MICRONIX USA

MMD-100 Appendix

1. Serial Port Setup

1.1 Quick Start Serial Port

The following Quick Start Guide is intended to provide a basic set-up of the MMD-100 to adjust configurations. The following paragraphs will provide a walkthrough of the steps needed to set-up the controller.

1. Install Drivers
    a. To ensure correct communication between the module and PC, install the proper drivers onto the communicating computer prior to connecting the MMD-100.
    b. The drivers may be found on the supplemental installation CD or can be downloaded from: http://www.ftdichip.com/Drivers/VCP.htm

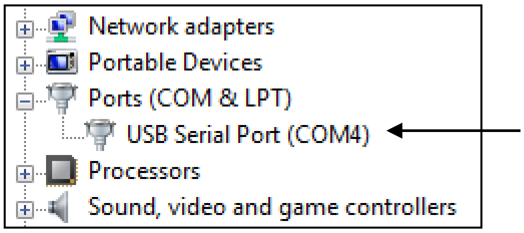
2. Connect Motion Devices
    a. A single MMD-100 controller is capable of driving one piezo motor in either open or closed loop.
    b. Connect the male D-sub 9-pin piezo motor cable to the Motor/Axis Input (as shown in the *Product Description*).
    c. If applicable, connect the female D-sub 9-pin closed loop feedback cable to the Encoder Input.

3. Connect Module/Stack to PC
    a. Use the supplied Mini USB to USB cable to connect the MMD-100 controller to the communicating PC. Only one USB cable is required per module/stack.

4. Power Up Controller
    a. Connect the controller to a 5V, regulated power supply with the correct amperage rating.
    b. Each MMD-100 requires 1A. If powering a stack; add up the amperage requirements of the individual controllers to determine the necessary power supply for the stack.

5. Check COM Port
    a. It is necessary to note the COM Port assigned to the MMD-100 when connecting to a PC.
        i. In Windows Vista Open the Device Manager:
            1 Windows Logo (in the bottom left corner by default)
            2 Control Panel
            3 Device Manager
        ii. In Window XP Open Device Manager:
            1 Start (in the bottom left corner by default)
            2 Control Panel
            3 System
            4 select the Hardware tab
            5 Click the device manager button
        iii. In Windows 7 Open the Device Manager:

b. After powering up the controller (Step 4), note the USB Serial Port assigned. See the figure below showing a snapshot of the Device Manager window:



Connected MMD-100 is assigned to COM4

6. Continue section 3.2 for setting up the serial port connection to adjust configuration.

1.2 Serial Port Setup

If the MMD-100 is not automatically recognized by your computer, you will have to first install the FTDI interface drivers before communicating with the controller. The drivers are supplied on the supplemental CD under the folder *MMD-100 Drivers* or can be downloaded from:

http://www.ftdichip.com/Drivers/VCP.htm

Below are the virtual serial port configuration settings necessary for correct communication setup:

Software Parameter	Setting
Data Bits	8
Stop Bits	1
Parity	No
Handshake	No
Baud rate	38400

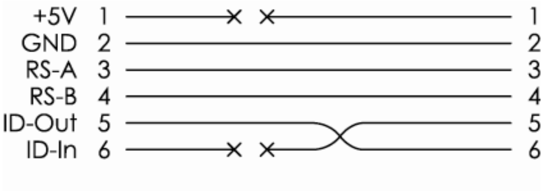
1.3 RJ11 RS485 Bus

The RS485 Intermodular RJ11 connector connects directly to the same Serial bus as the FTDI interface above. The RS485 line requires a terminating resistor of 22kΩ or higher.

1.4 RS-485 Intermodular Connector Cable Pin-out

The RS-485 Intermodular Connector Cable is used to daisy chain two MMD-100 modules together, allowing for alternative module configurations. The cable is directional and its orientation should be noted when configuring axis numbers, for the direction of the cable will determine axis order.

RS-485 Intermodular Connector Cable Pin-out:



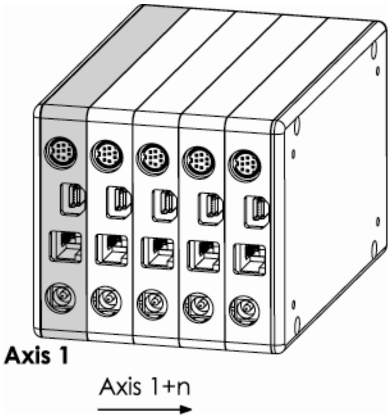
1.5 Axis Addressing

Auto Addressing is the default method of assigning axis numbers on start up. Controllers are automatically assigned axis numbers on every power up, starting with axis 1 and increasing consecutively until reaching axis 99.

