Control Protocol: Jupiter 4, Jupiter 8, Jupiter 12

34 min read

Introduction

About this Document

The purpose of this document is to provide a technical understanding of the Symetrix Control Protocol for Jupiter DSPs. It will define and illustrate the protocol used to communicate with the Jupiter products via a third-party interface.

Jupiter devices can be controlled by third-party controllers such as certain AMX or Crestron models, or any Ethernet equipped device that can be adapted to this protocol. The protocol consists of humanly readable text commands and responses. It is based on the SymNet Control Protocol (described in a separate document) and inherits many of the features from that system.

Control is achieved by using a scheme of pre-assigned controller numbers. Nearly anything that can be adjusted from the Jupiter software can be controlled externally by referencing the appropriate controller number. The controller numbers for each Jupiter App may be browsed in the Custom Preset Parameter Browser. Refer to the Jupiter software help file for more information.

Conventions Used in this Document

- A dollar sign ($) preceding a set of alphanumeric characters denotes a hex value. All other number values should be considered decimal values. Example: “$A0” represents the decimal value of “160.”
- Values enclosed in [square brackets] are optional parameters and do not need to be include. If omitted, default values will be used as described for each command.
- The term “control application” is used to refer to the Windows-based graphical user interface software provided by Symetrix to configure Jupiter devices.

General Notes

Connections

All Jupiter products are equipped with Ethernet ports. The same port is used for host control (Jupiter software) and third-party control (with this control protocol).

Ethernet Port Configuration

Generally, no special configuration is required for the Ethernet port. The single Ethernet port on the device may be used for both the control application and for external control. Take note of the device's IP address (listed in the Connection Wizard), as you will need to send all commands to this address. The commands Set Quiet Mode and Set Echo affect the Ethernet port. The device's default settings (Quiet Mode ON, Echo OFF) are typical for most applications, so most users will not need to know about these commands. However, they are also documented for reference.

Ethernet Control

The Ethernet protocol allows the use of the existing human-readable command language over an Ethernet network. The protocol is similar to Telnet in use. However, instead of using TCP as Telnet does, it uses UDP. And, it does not use any of the options or escape sequence found in Telnet. To use this feature, command strings following the command language can be sent as the payload of a UDP packet. The following rules should be observed in sending commands:

- Commands should be sent to UDP port number 48630 of the proper Symetrix device's IP address. The IP address may be found using the Connection Wizard.
- Commands should be formatted exactly as defined in this document.
- Command strings may or may not include a zero termination character.
- Commands should not be broken up across multiple packets
- If high reliability communications are required, responses to commands should be analyzed for success.

Responses to commands will exhibit the following behavior:

- Responses to each command issued are returned in a single packet unless the response is larger than a single packet can hold. Responses will not have any single carriage return-terminated line broken up across packets unless there is no carriage return in the response.
- Responses are returned to the IP address and source port number that sent the packet.
- Responses follow the configuration of the port (echo mode, quite mode and deaf mode).
- Responses do not include a zero-termination character.

- All transmissions originating from Symetrix devices will either be responses to commands or pushed data.

Each command sent to a Symetrix device contains information in the Ethernet packet header as to who sent the command, and hence, where a response will be sent. This source information is saved when a packet is received by a Symetrix device. All responses go to the last received IP address and port and this IP address and port number are saved in non-volatile memory across power cycles.

Until the first command is received, responses will not to know where they are supposed to be sent. This normally not an issue as communication from the Symetrix device is generally a response to a command. However, if the Symetrix device is set up to push control data, it will also be pushed out this UDP port. If no valid packets have ever been received by a Symetrix device, pushed data will not be sent out the Ethernet port.

RS-485 Control

RS485 control is generally done using one or more of the Symetrix ARC (Adaptive Remote Controller) devices. Further discussion of RS485 and the ARCs can be found on the Symetrix web site.

Parameter Notes

Faders

Faders can be controlled to the limits of their minimum and maximum values shown in the control application screens. A controller position of zero (0) will cause the minimum fader position to be realized. A controller position of 65535 will cause the maximum fader position to be realized. Increasing positions will move the fader linearly in dB.

