

MMC-203

Series



3-Axis Stepper Motor Controller Quick Start Guide

Table of Contents

1. Introduction		1-3
1.1 Product Description	1-3	
1.2 Features	1-4	
1.3 Package Contents	1-4	
2. Quick Start Guide		2-4
2.1 Quick Start Guide Overview	2-4	
2.2 Quick Start MMC-100 Motion Controller Platform	2-6	
2.3 Using the MMC-100 Motion Controller Platform	2-7	
3. Technical Information		3-8
3.1 MMC-203 Specifications	3-8	
3.2 Serial Port Setup	3-8	
3.3 RJ11 RS485 Bus	3-9	
4. Operation		4-9
4.1 Axis Addressing	4-9	
4.2 Feedback Control	4-10	
4.3 HOM, MLN, and MLP	4-10	
5. Commands		5-11
5.1 Command Line Syntax	5-11	
5.2 Command Line Format	5-11	
5.3 Global Commands	5-12	
5.4 Multiple Parameters	5-12	
5.5 Synchronous Move	5-12	
5.6 Program Mode	5-12	
5.7 Terminating Characters	5-13	
5.8 Error Messages	5-14	
6. Appendix		6-18
6.1 Encoder Input Pin-out	6-18	
6.2 Motor Pin-out	6-18	
6.3 8-Pin Din IO connector	6-18	
6.4 RS-485 Intermodular Connector Cable Pin-out	6-19	
6.5 Power supply pin out	6-19	
6-19		

