the flash card.

Parameters: 'n' can range from 1-25, indicating the corresponding column.

Response: The source of the column corresponding to 'n' (e.g. Stage, PSI, Temp).

Example:

```
-> LOGCOLUMN2SOURCE?↵           Command sent to the XL™.
Log Column 2 Source = Date ␍␊   Response from the XL™.
```

Command: LOGCOLUMNnSOURCE=XXXXXX

Description: Set the source of the n^{th} data column going to the card.

Parameters: 'n' ranges from 1-25. The XL™ expects the new source of the data column corresponding to 'n' (e.g. Stage, PSI, Temp).

Response: None.

Example:

```
-> LOGCOLUMN1SOURCE=Time↵
┘R└F
```

Command sent to the XL™.
Response from the XL™.

Command: LOGCOLUMNnHEADER?

Description: Read the column header of the n^{th} column of data to be sent to the flash card. This value will be placed at the top of each column in the file logged to the card.

Parameters: 'n' can range from 1-25, indicating the corresponding column.

Response: The response will be an 8 character ASCII text string.

Example:

```
-> GOESRRCOLUMN3HEADER?↵
Log Column 3 Header = WaterLvl┘R└F
```

Command sent to the XL™.
Response from the XL™.

Command: LOGCOLUMNnHEADER=XXXXXXXX

Description: Set the n^{th} column header.

Parameters: 'n' ranges from 1-25. The new column header can be up to 8 characters of ASCII text. If the value is shorter that 8 characters, it will be padded on the right side with spaces so the length equals 8.

Response: None.

Example:

```
-> LOGCOLUMN4HEADER=Temp (C)↵
┘R└F
```

Command sent to the XL™.
Response from the XL™.

Command: LOGCOLUMNnSKIPRATE?

Description: Read the skip rate for the n^{th} column of data being logged to the card. The skip rate determines how often this value is generated in the data. A skip rate of 1 means the value will be in each row of data. A skip rate of 4 means the value will be in every 4^{th} row of data.

Parameters: 'n' can range from 1-25.

Response: The response is an ASCII text string with the skip rate for the n[th] column.

Example:
```
-> LOGCOLUMN2SKIPRATE?↵                Command sent to the XL™.
 Log Column 2 Skip Rate = 4CRLF        Response from the XL™.
```

Command: LOGCOLUMNnSKIPRATE=XX

Description: Set the skip rate for the n[th] column of data to be logged to the flash card.

Parameters: 'n' ranges from 1-25.

Response: None.

Example:
```
-> LOGCOLUMN1SKIPRATE=1↵               Command sent to the XL™.
CRLF                                   Response from the XL™.
```

Command: LOGCOLUMNnDIGITS?

Description: Read the number of digits to be displayed after the decimal point.

Parameters: 'n' can range from 1-25.

Response: The response is an ASCII text string with the number of digits for the n[th] column.

Example:
```
-> LOGCOLUMN2DIGITS?↵                  Command sent to the XL™.
 Log Column 2 Digits = 1CRLF           Response from the XL™.
```

Command: LOGCOLUMNnDIGITS=XX

Description: Set the number of digits to which the corresponding value is to be expressed when it is logged to the flash card.

Parameters: 'n' ranges from 1-25.

Response: None.

Example:
```
-> LOGCOLUMN1DIGITS=2↵                 Command sent to the XL™.
CRLF                                   Response from the XL™.
```

C.11 GOES Commands

Command: GOESMODE?

Description: Read the current mode of operation of the GOES radio.

Parameters: None.

Response: An ASCII text string is returned. The response will be “None ”, “Timed ”, “Random”, or “Both “.

Example:

```
-> GOESMODE?↵                          Command sent to the XL™.
GOES Mode = Timed ⌐R└F                 Response sent to the host system.
```

Command: GOESMODE=XXXXXX

Description: Set the current mode of GOES radio operation.

Parameters: The flag can be set to “None ”, “Timed ”, “Random”, or “Both “.