Most volume faders have a range of -72 dB to +12 dB. In these cases, the following formula can be used to convert from controller position to dB:

Volume dB = -72 + 84*(CONTROLLER POSITION/65535)
If CONTROLLER POSITION = 0, Volume dB = OFF

Note that some faders have a different range than –72 to +12 dB. In this case, the formula will depend upon the actual fader range. The more general formula is shown below:

Volume dB = MINIMUM VALUE + (MAXIMUM VALUE – MINIMUM VALUE)*(CONTROLLER POSITION/65535)

Where MINIMUM VALUE is the fader's lower limit in dB and MAXIMUM VALUE is the fader's upper limit in dB.

Buttons

Buttons such as a mute or bypass can be controlled similarly with controller positions by sending the minimum value (0) to turn the switch off (button not pushed) and sending the maximum value (65535) to turn the switch on (button pushed). In some cases, the buttons use negative logic, i.e. 0 turns it on. These exceptions are noted in the Appendix for each product.

Input Selectors

A value of zero (0) will select the first input or output and a value of (65535) will select the last input or output. Other values are selected by sending evenly spaced (linear) values as shown by the formula below:

Controller Value = (INPUT NUMBER – 1)*65535/(NUMBER OF INPUTS – 1)

Meters

Meters can be read via Ethernet. The read back value will be linear in dB with 65535 representing +24 dBu (0 dBFS) and 0 representing -48 dBu (-72 dBFS) (or less). The formula below can be used to calculate a dB reading from a controller value:

Level dBu = 72*(CONTROLLER VALUE/65535) – 48
If CONTROLLER VALUE = 0, Level dBu <= -48 dBU

Input and output meters in some other modules such as Compressors, and AGCs can also be read via Ethernet. In this case, the read back value will be linear in dB with 65535 representing the maximum value shown on the meter and 0 representing the minimum value shown on the meter (or less). The formula below can be used to calculate a dB reading from a controller value:

Level dB = (MAXIMUM VALUE – MINIMUM VALUE)*(CONTROLLER VALUE/65535) + MINIMUM VALUE
If CONTROLLER VALUE = 0, Level dB <= MINIMUM VALUE

Note: Meters are a “read-only” parameter. Attempting to change the meter value will have no effect.

Other Parameters

Many other parameters such as compression ratios, delay times, EQ settings, and pans can also be controlled externally. For other parameter types, as in the above examples, sending a value of zero (0) will set the parameter to its minimum value and sending a value of (65535) will set it to its maximum value. Ratios, frequencies, width/Q, and

attack/release/hold times all use a logarithmic scale. Pans and delay times use a linear scale. Quantities expressed in dB such as gains, volumes, thresholds, and depths are linear in dB. When in doubt, experiment by changing a value from the control application and reading it back via Ethernet.

Getting Started

Protocol

The Control Protocol is a text-based (ASCII string) protocol. Commands are sent with simple character strings with terms separated by spaces and completed with a carriage return character (ASCII code decimal 13 or hex $0D). The general form for commands is:

<COMMAND> <PARAMETER> <PARAMETER> ... <CR>

A white space character (space, tab, etc.) must be included between the command and each parameter. Extra white space characters can be sent for readability if desired. In this document a single space will be used. If a command is accepted, the device will respond to each command with an acknowledgement string whose syntax varies with each command.

Control Commands

(CS) Controller Set

Use this command to move a controller position to a new absolute value. The command must specify the controller number and the new controller position. The syntax of the command is:

CS <CONTROLLER NUMBER> <CONTROLLER POSITION> <CR>

Where <CONTROLLER NUMBER> is the decimal controller number (1-10000) listed in the Appendix for each product, and <CONTROLLER POSITION> is a 16-bit number in decimal (0-65535).

If the command is accepted, the device will respond with the string: *ACK <CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*
A typical reason for failure is that the specified controller number does not exist.