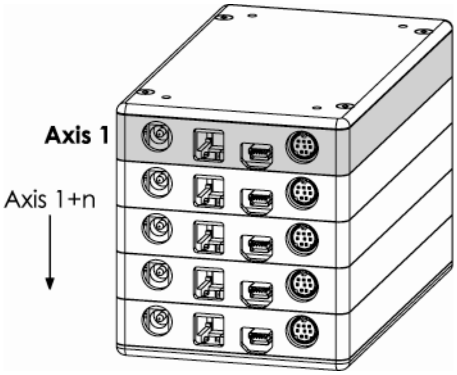
Manual axis numbers may be assigned to a unique controller using the ANR Command. This overrides Auto Addressing, as the controller stores the axis number until reassigned or reset back to Auto Addressing. In the case of having a mix of manually assigned and auto addressed controllers, the Auto Addressed axis numbers increase consecutively after each manually assigned axis in the stack. For example; in a stack of 5 controllers with the third controller manually assigned to axis 10, the axis numbers will read: `1, 2, 10, 11, 12`

If two controllers are accidentally assigned the same axis number, use a global command to reset all controllers back to Auto Addressing.

The figures shown below illustrate axis numbers for a 5 module stack with Auto Addressing assigned. Axis 1 is noted and shown in grey.



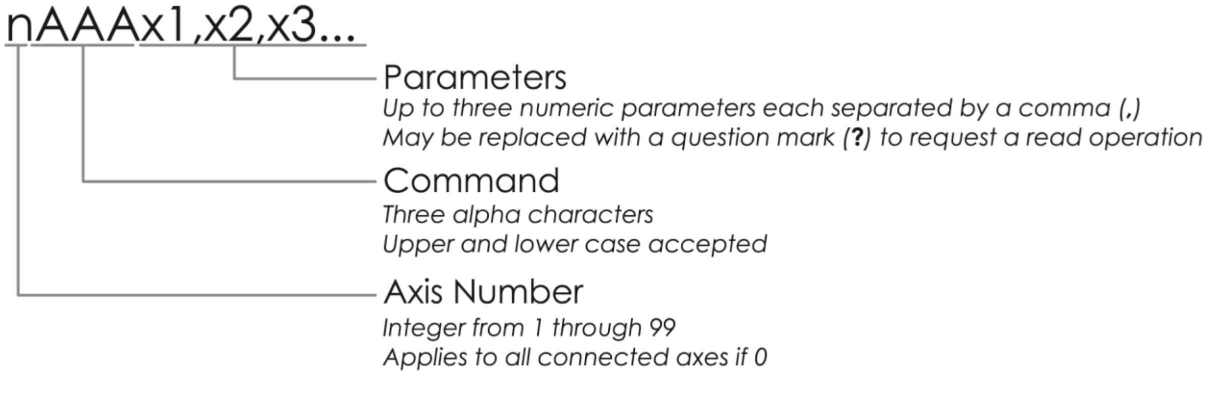
Horizontal stack (rear view)
With power inputs along bottom, Axis 1 is on the far left.



Vertical stack (rear view)
With power inputs along left hand side, Axis 1 is on the very top.

2. Commands

2.1 Command Line Syntax



nAAAx1,x2,x3...

Parameters

*Up to three numeric parameters each separated by a comma (**,**)*
*May be replaced with a question mark (**?**) to request a read operation*

Command

Three alpha characters
Upper and lower case accepted

Axis Number

Integer from 1 through 99
Applies to all connected axes if 0

There are three components to every command prompt. The first is the "<u>Axis Number</u>" which designates which controller, or axis, will receive the command. If the "Axis Number" is 0, then the command will be sent globally to all connected controllers. It is possible to connect up to 99 controllers; therefore the "Axis Number" will be an integer value from 0 through 99.

The second component is the "<u>Command</u>", which is always comprised of three letters. Each command is outlined, along with its corresponding parameters, in the *Command Description* section 5.9 of this manual.

The third and final component is the "<u>Parameter</u>". This portion is command dependent, meaning that the parameter value will change depending on the specific requirements of the "Command". Where applicable, a question mark (?) may be substituted to initiate a read operation which will return information regarding the particular command. There may be up to three separate parameters for a particular command, each parameter value is separated by a comma (,).

All white space (blank spaces) are ignored in the command format. The following are examples of equivalent commands:

```
4TRM13,45
4 TRM 13 , 45
```

2.2 Command Line Format

Commands are first executed in the order that they are input, then line by line. This means that two commands on the same line are executed significantly closer to each other than if they were on two separate lines. Each command is separated by a semicolon (;) and every command line ends in a terminator (EX: carriage return). The following is an example of a command line entry:

```
1ZRO;3ENC.01        | Axis 1, Set Position to 0.000000; Axis 3, Set Encoder resolution to 10nm
```

Using multiple commands on the same command line allows for synchronization of different commands to different axes. Up to 8 commands are allowed per command line.

Only one read operation is allowed per line. The controller will not send information unless requested to do so by a read operation.

2.3 Global Commands

Some commands have the option of being called globally. This means that you can send the same command to all available axes. To do this, replace the axis number of a global command with a '0'. For example; 0ACC 50 will set the acceleration of all available axes to 50 mm/s^2 [degrees/s^2].

2.4 Multiple Parameters

When dealing with a command that has multiple parameters, it is possible to change a single parameter by omitting numbers for the parameters that will remain unchanged. For example; `4PID,,3` will only change the third parameter to a new value, "3".