Response: None.

Example:

```
-> GOESMODE=RANDOM↵                    Command sent to the XL™.
⌐R└F                                   Response sent to the host system.
```

Command: GOESADDRESS?

Description: Read the 8 hexadecimal character DCP address that the GOES radio uses to interface with the collection satellite.

Parameters: None.

Response: An ASCII text string is returned with the DCP address. It is 8 characters long, and each character ranges from 0-9, A-F.

Example:

```
-> GOESADDRESS?↵                       Command sent to the XL™.
GOES Address = 14C028AF⌐R└F            Response sent to the host system.
```

Command: GOESADDRESS=XXXXXXXX

Description: Set the GOES DCP address.

Parameters: The address is to be 8 characters long. Each character ranges from 0-9,A-F.

Response: None.

Example:

```
-> GOESADDRESS=02A488C5↵
└R└F
```

Command sent to the XL™.
Response sent to the host system.

Command: GOESTIME?

Description: Read the time from the GOES radio. This time is used to synchronize with the satellite.

Parameters: None.

Response: An ASCII text string is returned with the GOES time.

Example:

```
-> GOESTIME?↵
GOES Time = 17:42:25└R└F
```

Command sent to the XL™.
Response sent to the host system.

Command: GOESTIME=hh:mm:ss

Description: Set the time of the GOES radio.

Parameters: The new time is to be entered in the format of HH:MM:SS, where H is hours, M is minutes, and S is seconds.

Response: None.

Example:

```
-> GOESTIME=08:30:00↵
└R└F
```

Command sent to the XL™.
Response sent to the host system.

Command: GOESSTCHANNEL?

Description: Read the 3 character Self Timed channel number.

Parameters: None.

Response: An ASCII text string is returned with the channel number. It is 3 characters long, and each character ranges from 0-9.

Example:

```
-> GOESSTCHANNEL?↵                Command sent to the XL™.
GOES ST Channel = 123CRLF         Response sent to the host system.
```

Command: GOESSTCHANNEL=XXX

Description: Set the channel number for the GOES radio to be used with the Self Timed mode.

Parameters: The channel number can range from 000-999.

Response: None.

Example:

```
-> GOESSTCHANNEL=321↵             Command sent to the XL™.
CRLF                              Response sent to the host system.
```

Command: GOESSTRATE?

Description: Read the Self Timed transmission rate. This is how often the GOES radio will transmit when in Self Timed mode.

Parameters: None.

Response: An ASCII text string is returned in the format of hh:mm:ss.

Example:

```
-> GOESSTRATE?↵                   Command sent to the XL™.
GOES ST Rate = 00:04:00CRLF       Response sent to the host system.
```

Command: GOESSTRATE=hh:mm:ss

Description: Set the transmission rate for the Self Timed mode of the GOES radio.

Parameters: The new rate is to be entered in the format of HH:MM:SS, where H is hours, M is minutes, and S is seconds.

Response: None.

Example:

```
-> GOESSTRATE=00:08:00↵           Command sent to the XL™.
CRLF                              Response sent to the host system.
```

Command: GOESSTOFFSET?

Description: Read the Self Timed transmission offset. This is a time value, indicating how long after midnight the first transmission takes place.

Parameters: None.

Response: An ASCII text string is returned in the format of hh:mm:ss.

Example:

```
-> GOESSTOFFSET?↵                    Command sent to the XL™.
GOES ST Offset = 00:02:00␍␊          Response sent to the host system.
```

Command: GOESSTOFFSET=hh:mm:ss

Description: Set the transmission offset for the Self Timed mode of the GOES radio.

Parameters: The new offset is to be entered in the format of HH:MM:SS, where H is hours, M is minutes, and S is seconds.

Response: None.

Example:

```
-> GOESSTOFFSET=00:03:00↵            Command sent to the XL™.
␍␊                                   Response sent to the host system.
```

Command: GOESSTFORMAT?

Description: Read the Self Timed transmission data format. This is the format in which the data is transmitted. The format can be either SHEF or BINARY.