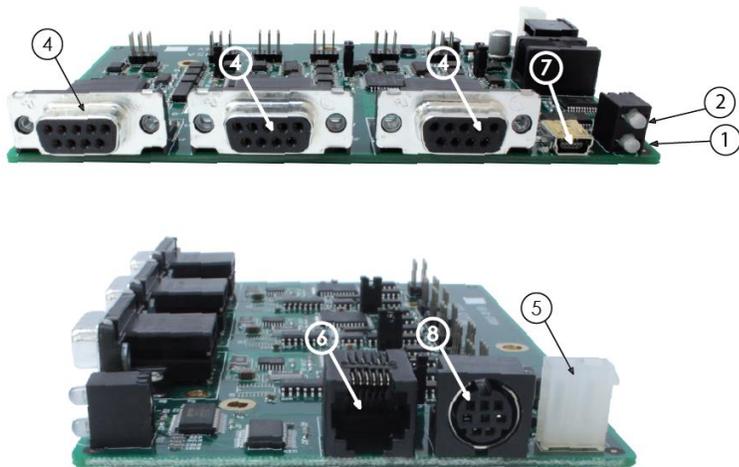
Command Index

Command	Description	Page	Command	Description	Page
ACC	Acceleration	16	MVR	Move Relative	56
AMX	Maximum Allowable Acceleration	17	PGL	Loop Program	57
ANR	Set Axis Number	18	PGM	Begin Program Recording	58
CER	Clear Errors	19	PGS	Run Program At Start-Up	59
CFG	Configuration Mode	20	PID	Set Feedback Constants	60
DAT	Dump Trace Data	21	POS	Position	61
DBD	Closed Loop Deadband	22	REZ	Set Resolution	62
DEC	Deceleration	23	RST	Perform Soft Reset	63
DEF	Restore Factory Defaults	24	RUN	Start Synchronous move	64
EAD	Set Analog or Digital Encoder	25	SAV	Save Axis Settings	65
ENC	Select Encoder Resolution	26	STA	Status Byte	66
END	End Program Recording	27	STP	Stop Motion	67
EPL	Encoder Polarity	28	SVP	Save Startup Position	68
ERA	Erase Program	29	SYN	Sync	69
ERR	Read and Clear Errors	30	TLN	Negative Soft Limit Position	70
EST	Emergency Stop	31	TLP	Positive Soft Limit Position	71
EXC	Execute Program	32	TRA	Perform Trace	72
FBK	Set Open or Closed Loop Mode	33	UMX	Max Micro Steps	73
FMR	Upload Firmware	34	UST	Micro steps	74
FSR	Full Steps Per Rev	35	VEL	Velocity	75
GRR	Gear Ratio	36	VER	Firmware Version	76
HCG	Home Configuration	37	VMX	Max. Allowable Velocity	77
HOM	Home	38	VRT	Encoder Velocity	78
JAC	Jog Acceleration and Deceleration	39	WST	Wait For Stop	79
JOG	Jog Mode	40	WSY	Wait For Sync	80
LCG	Limit Configuration	41	WTM	Wait For Time Period	81
LDR	Positive/Negative Limit Location	42	ZRO	Zero Position	82
LIM	Limit Status	43	ZZZ	Take Axis Offline	83
LSP	Lead Screw Pitch	44			
LST	Program List	45			
LPL	Limit Switch Polarity	46			
MCM	Max Motor Current	47			
MCS	Motor Current Setting	48			
MLN	Move to Negative Limit	49			
MLP	Move to Positive Limit	50			
MOT	Toggle Motor On/Off	51			
MPL	Motor Polarity	52			
MSA	Synchronous Move – Absolute	53			
MSR	Synchronous Move – Relative	54			
MVA	Move Absolute	55			

1. Introduction

1.1 Product Description

The MMC-203 is a high performance integrated stepper motor controller/driver designed to be used as a standalone 3 axis unit. The MMC-203 is capable of driving a 3 stepper motor with a resolution as fine as 2000 microsteps per fullstep in open loop. The closed loop resolution is dependent on the resolution of the encoder (typically 50 nm).



Open Loop MMC-203 shown

1. LED Error Indicator 1
 - a. Red – An error has occurred
2. LED Addressing Indicator 2
 - a. Orange – Stage is Unaddressed
 - b. Green – Stage has an address and is ready
3. Encoder Input, Male D-Sub 9 Pin Connector
4. Motor/Axis Output, Female D-Sub 9-Pin Connector
5. Power Supply, +24VDC, Regulated
6. RS485 - RJ11 Connector
7. USB Connector
8. I/O Connector

1.2 Features

- Integrated controller/driver for stepper motors
- Open loop/closed loop operation
- Open loop resolution of 4096 Micro steps per full step*
- Closed loop resolution dependent on the encoder (typically 50 nm)
- A quad B encoder feedback
- USB interface (one interface for up to 99 axes)
- Windows GUI and LabVIEW VI

1.3 Package Contents

If product is damaged or there are missing components, contact MICRONIX USA immediately. Do not discard product packaging in case of return shipment.

Package Contents:

- MMC-203 Controller
- User Manual
- Supplemental CD
- Power Cable

2. Quick Start Guide

2.1 Quick Start Guide Overview

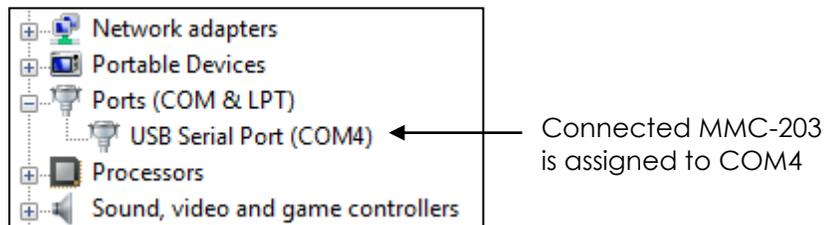
The following Quick Start Guide is intended to provide a basic set-up of the MMC-203 in the least amount of time. The following paragraphs will provide a walkthrough of the steps needed to set-up the controller and verify that the system is working correctly.

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 MMC-203.
 - 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 MMC-203 controller is capable of driving one stepper motor in either open or closed loop.
 - b. Connect the male D-sub 9-pin stepper 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.