2.5 Terminating Characters

When communicating with the controller, it is necessary to note the terminating characters involved in transmitting and receiving data. To send data to the controller, enter the desired commands in the command line followed by the new line and carriage return terminating characters [/n/r], or just the carriage return terminating character [/r]. When receiving, each line of data will be followed by the new line terminating character [/n] and the final line will end in the new line and carriage return terminating characters [/n/r]. The ASCII value for new line [/n] is 0X0A and for carriage return [/r] is 0X0D. The following is an example of data transmission:

`1ENC0.005 /n/r` | Axis 1, Set Enc Resolution to 5 nm/s [degrees/s^2] [New line, Carriage Return]

2.6 *Summary of Commands*

Command	Description	During Motion		Real-time		Program		Global		Page
		Set	Read	Set	Read	Set	Read	Set	Read	
ANR	Set Axis Number		✓	✓	✓			✓*		8
CER	Clear Errors	✓		✓				✓		9
DBD	Closed Loop Deadband		✓	✓	✓			✓		10
DEF	Restore Factory Defaults			✓						11
EAD	Set Analog or Digital Encoder		✓	✓	✓			✓		12
ENC	Select Encoder Resolution		✓	✓	✓			✓		13
EPL	Encoder Polarity		✓	✓	✓			✓		14
ERR	Read and Clear Errors		✓		✓					15
FBK	Set Open or Closed Loop Mode		✓	✓	✓					16
FMR	Upload Firmware			✓						17
HCG	Home Configuration		✓	✓	✓			✓		18
HOM	Home		✓	✓	✓	✓		✓		19
HST	Hard Stop Detection		✓	✓	✓			✓		20
IOD	IO Definition		✓	✓	✓					21
IOF	IO Function		✓	✓	✓					22
LCG	Limit Configuration		✓	✓	✓			✓		23
LPL	Limit Switch Polarity		✓	✓	✓					24
MLN	Move to Negative Limit			✓		✓		✓		25
MLP	Move to Positive Limit			✓		✓		✓		26
MOT	Toggle Motor On/Off		✓	✓	✓			✓		27
MPL	Motor Polarity		✓	✓	✓			✓		28
PDX	Distance per Pulse		✓	✓	✓					29
PID	Set Feedback Constants		✓	✓	✓					30
POS	Position		✓		✓					31
REZ	Set Resolution		✓	✓	✓					32
RST	Perform Soft Reset			✓				✓		33
SAV	Save Axis Settings			✓				✓		34
STA	Status Byte		✓		✓					35
VER	Firmware Version		✓		✓					36
ZRO	Zero Position			✓		✓				37
ZZZ	Take Axis Offline			✓				✓		38

* see ANR command page 8 for more info

2.7 Command Descriptions

ANR

Set Axis Number

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓			✓*	

Command Description:	This command is used to override Auto Addressing by manually assigning an axis number to a controller. Auto Addressing is the default method of assigning axis numbers on power up and may be reassigned to an axis by substituting a "0" for the parameter value. Simultaneous axis swapping is possible by using multiple ANR commands on the same command line. ∗This command can be called globally by specifying a '0' for the axis number; however it will only work if the new axis number parameter is set to '0' for auto-addressing.
Returns:	A read operation returns the following axis number values for the specified axis: 0 – Auto Addressing assigned (default) 1-99 – Manually assigned, axis number displayed
Syntax:	nANRx – Standard syntax nANR? – Read axis number value Error [#]: ANR? – Read operation with missing axis number [27] nANR – Missing new axis number parameter [28] ANRx – Missing axis number [30]
Parameter Description:	n[int] – Axis number x[int] – New axis number, 0 for Auto Addressing ? – Read axis number value
Parameter Range:	n – 0 to 99 x – 0 to 99
Related Commands:	None
Example:	`5ANR1;1ANR5` | Simultaneous axis swapping: Axis 5, Set to Axis 1, Set to axis 5 axis 1; – `4ANR0` | Axis 4 , Set to Auto Addressing. However it will remain axis 4 until the MMD-100 is reset

CER

Clear Errors

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
✓		✓				✓	

Command Description:	This command is used to clear all error messages without reading them.
Returns:	A read operation cannot be used with this command.
Syntax:	nCER – Standard syntax 0CER – All axes clear error messages
Parameter Description:	n[int] – Axis number
Parameter Range:	n – 0 to 99
Related Commands:	ERR
Example:	1CER | Axis 1, clear error messages – 0CER | All axes, clear error messages