Parameters: None.

Response: An ASCII text string is returned indication the format.

Example:

```
-> GOESSTFORMAT?↵                    Command sent to the XL™.
GOES ST Format = BINARY␍␊            Response sent to the host system.
```

Command: GOESSTFORMAT=XXXXXX

Description: Set the data format for the Self Timed mode of the GOES radio.

Parameters: The new format can be either “BINARY” or “SHEF”.

Response: None.

Example:
```
-> GOESSTFORMAT=SHEF↵
CRLF
```
Command sent to the XL™.
Response sent to the host system.

Command: GOESSTORDER?

Description: Read the Self Timed transmission data order. This is the order in which the data is transmitted. The order can be either SCAN or CHANNEL.

Parameters: None.

Response: An ASCII text string is returned indication the format.

Example:
```
-> GOESSTORDER?↵
GOES ST Order = SCAN  CRLF
```
Command sent to the XL™.
Response sent to the host system.

Command: GOESSTORDER=XXXXXX

Description: Set the data order for the Self Timed mode of the GOES radio.

Parameters: The new order can be either "CHANNEL" or "SCAN".

Response: None.

Example:
```
-> GOESSTORDER=CHANNEL↵
CRLF
```
Command sent to the XL™.
Response sent to the host system.

Command: GOESSTSENDSHEFCODES?

Description: Read the send SHEF codes flag. If this is set to "YES", each self timed GOES transmission will include SHEF code column headers.

Parameters: None.

Response: An ASCII text string is returned consisting of either "Yes" or "No ".

Example:
```
-> GOESSTSENDSHEFCODES?↵
GOES ST Send SHEF Codes = No CRLF
```
Command to XL™.
Response from XL™.

Command: GOESSTSENDSHEFCODES=XXX

Description: Set the flag to send SHEF codes with each self-timed transmission.

Parameters: The flag can be set to either “Yes” or “No “.

Response: None.

Example:

```
-> GOESSTSENDSHEFCODES=NO↵
┘R└F
```

Command sent to the XL™.
Response from the XL™.

Command: GOESSTAPPENDBATTERY?

Description: Read the append battery flag. If this flag is set, a battery reading will be appended to each GOES self-timed transmission.

Parameters: None.

Response: An ASCII text string is returned indicating whether or not the XL™ will append a battery reading when transmitting data. The response is either “Yes” or “No “.

Example:

```
-> GOESSTAPPENDBATTERY?↵
GOES ST Append Battery = Yes┘R└F
```

Command sent to the XL™.
Response from the XL™.

Command: GOESSTAPPENDBATTERY=XXX

Description: Set the append battery flag.

Parameters: The flag can be set to either “Yes” or “No “.

Response: None.

Example:

```
-> GOESSTAPPENDBATTERY=NO↵
┘R└F
```

Command sent to the XL™.
Response sent to the host system.

Command: GOESSTSCANSPERXMIT?

Description: Read the number of scans to be transmitted with each transmission.

Parameters: None.

Response: An ASCII text string with the number of scans per transmission.

Example: `-> GOESSTSCANSPERXMIT?↵` Command sent to the XL™.
`GOES ST Scans Per Xmit = 16CRLF` Response from the XL™.

Command: GOESSTSCANSPERXMIT=XXX

Description: Set the number of scans per transmission.

Parameters: This value can range from ???-???.

Response: None.

Example: `-> GOESSTSCANSPERXMIT=32↵` Command sent to the XL™.
`CRLF` Response sent to the host system.

Command: GOESSTBUFFERCOUNT?

Description: Counts the number of bytes currently in the GOES self timed buffer.

Parameters: None.

Response: An ascii text string is returned with an integer number representing the buffer count

Example: `-> GOESSTBUFFERCOUNT?↵` Command sent to the XL™.
`GOES ST Buffer Count = 512CRLF` Response from XL™.

Command: GOESSTCOLUMNnSOURCE?