*This value is theoretical actual value with vary depending on the attached stepper motor.

3. Connect Module/Stack to PC

- a. Use the supplied Mini USB to USB cable to connect the MMC-203 controller to the communicating PC. Only one USB cable is required per module/stack.
4. Power Up Controller
 - a. Connect the controller to a 24V, regulated power supply with the correct amperage rating.
 - b. Each MMC-203 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 MMC-203 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:



6. Continue to Quick Start MMC-100 Motion Controller Platform
 - a. The following section will help you get running with the MMC-100 Motion Controller Platform program.

2.2 Quick Start MMC-100 Motion Controller Platform

The following Quick Start Guide is intended to provide a basic set-up of the MMC-100 MCP program. The MMC-203 can be run with the MMC-100 MCP though some commands will not show up in the settings tab; you will be able to access the full functionality of the MMC-203 with the MMC-100 MCP. The following paragraphs will provide a walkthrough of the steps needed to install the program and verify that the system is working correctly.

1. Pre-Installation

- a. This guide assumes you have already run through the previous Quick Start guide and that the controller is on and connected to a Com port on your computer. Please verify that this is true.
- b. You will need the .NET Framework 4.0. If you are unsure if you have the .NET Framework 4.0 follow these steps.
 - i. Open the start menu (windows icon if using Vista).
 - ii. Open the Control Panel
 - iii. Open "Add or Remove Programs" ("Programs and Features" if using Vista)
 - iv. Scroll through the list and find "Microsoft .NET Framework" If it is 4.0 skip to step-2. Otherwise continue with step c.
- c. To install the .NET Framework 4.0 you will need a connection to the internet.
 - i. Navigate to this site:
<http://www.microsoft.com/downloads/details.aspx?FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992&displaylang=en>
 - ii. Download and run the web installer
 - iii. At the conclusion of this install you will be asked to restart your computer. Do this now.

2. Install

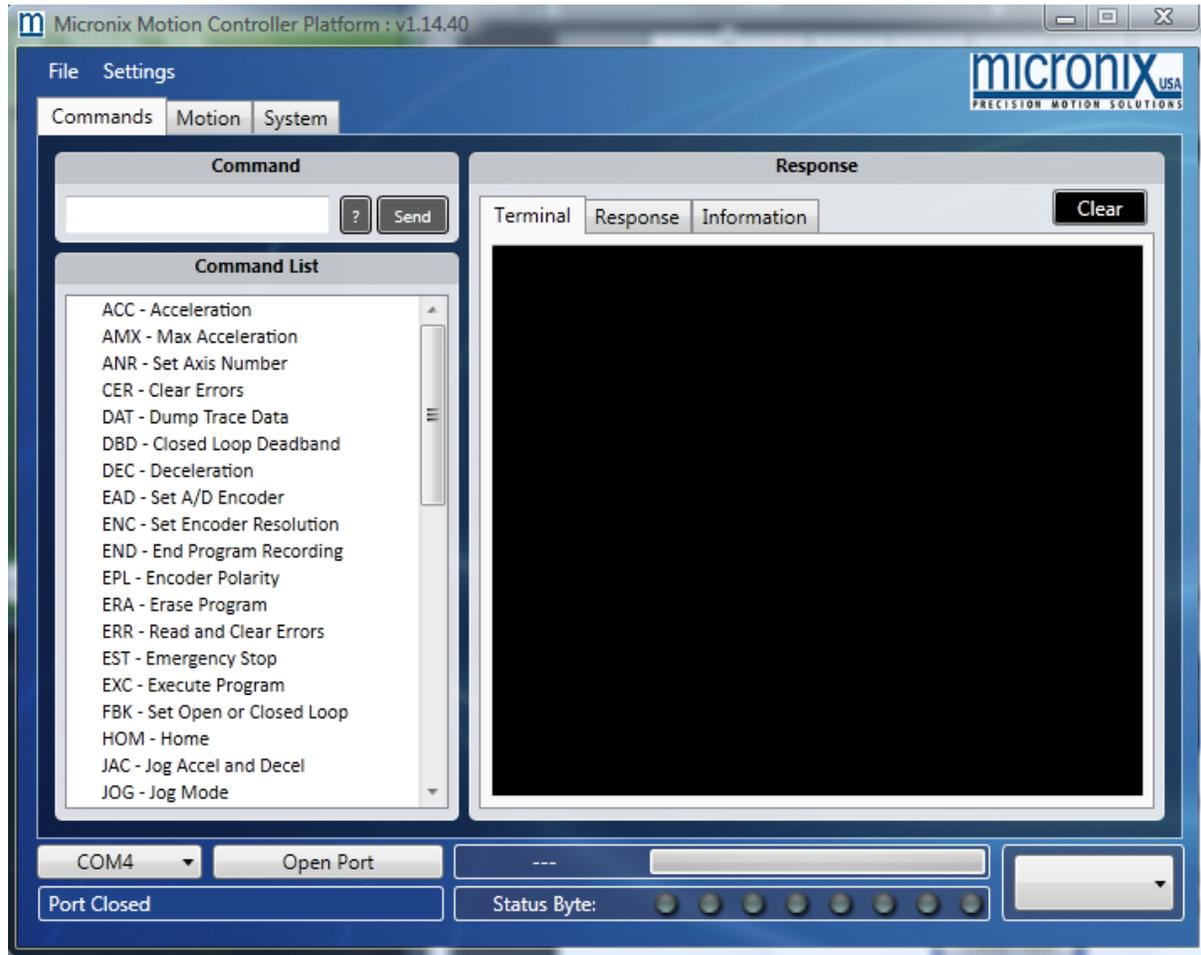
- a. To install the MMC-203 motion controller platform double click the setup.exe file on the supplied CD and follow the on screen instructions.