(CC) Change Controller

Use this command to move a controller to a new relative value. This command will increment or decrement a controller by a specified amount. The command must specify

the controller number, whether it should be incremented or decremented, and the amount to change by. The syntax of the command is:

CC <CONTROLLER NUMBER> <DEC/INC> <AMOUNT> <CR>

Where <CONTROLLER NUMBER> is the decimal controller number (1-10000) listed in the Appendix for each product, <DEC/INC> is 0 to decrement and 1 to increment, and <AMOUNT> is the amount to increment or decrement (a decimal number, 0-65535). If the amount to be decremented or incremented causes the parameter to exceed its minimum or maximum value, the value will be limited to its minimum or maximum value. For example, if you increment a parameter by 10 and its current value is 65530, the new value will be limited to 65535.

If the command is accepted, the device will respond with the string: *ACK <CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*
A typical reason for failure is that the specified controller number does not exist.

(GS) Get Controller

This command will return the controller position (value) associated with a specific controller number. The command must specify the controller number. The syntax of the command is:

GS <CONTROLLER NUMBER> <CR>

Where <CONTROLLER NUMBER> is the decimal controller number (1-10000) listed in the Appendix for each product.

If the command is accepted, the device will respond with the string: *<CONTROLLER NUMBER> <CR>*
Where controller position is a 16-bit number in decimal (0-65535).

If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*
A typical reason for failure is that the specified controller number does not exist.

If the value being requested is a button that only has two states, the returned values will be either 0 or 65535, regardless of the actual value sent to the controller. For example, assume controller number 1 controls a mute button. If you send CS <1> <754>, and then GS <1>, it will return 0, not 754. More generally, if the parameter you are controlling has granularity coarser than the 16-bit values used, the returned values will be quantized to the

granularity of the parameter. Controls where you might observe this effect are buttons as mentioned above and input selectors.

(GS2) Get Controller with Controller Number

This command will return the controller number with controller position (value) associated with it together in the same string. This command is provided at the request of AMX/Crestron programmers to make it easier to interpret and parse returned controller positions. The command must specify the controller number. The syntax of the command is:

GS2 <CONTROLLER NUMBER><CR>

Where <CONTROLLER NUMBER> is the decimal controller number (1-10000) listed in the Appendix for each product.

If the command is accepted, the device will respond with the string: <*CONTROLLER NUMBER*><*CONTROLLER POSITION*> <*CR*>

Where <CONTROLLER POSITION>is a 16-bit number in decimal (0-65535)

If the command is interpreted but fails for any reason the device will respond with the string: *NAK* <*CR*>
A typical reason for failure is that the specified controller number does not exist.

(GSB) Get Controller Block

This command will return the controller position (value) of a specific range of consecutive controller numbers. The command must specify the starting controller number and the number of consecutive controllers to return. The syntax of the command is:

GSB <CONTROLLER NUMBER> <BLOCK SIZE> <CR>

Where <CONTROLLER NUMBER> is the decimal controller number (1-10000) listed in the Appendix for each product and <BLOCK SIZE> is the number of consecutive controllers. Note that <BLOCK SIZE> can be at most 256.

If the command is accepted, the device will respond with the string:

<*CONTROLLER POSITION1*> <*CR*>
<*CONTROLLER POSITION2*> <*CR*>
<*CONTROLLER POSITION3*> <*CR*>
…
<*CONTROLLER POSITIONn*> <*CR*>

Where <CONTROLLER POSITIONn> is a 16-bit number in decimal (0-65535), or -1 if a controller does not exist. The values will always be five digits, with leading zeros added as necessary (e.g. 7 would be returned as 00007 <CR> and -1 would be returned as -0001 <CR>.)

If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*
A typical reason for failure is that the requested block size is larger than 256. For more information and tips on reading back controller numbers, see the GS command.

Example command sent:
GSB 9 3 <CR>

Example Response:
32321 <CR>
00256 <CR>
00003 <CR>

(GSB2) Get Controller Block with Controller Number

This command will return the controller number with controller position (value) associated with it for a specific range of consecutive controller numbers. The command is very similar to GSB described above, but the return string may be easier to process in some systems. The command must specify the starting controller number and the number of consecutive controllers to return. The syntax of the command is:

GSB2 <CONTROLLER NUMBER> <BLOCK SIZE> <CR>

Where <CONTROLLER NUMBER> is the decimal controller number (1-10000) listed in the Appendix for each product and <BLOCK SIZE> is the number of consecutive controllers. Note that <BLOCK SIZE> can be at most 256.