Description: Read the source of the n^{th} column of data to be sent in the GOES self timed transmission.

Parameters: 'n' can range from 1-20, indicating the corresponding column.

Response: The source of the column corresponding to 'n' (e.g. Stage, PSI, Temp).

Example: `-> GOESSTCOLUMN2SOURCE?↵` Command sent to XL™.
`GOES ST Column 2 Source = Date CRLF` Response from XL™.

Command: **GOESSTCOLUMNnSOURCE**=XXXXXX

Description: Set the source of the nth data column of the self timed GOES transmission.

Parameters: 'n' ranges from 1-20. The XL™ expects the new source of the data column corresponding to 'n' (e.g. Stage, PSI, Temp).

Response: None.

Example:

```
-> GOESSTCOLUMN15SOURCE=Temp↵
⌐R⌊F
```

Command sent to the XL™.
Response from the XL™.

Command: **GOESSTCOLUMNnSHEFCODE?**

Description: Read the SHEF code of the nth column of data to sent in the GOES self timed transmission. The SHEF code is included as a description of the value. The SHEF code pneumonics are pre-defined. (See GOES section for details)

Parameters: 'n' can range from 1-20.

Response: The response is an ASCII text string with the SHEF code for the nth column.

Example:

```
-> GOESSTCOLUMN1SHEFCODE?↵
 GOES ST Column 1 SHEF Code = SH⌐R⌊F
```

Command.
Response.

Command: **GOESSTCOLUMNnSHEFCODE**=XX

Description: Set the SHEF code for the nth column of self timed GOES data.

Parameters: 'n' ranges from 1-20.

Response: None.

Example:

```
-> GOESSTCOLUMN3SHEFCODE=DT↵
⌐R⌊F
```

Command sent to the XL™.
Response from the XL™.

Command: GOESSTCOLUMNnSKIPRATE?

Description: Read the skip rate for the nth column of GOES self timed data. The skip rate determines how often this value is generated in the data. A skip rate of 1 means the value will be in each row of data. A skip rate of 4 means the value will be in every 4th row of data.

Parameters: 'n' can range from 1-20.

Response: The response is an ASCII text string with the skip rate for the nth column.

Example:

```
-> GOESSTCOLUMN1SKIPRATE?↵                    Command.
 GOES ST Column 2 Skip Rate = 4CRLF           Response.
```

Command: GOESSTCOLUMNnSKIPRATE=XX

Description: Set the skip rate for the nth column of self timed GOES data.

Parameters: 'n' ranges from 1-20.

Response: None.

Example:

```
-> GOESSTCOLUMN1SKIPRATE=1↵                   Command sent to the XL™.
CRLF                                          Response from the XL™.
```

Command: GOESRRCHANNEL?

Description: Read the channel to be used for GOES random transmissions.

Parameters: None.

Response: An ASCII text string is returned with the channel.

Example:

```
-> GOESRRCHANNEL?↵                            Command sent to the XL™.
GOES RR CHANNEL = 123CRLF                     Response sent to the host system.
```

Command: GOESRRCHANNEL=XXX

Description: Set the channel for random transmissions of the GOES radio.

Parameters: The channel is a 3 character decimal value, ranging from 000-200.

Response: None.

Example:
```
-> GOESRRCHANNEL=111↵
CRLF
```
Command sent to the XL™.
Response sent to the host system.

Command: GOESRRRATE?

Description: Read the random transmission rate. This determines the window during which the GOES radio will transmit after detecting a random trigger.

Parameters: None.

Response: An ASCII text string is returned in the format of hh:mm:ss.

Example:
```
-> GOESRRRATE?↵
GOES RR Rate = 00:00:30CRLF
```
Command sent to the XL™.
Response sent to the host system.

Command: GOESRRRATE=hh:mm:ss

Description: Set the transmission rate for the random report mode of the GOES radio.

Parameters: The new rate is to be entered in the format of HH:MM:SS, where H is hours, M is minutes, and S is seconds.