DBD Closed Loop Deadband

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓			✓	

Command Description:	This command is used to set the acceptable deadband and deadband timeout values. Deadband refers to the number of encoder counts (±) from the target that is considered acceptable. If the parameter (x1) is set to "0", the controller will continuously oscillate around the target. Deadband timeout refers to the amount of time that the controller will try to move into the deadband area. If the parameter (x2) is set to "0", the controller will seek continuously.
Returns:	A read operation returns the deadband and deadband timeout values for the specified axis.
Syntax:	nDBDx1,x2 – Standard syntax nDBD? – Read deadband and deadband timeout values 0DBDx1,x2 – All axes set deadband and deadband timeout values Error [#]: DBD? – Read operation with missing axis number [27] nDBD – Missing deadband and deadband timeout parameter values [28]
Parameter Description:	n[int] – Axis number x1[int] – Deadband x2[float] – Deadband timeout ? – Read deadband and deadband timeout values
Parameter Range:	n – 0 to 99 x1 – Encoder dependent, 0 for continuous, Encoder Counts x2 – Encoder dependent, 0 for infinite, Seconds (default 0)
Related Commands:	ENC, EPL
Example:	`1DBD10,1` | Axis 1, Set deadband to 10 encoder counts & deadband timeout to 1 second - `4DBD5,0` | Axis 4, Set deadband to 5 encoder counts & deadband timeout to infinite

DEF

Restore Factory Defaults

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
		✓					

Command Description:	This command restores the factory default parameters.
Returns:	A read operation is not available with this command.
Syntax:	nDEF – Standard syntax Error [#]: DEF – Missing axis number [30]
Parameter Description:	n[int] – Axis number
Parameter Range:	n – 1 to 99
Related Commands:	SAV
Example:	`1DEF` | Axis 2, Set deceleration value to 1.25 mm/s^2 [degrees/s^2]

EAD

Set Analog or Digital Encoder

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓			✓	

Command Description:	This command is used to specify whether the encoder signal for a specified axis is analog or digital. A power cycle is necessary to enact a change in this parameter.
Returns:	A read operation returns the following encoder mode values for the specified axis: 0 – Digital 1 – Analog
Syntax:	nEADx – Standard syntax nEAD? – Read encoder mode value 0EADx – All axes set encoder value Error [#]: xEAD – Missing encoder mode parameter [28] EAD? – Read operation with missing axis number [27]
Parameter Description:	n[int] – Axis number x[int] – Encoder mode ? – Read encoder mode value
Parameter Range:	n – 0 to 99 x – 0 for digital, 1 for analog
Related Commands:	ENC
Example:	`9EAD0` | Axis 9, Set encoder parameter to digital

ENC

Set Encoder Resolution

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓			✓	

Command Description:	This command is used to set the desired encoder resolution for the specified axis. When a digital encoder is connected, encoder resolution is determined by the encoder itself. Analog encoder resolution can be set by the controller.
Returns:	A read operation returns the encoder resolution value for the specified axis.
Syntax:	nENCx – Standard syntax nENC? – Read encoder resolution value 0ENCx – All axes execute encoder resolution value Error [#]: ENC? – Read operation with missing axis number [27] nENC – Missing encoder resolution parameter [28]
Parameter Description:	n[int] – Axis number x[float] – Encoder resolution ? – Read encoder resolution value
Parameter Range:	n – 0 to 99 x – 0.001 to 999.999 µm/count (milli-degrees/count)
Related Commands:	EAD
Example:	`2ENC10` | Axis 2, Set encoder resolution to 10 microns/count (10 milli-degrees/count)

EPL

Encoder Polarity

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓			✓	

Command Description:	This command is used to switch the encoder signal polarity for the specified axis. If the controller doesn't seem to be recording encoder position correctly, the polarity of the encoder signals could be reversed. Use this command to switch from the default setting (normal operation, n=0).
Returns:	A read operation returns the following encoder polarity values for the specified axis: 0 – Normal operation 1 – Reverse operation
Syntax:	nEPLx – Standard syntax nEPL? – Read encoder polarity value 0EPLx – All axes execute encoder polarity value Error [#]: EPL? – Read operation with missing axis number [27] nEPL – Missing encoder polarity parameter [28]
Parameter Description:	n[int] – Axis number x[float] – Encoder polarity ? – Read encoder polarity value
Parameter Range:	n – 0 to 99 x – 0 for normal operation, 1 for reverse operation
Related Commands:	DBD
Example:	`13EPL0` | Axis 13, Set encoder polarity to normal operation - `6EPL1` | Axis 6, Set encoder polarity to reverse operation

ERR

Read and Clear Errors

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓		✓				

Command Description:	This command is used to read and clear any pending error messages.
Returns:	A read operation returns a list of error messages for the specified axis in the following format. "AAA" signifies the specific command name that the error corresponds to. Error Number – Description [AAA]
Syntax:	nERR? – Standard syntax Error [#]: ERR? – Read operation with missing axis number [123]
Parameter Description:	n[int] – Axis number ? – Read error messages
Parameter Range:	n – 1 to 99
Related Commands:	None
Example:	`3ERR?` | Axis 3, Read error messages