If the command is accepted, the device will respond with the string:

#<CONTROLLER NUMBER1>=<CONTROLLER POSITION1> <CR>
#<CONTROLLER NUMBER2>=<CONTROLLER POSITION2> <CR>
#<CONTROLLER NUMBER3>=<CONTROLLER POSITION3> <CR>
...
#<CONTROLLER NUMBERn>=<CONTROLLER POSITIONn> <CR>

Where <CONTROLLER NUMBERn> is the decimal controller number (1-10000) listed in the Appendix for each product and <CONTROLLER POSITIONn> is a 16-bit number in decimal (0-65535), or -1 if a controller does not exist. The values for the controller number and

position will always be five digits, with leading zeros added as necessary (e.g. 7 would be returned as 00007 and -1 would be returned as -0001).

If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*
A typical reason for failure is that the requested block size is larger than 256. For more information and tips on reading back controller numbers, see the GS command.

Example command sent:
GSB2 9 3

Example Response:
#00009=32321 <CR>
#00010=00256 <CR>
#00011=00003 <CR>

(GPR) Get Preset

This command will return the last preset that was loaded. The syntax of the command is:

GPR D <CR>

If the command is accepted, the device will respond with the string: *PrstD=<PRESET NUMBER> <CR>*

The <PRESET NUMBER> return value will be 0-50. A return value of 0 indicates that no preset has been recalled. The value for the preset number will always be 4 digits, with leading zeros added as necessary (e.g. 7 would be returned as 0007).

If the command is interpreted but fails for any reason, the device will respond with the string: *NAK <CR>*

(LP) Load Preset

This command will load the specified preset (1-50). The syntax of the command is:

LP <PRESET NUMBER> <CR>

Where <PRESET NUMBER> = 1-50 as defined in the control application.

If the command is accepted, the device will respond with the string: *ACK <CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*
A typical reason for failure is that the specified preset has not been defined.

(FU) Flash Unit

This command momentarily flashes the front panel LEDs of the device. This command can be used as a quick test to verify communications. The syntax of the command is:

FU <CR>

If the command is accepted, the LEDs will flash and the device will respond with the string: *ACK <CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*

LEDs on devices other than the one to which you are physically connected can be flashed by using the Set Device command.

Push Commands

Symetrix devices can send out unsolicited Ethernet data. All parameters that can be externally controlled can be set up to automatically send out their values whenever they change. This method, referred to as pushing data, can be used instead of or in addition to polling (asking for data). When using this feature, take care that your system can handle the volume of data you set up and that it can differentiate between responses to commands and unsolicited data. Commands used to control the push feature are described below. Also, the following questions and answers provide a detailed discussion of this feature, including realworld problems and solutions.

When is data pushed?

For data to be pushed 1) the push feature must be globally enabled and 2) individual parameters must be enabled to push using the Push Enable command. Then, the controller value will be sent out 1) whenever the control's underlying parameter changes or 2) when a refresh command is issued. Regardless of if the parameter change is made via the control application, RS-485, preset recall, analog control, or any other method, the data will be pushed. This means for example that if your control system changes a controller value set up for push, you will immediately receive notification of that change.

Where is the data pushed?

The data is sent out the Ethernet port of the Symetrix device.

What does the pushed data look like?

The format for unsolicited or "push" data is the same as the GSB2 command. Strings consist of the controller number and its value in the following format:

#<CONTROLLER NUMBER>=<CONTROLLER POSITION> <CR>

Where <CONTROLLER NUMBER> is the decimal controller number (1-10000) listed in the Appendix for each product and <CONTROLLER POSITION> is a 16-bit number in decimal (0-65535). The values for the controller number and position will always be five digits, with leading zeros added as necessary (e.g. 7 would be returned as 00007).

Up to 64 strings, separated with a as shown, may be sent out together. Example:

#00007=12321 <CR>
#00324=00128 <CR>
#10000=65535 <CR>

Should I use the push feature or poll for parameter changes?