Response: None.

Example:
```
-> GOESRRRATE=00:01:00↵
CRLF
```
Command sent to the XL™.
Response sent to the host system.

Command: GOESRRTRIGGERMODE?

Description: Read the current mode of random report operation of the GOES radio. This value, along with trigger source, determines what conditions are checked when evaluating whether or not to generate a GOES random report.

Parameters: None.

Response: An ASCII text string is returned. The response will be "Always ", "> Set Pt", "< Set Pt", or "Delta ". (See GOES section for details)

Example:
```
-> GOESRRTRIGGERMODE?↵
GOES RR Trigger Mode = Always  CRLF
```
Command sent to XL™.
Response from XL™.

Command: **GOESRRTRIGGERMODE=XXXXXX**

Description: Set the current mode of GOES radio random report operation.

Parameters: The flag can be set to "Always ", "> Set Pt", "< Set Pt", or "Delta ".

Response: None.

Example:

`-> GOESRRTRIGGERMODE=> Set Pt↩`	Command sent to XL™.
`CRLF`	Response from XL™.

Command: **GOESRRTRIGGERSOURCE?**

Description: Read the source that will be evaluated when determining whether or not to trigger a random report.

Parameters: None

Response: The response will be an ASCII text string with the current RR trigger source (e.g. Stage, PSI, Temp).

Example:

`-> GOESRRTRIGGERSOURCE?↩`	Command sent to XL™.
`GOES RR Trigger Source = Stage CRLF`	Response from XL™.

Command: **GOESRRTRIGGERSOURCE**=XXXXXX

Description: Set the GOES random report trigger source.

Parameters: The new trigger source value for the random report (e.g. Stage, PSI, Temp).

Response: None.

Example:

`-> GOESRRTRIGGERSOURCE=PSI↩`	Command sent to the XL™.
`CRLF`	Response sent to the host system.

Command: GOESRRTRIGGERPOINT?

Description: Read the trigger point for the GOES random report. The value indicated by RR source is compared to this value. Depending on the RR trigger mode, different comparisons will be made. If the trigger conditions are met, a random report is generated.

Parameters: None.

Response: An ASCII text string is returned with a real number indicating the trigger point.

Example:

```
-> GOESRRTRIGGERPOINT?↵                    Command sent to XL™.
GOES RR Trigger Point = 2.5CRLF            Response from XL™.
```

Command: GOESRRTRIGGERPOINT=XXXXX

Description: Set the RR trigger point.

Parameters: The XL™ accepts a real number represented in ASCII text format.

Response: None.

Example:

```
-> GOESRRTRIGGERPOINT=3.3↵                 Command sent to XL™.
CRLF                                       Response from XL™.
```

Command: GOESRRFORCEXMIT?

Description: Immediately forces a GOES random report transmission.

Parameters: None.

Response: None

Example:

```
-> GOESRRFORCEXMIT↵                        Command sent to the XL™.
CRLF                                       Response from XL™.
```

Command: GOESRRBUFFERCOUNT?

Description: Counts the number of bytes currently in the GOES random report buffer.

Parameters: None.

Response: An ascii text string is returned with an integer number representing the buffer count

Example:
```
-> GOESRRBUFFERCOUNT?↵                Command sent to the XL™.
GOES RR Buffer Count = 232␍␊          Response from XL™.
```

Command: GOESRRCOLUMNnSOURCE?

Description: Read the source of the n[th] column of data to be sent in the GOES random report transmission.

Parameters: 'n' can range from 1-5, indicating the corresponding column.

Response: The source of the column corresponding to 'n' (e.g. Stage, PSI, Temp).

Example:
```
-> GOESRRCOLUMN2SOURCE?↵              Command sent to XL™.
 GOES RR Column 2 Source = Date ␍␊    Response from XL™.
```

Command: GOESRRCOLUMNnSOURCE=XXXXXX

Description: Set the source of the n[th] data column of the RR GOES transmission.