FBK

Set Open or Closed Loop Mode

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓				

Command Description:	This command is used to select the feedback mode of the controller. See section 4.2 for more details
Returns:	A read operation returns the following loop mode values for the specified axis: 0 – Open Loop [default] 1 – Clean Open Loop 2 – Clean Open Loop Movement, Closed Loop deceleration 3 – Closed Loop
Syntax:	nFBKx – Standard syntax nFBK? – Read encoder mode value Error [#]: FBKx – Missing axis number [30] FBK? – Read operation with missing axis number [27] nFBK – Missing closed/open loop parameter [28]
Parameter Description:	n[int] – Axis number x[float] – Open/closed loop mode ? – Read encoder mode value
Parameter Range:	n – 1 to 99 x – 0 for open loop mode, 1 for clean sounding open loop mode, 2 for open loop with closed loop deceleration, 3 closed loop
Related Commands:	ENC, EAD, EPL, DBD
Example:	`2FBK3` | Axis 2, Set closed loop mode

FMR

Upload Firmware

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
		✓					

Command Description:	This command is used by the bootloader to upload new firmware to the specified axis.
Returns:	A read operation cannot be used with this command.
Syntax:	nFMR – Standard syntax Error [#]: FMR – Missing axis number [30]
Parameter Description:	n[int] – Axis number
Parameter Range:	n – 1 to 99
Related Commands:	VER
Example:	`1FMR` | Axis 1, upload new firmware

HCG

Home Configuration

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
		✓		✓		✓	

Command Description:	This command is used to select the direction of motion when the Home [HOM] command is initialized.
Returns:	A read operation returns the current direction setting: 0 – Home starts in the direction of the negative limit 1 – Home starts in the direction of the positive limit
Syntax:	nHCGx – Standard syntax 0HCGx – All axes set direction nHCG? – Read direction setting Error [#]: HCG? – Read operation with missing axis number [27] nHCG – Missing direction setting [28]
Parameter Description:	n[int] – Axis number x[int] – Set direction of motion.
Parameter Range:	n – 0 to 99 x – 0 for setting motion in the direction of the negative limit 1 for setting motion in the direction of the positive limit 2 for setting home command to home to negative hard limit 3 for setting home command to home to positive hard limit.
Related Commands:	HOM
Example:	`3HCG0` | Axis 3, Set initial direction of Home command towards the negative limit – `0HCG1` | All Axes, Set initial direction of Home command towards the positive limit

HOM

Home

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓	✓		✓	

Command Description:	This command is used to find the home (zero) position for a specified axis. An error will occur if there is no encoder signal at the time of execution. Home is configured using the HCG command. This command will jog the stage till it reaches the limit configured by the HCG command. It will then acquire the index position. HOM requires an encoder on the attached stage.
Returns:	A read parameter returns the following calibration values for the specified axis: 0 – Not calibrated to home position 1 – Calibrated to home position
Syntax:	nHOM – Standard syntax nHOM? – Returns 1 if homed since last startup otherwise returns 0 0HOM – All axes execute home position Error [#]: HOM? – Read operation with missing axis number [27]
Parameter Description:	n[int] – Axis number
Parameter Range:	n – 0 to 99
Related Commands:	HCG
Example:	`1HOM` | Axis 1, Move to home position

HST

Hard Stop Detection

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓			✓	

Command Description:	This command is used to enable or disable hard stop detection. Hard stop detection will automatically stop motion when a hard stop is detected. After a hard stop is detected the stage will not be able to travel in the direction the hard stop was detected until it is moved in the opposite direction first. HST requires an encoder on the attached stage.
Returns:	A read operation returns the following Hard Stop Detection values for the specified axis: 0 – Disable 1 – Enable
Syntax:	nHSTx – Standard syntax nHST? – Read hard stop detection value 0HSTx – All axes set hard stop detection value Error [#]: xHST – Missing encoder mode parameter [28] HST? – Read operation with missing axis number [27]
Parameter Description:	n[int] – Axis number x[int] – Enable/Disable ? – Read Hard Stop Detection value
Parameter Range:	n – 0 to 99 x – 0 - disabled, 1 - enabled
Related Commands:	
Example:	`9HST1` | Axis 9, Set HST enabled

IOD

Set IO Definition

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
		✓	✓				

Command Description:	This command is used to select Input or Output for one of the IO pins on the 8-Pin Din connector.
Returns:	A read operation is not available with this command.
Syntax:	nIODx1,x2 – Standard syntax nIOD? – Read encoder mode value Error [#]: IODx1,x2 – Missing axis number [30] IOD? – Read operation with missing axis number [27] nIOD – Missing closed/open loop parameter [28]
Parameter Description:	n[int] – Axis number x1[int] – IO Pin x2[int] – Input/ Output ? – Read encoder mode value
Parameter Range:	n – 1 to 99 x1 – 1 – IO1 (output only) 2 – IO2 – set to 1 3 – IO3 – set to 1 4 – IO4 x2 – 0 – Output 1 - Input
Related Commands:	IOF
Example:	`2IOD4,1` | Axis 4, Set IO2 to an Input