The decision is up to you. Use whichever method makes more sense for your application and control system. In fact, some "control-only" applications may not need to use either. For example if the only thing controlling the device is your control system, then you know when anything changes. Manual polling is often simpler to implement initially because all data from the device is a direct response to command you send it, simplifying parsing. However, in situations where a large number of parameters that change infrequently need to be monitored, pushing may make more sense. You may also prefer the convenience of not needing to set up a timer to continually poll parameters for changes. Use whatever method is appropriate for your situation.

How often is data pushed?

If there is data to be pushed, it is normally sent out every 100 milliseconds. This is called the push interval. While 100 ms is the default, the push interval can be changed via a Set Push Interval command.

Can I push meter data?

Yes, meters can be enabled for push. Keep in mind that with normal audio signals connected to a meter, the meter value will most likely be changing constantly, so you will typically see the meter data being pushed at every 100 ms interval. However, a Set Push Threshold command can be used to prevent pushes until the data differs by a specified amount (by default, this amount is 1).

How can I control the amount of data pushed?

There are several methods for controlling pushed data. First, since pushed data is enabled on a per-control basis, your first line of defense is to limit it to only certain controls. Second, pushing can be globally turned on and off using a command. Third, pushing can be enabled for just a range of controller numbers. Fourth, the Set Push Threshold command

can be used to prevent pushes until the data changes by a specified amount. Fifth, the Set Push Interval command controls how often the data is pushed, useful for meters and other data that changes frequently. Finally, the Push Refresh and Push Clear commands provide additional methods of control.

I want to refresh everything to make sure my control system is synchronized to the hardware. How can I receive all data even if it hasn't changed?

Use the Push Refresh command. Alternatively, you could use the Get Controller commands to manually ask for the controls you are concerned with.

Sometimes my control system turns off push for an extended period of time. When I turn it on, will I be notified of all changes that occurred while push was turned off?

Yes, by default, all changes made while push was off will be immediately reported as soon as it is turned on. This applies to both turning push off globally or for individual controllers via the Push Disable command. Take care that your system can handle the potentially large amount of data that can be generated. It may be helpful to "gradually" turn on the push feature, enabling a small range of controller numbers at once. You can also use the Push Clear command to deal with this scenario. It allows you to effectively ignore all previous unreported changes.

What is the difference between the Global Push Enable/Disable (PU) command and the Push Enable (PUE) and Push Disable (PUD) commands? Why are there 2 different ways to specify a range of controllers?

The Global Push Enable/Disable command can be used to completely turn off push, or turn on push for all or a single contiguous range of controller numbers. In contrast, the Push Enable/Disable command allows much finer control. Individual (non-contiguous) controllers can be turned on and off, hence multiple ranges are supported.

The reason both methods are provided is for backwards compatibility. The less flexible "single range" global PU command was added first. Later, the more flexible PUE and PUD commands were added as an enhancement. The older global method was left in so existing programs wouldn't need to be modified. We recommend that you use either one system or the other exclusively. Do not combine them. New designs should use the PUE and PUD commands and never use the PU command with a range specified.

How does push work at power-up?

When a device is first powered up, push is globally turned on but all controllers are individually disabled. All controller numbers are assumed to have changed. This means that after power-up, the first time you enable a controller to push, you will immediately

receive its current value. This can be prevented by issuing a Push Clear command before issuing the Push Enable command.

I'm not receiving unsolicited data. Any suggestions for troubleshooting?

First of all, make sure that general communication is working between your control system and the Symetrix Ethernet port. Make sure you can send commands and receive ACK messages. Try the Flash Unit command.

For Ethernet, make sure the Ethernet port is connected to the same network as the control system. Verify the connection LED on the Ethernet jack and/or switch is lit. Verify you can "ping" the device using its IP address.

Make sure the push feature has been globally enabled using the Global Push Enable/Disable command. Push is globally enabled on power-up, but may be turned off via Ethernet. Power cycling the device is a quick way to verify this.

Make sure the individual controllers have been enabled using the Push Enable command. Push is disabled for all controllers on power-up, and must be turned on via Ethernet. Sending a PUE command is a quick way to enable all controllers.