Parameters: 'n' ranges from 1-5. The XL™ expects the new source of the data column corresponding to 'n' (e.g. Stage, PSI, Temp).

Response: None.

Example:
```
-> GOESRRCOLUMN1SOURCE=Time↵          Command sent to the XL™.
␍␊                                    Response from the XL™.
```

C.11 Bubbler Commands

Command: BUBBLERATE?

Description: Read the H-355 bubble rate. The value indicates bubbles per minutes.

Parameters: None.

Response: An ASCII text string is returned with the bubble rate.

Example:
```
-> BUBBLERATE?↵                       Command sent to the XL™.
Bubble Rate = 45␍␊                    Response sent to the host system.
```

Command: BUBBLERATE=XXX

Description: Set the H-355 bubble rate.

Parameters: The range for the bubble rate is 30-120.

Response: None.

Example:

```
-> BUBBLERATE=60↵
┘R└F
```

Command sent to the XL™.
Response sent to the host system.

Command: PURGEPSI?

Description: Read the H-355 purge pressure. This is the pressure with which the Bubbler purges.

Parameters: None.

Response: An ASCII text string is returned indicating the purge pressure.

Example:

```
-> PURGEPSI?↵
Purge PSI = 40┘R└F
```

Command sent to the XL™.
Response sent to the host system.

Command: PURGEPSI=XX

Description: Set the H-355 purge pressure.

Parameters: The pressure range for Purge PSI is 15-80.

Response: None.

Example:

```
-> PURGEPSI=40↵
┘R└F
```

Command sent to the XL™.
Response sent to the host system.

Command: PURGETHRESHOLD?

Description: Read the H-355 purge threshold. The purge threshold is the pressure at which the H-355 automatically purges.

Parameters: None.

Response: An ASCII text string is returned indicating the purge threshold.

Example:

```
-> PURGETHRESHOLD?↵
Purge Threshold = 40ᶜᴿᴸᶠ
```

Command sent to the XL™.
Response sent to the host system.

Command: PURGETHRESHOLD=XX

Description: Set the H-355 purge threshold.

Parameters: The range for the purge threshold is 10-65.

Response: None.

Example:

```
-> PURGETHRESHOLD=50↵
ᶜᴿᴸᶠ
```

Command sent to the XL™.
Response sent to the host system.

Command: PURGEDURATION?

Description: Read the H-355 purge duration. This is the length of time (in seconds) for which the bubbler will purge.

Parameters: None.

Response: An ASCII text string is returned indicating the purge duration of the H-355.

Example:

```
-> PURGEDURATION?↵
Purge Duration = 120ᶜᴿᴸᶠ
```

Command sent to the XL™.
Response sent to the host system.

Command: PURGEDURATION=XXX

Description: Set the H-355 purge duration.

Parameters: The new time is to be entered in seconds, ranging from 30-240.

Response: None.

Example:

```
-> PURGEDURATION=90↵
ᶜᴿᴸᶠ
```

Command sent to the XL™.
Response sent to the host system.

Command: TANKPSI?

Description: Read the tank pressure in the H-355.

Parameters: None.

Response: An ASCII text string representing the current tank pressure.

Example:

```
-> TANKPSI?↵                    Command sent to the XL™.
Tank PSI = 24.3CRLF             Response sent to the host system.
```

Command: LINEPSI?

Description: Read the line pressure in the H-355.

Parameters: None.

Response: An ASCII text string representing the current line pressure.

Example:

```
-> LINEPSI?↵                    Command sent to the XL™.
Line PSI = 18.0CRLF             Response sent to the host system.
```

Command: PURGENOW

Description: Force the H-355 to purge immediately.

Parameters: None.

Response: None.

Example:

```
-> PURGENOW↵                    Command sent to the XL™.
CRLF                            Response sent to the host system.
```

Command: AUTOPURGE?

Description: Read the Auto Purge flag. The XL™ can trigger a purge when the H-355 pressure gets to a certain point. See Also: Purge Threshold.

Parameters: None.