IOF

Set IO Function

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
		✓	✓				

Command Description:	This command is used to select the function of an IO pin.
Returns:	A read operation is not available with this command.
Syntax:	nIOFx1,x2 – Standard syntax nIOF? – Read encoder mode value Error [#]: IOFx1,x2 – Missing axis number [30] IOF? – Read operation with missing axis number [27] nIOF – Missing closed/open loop parameter [28]
Parameter Description:	n [int] – Axis number x1 [int] – IO Pin x2 [int] – IO Function ? – Read encoder mode value
Parameter Range:	n – 1 to 99 x1 – 1 – IO1 – output only 2 – IO2 – set to 9 3 – IO3 – set to 10 4 – IO4 x2 – 0 – No function 2 – Output pulse trigger when in position 3 – Output level when in position will appear as a falling edge when in position is triggered, and a rising edge once it has completed 4 – Input Home on rising edge 5 – Input Motor on (rising edge) Motor off (falling edge) 6 – Home Status Complete Output, will appear as a falling edge when the Home is triggered, and a rising edge once it has completed
Related Commands:	IOD
Example:	`2IOF2,1` | Axis 2, Set IO2 to data logging trigger

LCG

Limit Configuration

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓			✓	

Command Description:	This command selects whether the limit switch inputs on the motor connector are ignored, otherwise it will stop motion.
Returns:	A read operation is not available with this command.
Syntax:	nLCGx – Standard syntax Error(s): LCGx – Missing axis number [30] nLCG – Missing program number parameter [28]
Parameter Description:	n[int] – Axis number x[int] – 0 – ignore [default] 1– active
Parameter Range:	n – 1 to 99 x – 0 – ignore [default] 1– active
Related Commands:	LPL
Example:	`1LCG1` | Axis 1, set limit switches active

LPL

Limit Switch Polarity

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓				

Command Description:	This command sets whether the limit switch inputs are active high[1] or low[0]
Returns:	A read operation returns the program table for the specified axis.
Syntax:	nLPLx – Standard syntax Error(s): LPLx – Missing axis number [30] nLPL – Missing program number parameter [28]
Parameter Description:	n[int] – Axis number x - 0 –Active Low - 1 – Active High
Parameter Range:	n – 1 to 99 x – 0 – active low [default] 1– active high
Related Commands:	LCG
Example:	`6LPL1` | Axis 5, limit switches set to active high

MLN

Move to Negative Limit

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
		✓		✓		✓	

Command Description:	This command initiates a move to the negative limit position. Upon reaching the negative hard limit the controller will then move the stage back from the hard limit and stop. An error will occur if there is no encoder signal at the time of execution.
Returns:	A read operation is not available with this command.
Syntax:	nMLN – Standard syntax 0MLN – All axes execute move to negative limit position Error [#]: MLN – Missing axis number [30]
Parameter Description:	n[int] – Axis number
Parameter Range:	n – 0 to 99
Related Commands:	MLP
Example:	`8MLN` | Axis 8, Move to negative limit position – `0MLN` | All Axes, Move to negative limit position

MLP

Move to Positive Limit

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
		✓		✓		✓	

Command Description:	This command initiates a move to the positive limit position. Upon reaching the positive hard limit the controller will then move the stage back from the hard limit and stop. An error will occur if there is no encoder signal at the time of execution.
Returns:	A read operation is not available with this command.
Syntax:	nMLP – Standard syntax 0MLP – All axes execute move to positive limit position Error [#]: MLP – Missing axis number [30]
Parameter Description:	n[int] – Axis number
Parameter Range:	n – 0 to 99
Related Commands:	MLN
Example:	`1MLP` | Axis 1, Move to positive limit position – `0MLP` | All Axes, Move to positive limit position

MOT

Toggle Motor Off/On

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓			✓	

Command Description:	This command is used to turn the motor current flow "Off" or "On" for a specified axis. Primarily used for stages utilizing stepper motors where the motor would be unable to rotate freely while powered.
Returns:	A read operation returns the following motor current off/on values for the specified axis: 0 – Motor current is off 1 – Motor current is on
Syntax:	nMOTx – Standard syntax nMOT? – Read motor current off/on value 0MOTx – All axes set motor value Error [#]: MOT? – Read operation with missing axis number [27] xMOT – Missing motor off/on parameter [28]
Parameter Description:	n[int] – Axis number x[float] – Motor current off/on ? – Read motor current off/on value
Parameter Range:	n – 0 to 99 x – 0 for motor current off 1 for motor current on
Related Commands:	None
Example:	`1MOT0` | Axis1, Set motor current to off

MPL

Toggle Motor Polarity

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓			✓	

Command Description:	This command set the motor polarity for the specified axis. If the theoretical positive direction is away from the motor, changing this setting will make the theoretical positive direction towards to motor.
Returns:	A read operation returns the current motor polarity setting for the specified axis.
Syntax:	nMPLx – Standard syntax nMPL? – Read motor current off/on value 0MPLx – All axes set motor value Error [#]: MPL? – Read operation with missing axis number [27] nMPL – Missing motor off/on parameter [28]
Parameter Description:	n[int] – Axis number x[float] – Motor Polarity setting ? – Read motor current off/on value
Parameter Range:	n – 0 to 99 x – 0 Normal 1 Reverse
Related Commands:	MVR
Example:	`1MPL0` | Axis1, To normal Polarity

PDX

Distance per Pulse

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓			✓	

Command Description: This command is used to set the desired distance per input pulse for the specified axis.