Make sure the parameter to be pushed is changing. Change the parameter via the control application, a Controller Set command, or other method. You can also use the Push Refresh command to force the data to be sent. If you have changed the push threshold, make sure the parameter is changing by an amount larger than the threshold.

For Ethernet, the device needs to know the proper IP address to send the data. Make sure at least one command has been sent from the control system to the device. If the control system ever changes IP addresses, another command must be sent to establish the new address.

What are the limitations of this feature?

If multiple parameters change at the same time, up to 64 controller numbers will be sent out during each push interval (default 100 ms) until all have been sent out. If a large amount of data is being pushed, we recommend you verify your system can support the amount of data being pushed.

Commands Related to Push

(PU) Global Push Enable/Disable

This command enables or disables the push feature. When enabling, a range of controllers can be specified to allow pushing only certain values. Disabling is always global and prevents any unsolicited data from being pushed. The syntax of the command is:

PU <ON/OFF> [<LOW> [<HIGH>]] <CR>

Where <ON/OFF> is 0=OFF and 1=ON. <LOW> is the optional lowest controller number to push (only valid when enabling) and <HIGH> is the optional highest controller number to push (only valid when enabling). <LOW> and <HIGH> are both decimal controller numbers (1-10000) listed in the Appendix for each product. If no controller numbers are specified, the entire range of 1-10000 will be enabled for push. If only one controller number is specified, it is assumed to be the <LOW> value and the range from that number up to 10000 will be pushed. If two controller numbers are specified, the range formed by those values (including the values themselves) will be enabled for push. <LOW> must be less than or equal to <HIGH>. When enabling, the range specified overrides any previous ranges, i.e. it replaces the range, rather than adding to it.

If the command is accepted, the device will respond with the string: *ACK <CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*

At power-on, push is always enabled. Remember that individual controller numbers must be enabled using the Push Enable command as well. Data is pushed whenever a change in that controller occurs or if forced to refresh using the Push Refresh command.

Note: Global Push Enable with a range specified, e.g. PU 1 100 200<CR> is not recommended. Instead, we recommend always globally enabling the entire range using PU 1<CR> and using the Push Enable command for individual control.

(PUE) Push Enable

This command enables the push feature for an individual controller or range of controllers. The syntax of the command is:

PUE [<LOW> [<HIGH>]] <CR>

Where <LOW> is the optional lowest controller number to push and <HIGH> is the optional highest controller number to push. <LOW> and <HIGH> are both decimal controller numbers (1-10000) listed in the Appendix for each product. If no controller numbers are specified, the entire range of 1-10000 will be enabled for push. If only one controller number is specified, only that controller number is enabled. If two controller numbers are specified, the range formed by those values (including the values themselves) will be

enabled for push. <LOW> must be less than or equal to <HIGH>. Multiple PUE commands can be used to enable non-contiguous controller numbers since changes are additive.

If the command is accepted, the device will respond with the string: *ACK <CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*

At power-on, push is disabled for all controllers in Jupiter devices. Data is pushed whenever a change in an enabled controller occurs or if forced to refresh using the Push Refresh command. Changes that happen while a control is disabled will be pushed immediately upon enabling that control. The Push Disable command is the inverse of this command and provides a way to turn off controllers for push.

(PUD) Push Disable

This command enables the push feature for an individual controller or range of controllers. The syntax of the command is:

PUD [<LOW> [<HIGH>]] <CR>

Where <LOW> is the optional lowest controller number to stop pushing and <HIGH> is the optional highest controller number to stop pushing. <LOW> and <HIGH> are both decimal controller numbers (1-10000) listed in the Appendix for each product. If no controller numbers are specified, the entire range of 1-10000 will be disabled for push. If only one controller number is specified, only that controller number is disabled. If two controller numbers are specified, the range formed by those values (including the values themselves) will be disabled for push. <LOW> must be less than or equal to <HIGH>. Multiple PUD commands can be used to disable non-contiguous controller numbers since changes are subtractive.

If the command is accepted, the device will respond with the string: *ACK <CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*

At power-on, push is disabled for all controllers in Jupiter devices. The Push Enable command is the inverse of this command and provides a way to turn on controllers for push.