Response: An ASCII text string is returned indicating whether or not the XL™ will trigger the H-355 to purge at the purge threshold. The response is either “Yes” or “No “.

Example: `-> AUTOPURGE?↵` Command sent to the XL™.
`Auto Purge = No ⌐R└F` Response sent to the host system.

Command: AUTOPURGE=XXX

Description: Set the Auto Purge flag.

Parameters: The flag can be set to either "Yes" or "No ".

Response: None.

Example: `-> AUTOPURGE=YES↵` Command sent to the XL™.
`⌐R└F` Response sent to the host system.

Command: TIMEDPURGE?

Description: Read the Timed Purge value. Based on this value, the XL™ will trigger an automatic purge. If set to 1, the XL™ purges the bubbler daily; if set to 2, the XL™ will purge every 2 days; if set to 3, every 3 days, and so on. If this value is set to 0, the XL™ will not purge the bubbler on a timed basis.

Parameters: None.

Response: An ASCII text string is returned, indicating the timed purge interval (in days).

Example: `-> TIMEDPURGE?↵` Command sent to the XL™.
`Timed Purge = 1⌐R└F` Response sent to the host system.

Command: TIMEDPURGE=XXX

Description: Set the Timed Purge value. This is the number of days between automatic purges. If this value is set to 0, the XL™ will not trigger a timed purge.

Parameters: The new timed purge value (in days).

Response: None.

Example: `-> TIMEDPURGE=2↵` Command sent to the XL™.
`⌐R└F` Response sent to the host system.

Command: LASTPURGE?

Description: Read the time at which the H-355 last purged.

Parameters: None.

Response: An ASCII text string is returned indicating the time of the most recent purge.

Example:

`-> LASTPURGE?↵`	Command sent to the XL™.
`Last Purge = 08:21:48CRLF`	Response sent to the host system.

C.12 Miscellaneous Commands

Command: OFF

Description: Turn the XL™ off. This command is also automatically executed if there has been no user interaction for the last two minutes.

Parameters: None:

Response: The text string 'Off' will be sent back to the host. This is useful when the unit times out and turns off automatically.

Example:

`-> OFF↵`	Command sent to the XL™.
`OFFCRLF`	The unit powers down to a low power mode.

Command: ECHOOFF

Description: Turn off the echoing of the commands. This is most useful if a user develops there own interface to the logger. Not having to work with the echoing command makes the design of the custom interface much simpler. To turn the echo back on use the OFF command and then turn the logger back on.

Parameters: None.

Response: None.

Example:

`-> ECHOOFF↵`	Command sent to the XL™.
	The logger will no longer echo key presses to the host system.

C.13 Advanced Commands

Command: MENU

Description: Display a menu structure on the host system. This will use a VT52 terminal emulation mode. When leaving the menu the XL™ will turn off. This will be the normal mode of operation for human interaction. Chapter 3 is devoted completely to this command.

Parameters: None.

Response: None.

Example: `-> MENU↵` Command sent to the XL™
The logger displays a menu structure on the host systems terminal screen.

Command: TESTSCAN

Description: This command causes the XL™ to execute the scan list. Any values returned will depend on the scan list. If the scan list utilizes the scan list command "TXA" then the value in the A register will be returned. See the description for the "TXA" command in chapter 7 for more information on possible uses.

Parameters: None.

Response: The response will vary based on the scan list commands. The response may or may not be terminated with a carriage return, line feed.

Example: Suppose the scan list commands were as follows:

```
SDIM     3                  ; read SDI12 sensor at address 3
TXA                         ; it has 3 parameters, send the first value
PULL                        ; put next value in the A register.
TXA                         ; transmit the second value.
PULL                        ; put the third value in the A register.
TXA                         ; send the third value.
TCR                         ; send a carriage return, and line feed.
```

Now if the SCN command was sent to the logger.

`-> TESTSCAN↵` Command sent to the XL™.
`23.153 459.21 9.1388⌐R└F` Response sent to the host system.