Returns: A read operation returns the distance per pulse value for the specified axis.

Syntax:

nPDXx – Standard syntax
nPDX? – Read distance per pulse value
0PDXx – All axes execute distance per pulse value

Error [#]:
PDX? – Read operation with missing axis number [27]
nPDX – Missing distance per pulse parameter [28]

Parameter Description:

n[int] – Axis number
x[float] – Distance per Pulse in nm
? – Read distance per pulse value

Parameter Range:

n – 0 to 99
x – 0.5 to 150 nm per pulse (increments of 0.5nm)

Related Commands:

Example: `2PDX1.5` | Axis 2, Set the distance traveled per pulse to 1.5nm

PID

Set Feedback Constants

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓				

Command Description:	This command is used to set the encoder feedback constants for a specified controller.
Returns:	A read operation returns the encoder feedback constant values for the specified axis.
Syntax:	nPIDx1,x2,x3 – Standard syntax nPID? – Read encoder feedback constant values Error(s): PIDx1,x2,x3 – Missing axis number [30] PID? – Read operation with missing axis number [27] nPID – Missing encoder feedback constant parameters [28]
Parameter Description:	n[int] – Axis number x1[float] – K_p (proportional constant, piezo only) x2[float] – K_i (integral constant, stepper only) x3 [float] – K_d (derivative constant, stepper only) ? – Read encoder feedback constants and values
Parameter Range:	n – 1 to 99 x1 – 0.000 to 1.000 x2 – 0.000 to 1.000 x3 – 0.000 to 1.000
Related Commands:	FBK, ENC, POS
Example:	`5PID.02,.04,.05` | Axis 5, Set encoder feedback constants to 0.02, 0.04 and 0.05, respectively – `2PID.03,,` | Axis 2, Set encoder feedback constant K_p to 0.03, other constants remain unchanged – `4PID,,.07` | Axis 4, Set encoder feedback constant K_d to 0.07, other constants remain unchanged

POS

Position

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓		✓				

Command Description:	This command is used to read the position information from the specified axis controller
Returns:	A read operation returns the position values in mm for the specified axis in the following format: [Theoretical position in mm; Encoder position in mm] [Theoretical position in degrees; Encoder position in degrees]
Syntax:	nPOS? – Standard syntax Error(s): POS? – Read operation with missing axis number [27]
Parameter Description:	n[int] – Axis number ? – Read position values
Parameter Range:	n – 1 to 99
Related Commands:	MVR
Example:	`4POS?` | Axis 4, Read position values

REZ

Set Resolution

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓	✓	✓				

Command Description:	This command is used to set the DAC (digital to analog converter) steps per micron resolution for the specified axis. This value should be calibrated in order to receive correct closed loop position information.
Returns:	A read operation returns the resolution value in steps per micron for the specified axis.
Syntax:	nREZx – Standard syntax nREZ? – Read steps per micron resolution value Error(s): REZ? – Read operation with missing axis number [27] REZx – Missing axis number [30] nREZ – Missing steps per micron resolution parameter [28]
Parameter Description:	n[int] – Axis number x[float] – Steps per micron resolution (steps/milidegrees) (default is 20,000) ? – Read steps per micron resolution value (steps/millidegrees)
Parameter Range:	n – 1 to 99 x – 0 to 99 DAC steps per micron (steps/milli-degrees)
Related Commands:	None
Example:	`9REZ25` | Axis 9, Set resolution to 25 steps/micron [steps/millidegrees] – `3REZ?` | Axis 3, Read steps/micron [steps/degrees] resolution value

RST

Perform Soft Reset

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
		✓				✓	

Command Description:	This command is used to perform a soft reset of the specified axis.
Returns:	A read operation cannot be used with this command.
Syntax:	nRST – Standard syntax 0RST – All axes execute soft reset
Parameter Description:	n[int] – Axis number
Parameter Range:	n – 1 to 99
Related Commands:	None
Example:	`8RST` | Axis 8, execute soft reset

SAV

Save Axis Settings

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
		✓				✓	

Command Description:	This command is used to save all settings for the specified axis. This allows an axis to be configured on power up.
Returns:	A read operation cannot be used with this command.
Syntax:	nSAV – Standard syntax 0SAV – All axes save settings
Parameter Description:	n[int] – Axis number
Parameter Range:	n – 0 to 99
Related Commands:	None
Example:	`16SAV` | Axis 16, save settings

STA

Status Byte

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓		✓				

Command Description:

This command is used to check the status register for a specified axis.

Returns:

A read operation will return an integer from 0 to 255 describing the status of the axis. The byte must be decoded in binary to determine the value of each bit.

Bit	7	6	5	4	3	2	1	0
Name	ERR	ACC	CNST	DEC	STP	PGM	PLS	NLS

Note: Bits 2, 1 and 0 are unused

Bit 7: 1 – One or more errors have occurred. Use ERR? Or CER to clear.
0 – No Errors have occurred.

Bit 6: 1 – Currently in Acceleration phase of motion.
0 – Not in Acceleration phase of motion.