(GPU) Get Push-enabled Controllers

This command returns a list of all controllers currently enabled for push. A range may optionally be specified to limit the display to controllers enabled for push within that range. The syntax of the command is:

GPU [<LOW> [<HIGH>]] <CR>

Where <LOW> is the optional lowest controller number to inquire about and <HIGH> is the optional highest controller number to inquire about. <LOW> and <HIGH> are both decimal controller numbers (1-10000) listed in the Appendix for each product. If no controller numbers are specified, the entire range of 1-10000 will be inquired about. If only one controller number is specified, it is assumed to be the <LOW> value and the range from that number up to 10000 will be inquired about. If two controller numbers are specified, the range formed by those values (including the values themselves) will be inquired about. <LOW> must be less than or equal to <HIGH>.

If the command is accepted, the device will respond with a list of enabled controller numbers separated by <*CR*>.
If no controllers are enabled, it returns the string: *ACK* <*CR*>
If the command is interpreted but fails for any reason the device will respond with the string: *NAK* <*CR*>

Special case: Entering GPU 0<CR> will return a list settings related to push. It begins with Global=<0/1> to show if push is globally enabled (1) or disabled (0). This is followed by five 5-digit values showing the settings of 1) the global lower limit, 2) the global upper limit, 3) the threshold for parameters, 4) the threshold for meters, and 5) the push interval in milliseconds. The default printout would look like this:

Global=1 <CR>
00001 10000 00001 00001 00100 <CR>

(PUR) Push Refresh

This command causes data to be pushed immediately even if it hasn't changed (assuming push is enabled). This may be useful when trying to synchronize a control system to the device. A range of controllers can be specified to refresh only certain values. The syntax of the command is:

PUR [<LOW> [<HIGH>]] <CR>

Where <LOW> is the optional lowest controller number to refresh and <HIGH> is the optional highest controller number to refresh. <LOW> and <HIGH> are both decimal controller numbers (1-10000) listed in the Appendix for each product. If no controller numbers are specified, the entire range of 1-10000 will be refreshed. If only one controller number is specified, it is assumed to be the <LOW> value and the range from that number up to 10000 will be refreshed. If two controller numbers are specified, the range formed by

those values (including the values themselves) will be refreshed. <LOW> must be less than or equal to <HIGH>.

If the command is accepted, the device will respond with the string: *ACK <CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*

At power-on, all controller values are assumed to have changed, so it acts as if a full refresh was performed. In addition, push must be enabled for the range of controllers you are refreshing (see Push Enable). Controller numbers that don't meet this criterion will not be affected by the Push Refresh command. In other words, if a controller is not enabled for push, refreshing it won't cause the value to be pushed even if that controller is later enabled. The controller must be enabled for push at the time the Push Refresh command is issued.

(PUC) Push Clear

This command causes previous changes in data to be ignored and not pushed. It may be desirable to issue this command when first enabling push to prevent being swamped by the flood of incoming data. A range of controllers can be specified to clear only certain values. The syntax of the command is:

PUC [<LOW> [<HIGH>]] <CR>

Where <LOW> is the optional lowest controller number to clear and <HIGH> is the optional highest controller number to clear. <LOW> and <HIGH> are both decimal controller numbers (1-10000) listed in the Appendix for each product. If no controller numbers are specified, the entire range of 1-10000 will be cleared. If only one controller number is specified, it is assumed to be the <LOW> value and the range from that number up to 10000 will be cleared. If two controller numbers are specified, the range formed by those values (including the values themselves) will be cleared. <LOW> must be less than or equal to <HIGH>.

If the command is accepted, the device will respond with the string: *ACK <CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*

It may be useful to issue this command if push has been disabled for a long time and then is about to be re-enabled. Otherwise, you will immediately receive notification for all changes that occurred during the disabled time.

(PUI) Set Push Interval

This command changes the minimum length of time between consecutive pushes of data. (See "How often is data pushed?" above for more information.) At power-up, this value defaults to 100 milliseconds. The syntax of the command is:

PUI <MILLISECONDS> <CR>

<MILLISECONDS> is the push interval in milliseconds, between 20 ms and 30,000 ms (30 seconds).

If the command is accepted, the device will respond with the string: *ACK <CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*

While setting a short interval can speed up the push response, it may have a negative impact on overall system performance. The shorter the interval, the more time will be spent looking for push data. This can slow down responses to other commands and the control application. Therefore, we recommend using the longest interval that is practical, especially if data is being pushed while the control application is on-line. The default value of 100 milliseconds usually provides a good compromise between prompt reports of changing data and overall system performance.

(PUT) Set Push Threshold

This command changes the push threshold value. Recall that data is only pushed when it changes. The threshold is the amount a value must change from the previous push before it is pushed again. For example, if a controller value was 10,000 and the threshold was 1,000, the data would not be pushed again until the value rose to at least 11,000 or fell to 9,000 or below.

The device actually maintains two different thresholds: one for parameter data such as faders and buttons, and another for meters (including LEDs). These two thresholds can be set to the same value or be different. It may be desirable to use a fairly large threshold for meters to avoid constant pushing of values. The power-on default for both of these values is 1.

The syntax of the command is:

PUT [<PARAMETER THRESH>] [<METER THRESH>]] <CR>

Where <PARAMETER THRESH> is the optional threshold for parameters other than meters (e.g. faders and buttons) and <METER THRESH> is the optional threshold for meters. Both values must be between 0 and 65535. If neither threshold is specified, both thresholds are

set to the default of 1. If only one threshold is specified, that value is used for both the parameter and meter thresholds.

If the command is accepted, the device will respond with the string: *ACK <CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*

Technical Note: The threshold is a "greater than or equal to" type parameter, meaning it must be met (or exceeded) to trigger a push. For example: if the threshold is 1 and the last value pushed was 10,000, then a new value of 10,001 or 9,999 would cause a push to occur. Note that it is possible to set the threshold to zero. In this case, the value will be pushed if there is any change at all to the underlying DSP variable – even if the change is so small that the pushed controller value is identical (which may happen due to the limited resolution of the 16-bit controller value scheme).

Setup Commands

Note: If you ever find yourself in a situation where you are not sure of the accessory controller port settings, you can use the control application to change the settings with the Accessory Port Settings dialog under the Tools menu. Alternatively, the rear panel reset button can be used to return the settings to factory defaults. However, that should be only used as a last resort since it also resets many other things.

(SQ) Set Quiet Mode

The Set Quiet Mode command controls the text output of the control port during responses. When quiet mode is turned on, it restricts the output to just ACK, NAK or simple values. All command descriptions above assume that quiet mode is turned ON. Quiet mode ON should generally be used for normal operation.

When quiet mode is set to OFF, lengthy strings intended to be read by humans are sent in response to commands. This mode is useful when using a terminal program for testing or debugging. The syntax of the command is:

SQ <ON/OFF> <CR>

Where <ON/OFF> is 0 = OFF, 1 = ON.

If the "SQ 0" command is accepted, the device will respond with the string: *Setting Quiet Mode to false.<CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*

The quiet mode state is saved in non-volatile memory. It does not need to be continually set. It will not hurt the device to be repeatedly set with the same value as it is only written if a different value is set. Note: New devices default to quiet mode ON.

(EH) Set Echo Mode

The Set Echo Mode command controls the text output of the control port during commands. When echo mode is turned on, all characters that are received on the Ethernet port are sent or “echoed” back. This mode is useful when using a terminal program for testing or debugging. When echo mode is turned off, the characters received are not echoed back. All command descriptions above assume that echo mode is turned off. Echo mode OFF should generally be used for normal operation. The syntax of the command is:

EH <ON/OFF> <CR>

Where <ON/OFF> is 0 = OFF, 1 = ON.

If the command is accepted, the device will respond with the string: *ACK <CR>*
If the command is interpreted but fails for any reason the device will respond with the string: *NAK <CR>*

The echo mode state is saved in non-volatile memory. It does not need to be continually set. It will not hurt the device to be repeatedly set with the same value as it is only written if a different value is set.

Jupiter App Controller Numbers

In the initial release of Jupiter software, the best method for retreiving Controller Numbers for use with third-party control systems is with the Custom Preset Parameter Browser. The Controller Numbers addressable within each Jupiter App are displayed in the far right column of the lower section.