Bit 5: 1 – Currently in Constant Velocity phase of motion.
0 – Not in Constant Velocity phase of motion.

Bit 4: 1 – Currently in Deceleration phase of motion.
0 – Not in Deceleration phase of motion.

Bit 3: 1 – Stage has stopped. (If in Closed Loop Stage is on target)
0 – Stage is moving. (If in Closed Loop Stage is off target)

Bit 2: 1 – A Program is currently running
0 – No program is running

Bit 1: 1 – Positive Switch is Activated
0 – Positive Switch is not Activated

Bit 0: 1 – Negative Switch is Activated
0 – Negative Switch is not Activated

Syntax:

nSTA? – Standard syntax

Error(s):
STA? – Read operation with missing axis number [27]
nSTA – Missing read operation parameter [28]

Parameter Description:

n[int] – Axis number
? – Read status register

Parameter Range:

n – 1 to 99

Related Commands:

None

Example:

`6STA?` | Axis 6, Read status register

VER

Firmware Version

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
	✓		✓				

Command Description:	This command is used to check the firmware version for the specified axis.
Returns:	A read operation returns the firmware version for the specified axis.
Syntax:	nVER? – Standard syntax Error(s): VER? – Read operation with missing axis number [27] nVER – Missing read operation parameter [28]
Parameter Description:	n[int] – Axis number ? – Read firmware version
Parameter Range:	n – 1 to 99
Related Commands:	None
Example:	`11VER?` | Axis 11, Read firmware version

ZRO

Zero Position

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
		✓		✓			

Command Description:	This command is used to set the absolute zero position for the specified axis.
Returns:	A read operation cannot be used with this command.
Syntax:	nZRO – Standard syntax Error [#]: ZRO – Missing axis number [123]
Parameter Description:	n[int] – Axis number
Parameter Range:	n – 1 to 99
Related Commands:	None
Example:	1ZRO | Axis 1, set current position as absolute zero

ZZZ

Take Axis Offline

During Motion		Real-time		Program		Global	
Set	Read	Set	Read	Set	Read	Set	Read
		✓				✓	

Command Description:	This command is used to take the specified axes offline. An offline axis will not respond until the power is cycled.
Returns:	A read operation cannot be used with this command.
Syntax:	nZZZ – Standard syntax ZZZ – Missing axis number, all axes set to offline
Parameter Description:	n[int] – Axis number
Parameter Range:	n – 1 to 99
Related Commands:	None
Example:	

2.8 Error Messages

Error Number	Name	Description
10	Receive Buffer Overrun	The Receive Buffer has reached or exceeded maximum capacity.
11	Motor Disabled	The command that triggered this error was trying to move the servo while it was disabled.
12	No Encoder Detected	The command that triggered this error was trying to access encoder data when no encoder was attached.
13	Index Not Found	The controller moved across the full range of motion and did not find an index.
14	Home Requires Encoder	The HOM command requires an encoder signal.
20	Command is Read Only	The command that triggered this error only supports read operations. The command must be followed by a question mark to be accepted. Ex: XXX?
21	One Read Operation Per Line	Multiple read operations on the same command line. Only one read operation is allowed per line, even if addressed to separate axes.
22	Too Many Commands On Line	The maximum number of allowed commands per command line has been exceeded. No more than 8 commands are allowed on a single command line.
23	Line Character Limit Exceeded	The maximum number of characters per command line has been exceeded. Each line has an 80 character limit.
24	Missing Axis Number	The controller could not find an axis number or the beginning of an instruction. Check the beginning of the command for erroneous characters.
25	Malformed Command	The controller could not find a 3-letter instruction in the input. Check to ensure that each instruction in the line has exactly 3 letters referring to a command.
26	Invalid Command	The 3-letter instruction entered is not a valid command. Ensure that the 3-letter instruction is a recognizable command.

27	Global Read Operation Request	A read request for a command was entered without an axis number. A read request cannot be used in a global context.
28	Invalid Parameter Type	1. The parameter entered does not correspond to the type of number that the instruction requires. For example, the command may expect an integer value, therefore sending a floating point value will trigger this error. 2. The allowable precision for a parameter has been exceeded. For example, velocity can be specified with a precision of 0.001 mm/sec. If a more precise velocity value of 0.0001 mm/sec is entered, this error will be triggered. Refer to the command pages for the type of parameter that each command expects.
29	Invalid Character in Parameter	There is an alpha character in a parameter that should be a numeric character.
30	Command Cannot Be Used In Global Context	The command entered must be addressed to a specific axis number. Not all commands can be used in a global context. Check the specific command page or the table of commands for more info.
31	Parameter Out Of Bounds	The parameter is out of bounds. The current state of the controller will not allow this parameter to be used. Check the command page for more information.
36	Command Cannot Be Executed During Motion	Only certain commands can be executed when motion is in progress. Check the command pages for information on individual commands.
38	Read Not Available For This Command	This error is triggered by a read request from a command that does not support a read operation.
81	Analog Encoder Not Available In this Version	The current version of firmware installed does not support Analog Encoders.