3. Run

- a. The installer placed a start menu short-cut to the MMC-100 MCP program. Make sure that your MMC-203 is connected to your computer, powered on, and connected to a valid COM port as discussed in section 2.1
- b. Open the start menu (or windows icon for vista)
- c. Open the 'all programs' tab
- d. Open the MICRONIX USA folder
- e. Run the MMC-100 MCP program

2.3 Using the MMC-100 Motion Controller Platform

In the Quick Start Guide Overview you connected your MMC-203 to your computer. In the Quick Start MMC-203 Motion Controller Platform you installed and ran the MMC-100 MCP software. This section will describe the capabilities of the MMC-100 MCP program and give you a brief understanding of how to use it.

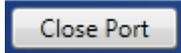


1. Port Control – The picture below depicts the program when the Port has been opened

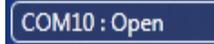


- a. Select the COM port associated with your MMC-203 COM10 as discussed in section 2.1, step 5.
- b. Click the Open Port button to connect to the MMC-203 Open Port

- i. This button should change giving you the option to close the port



- c. The Port field should change to indicate the Port is Open



and the terminal should populate with some information about the system and then turn blank. You are now ready to start moving a stage with your MMC-203. For more information about this program see the MMC-100 MCP program guide.

- 2. More information – more information about the MMC-100 MCP can be found in the MMC-100 MCP program guide.

3. Technical Information

3.1 MMC-203 Specifications

Parameter	Description
Axes	1 (stackable up to 99 axes)
Motor Type	Stepper motors
Interface	USB 2.0 compliant
Commands	ASCII Commands
Trajectory Mode	Trapezoidal velocity profile
Servo Clock	10 kHz
Trajectory Update	1 kHz
Power Supply	Regulated 24V DC (1A per module/axis*)
Enclosure Dimensions	145 x 85 x 25
Software Interface	MMC-203 MCP, LabVIEW VI's

*A single power supply may be used per stack. Each module/axis requires 1A, therefore add up Individual module amperages to determine the power supply amperage requirement.

3.2 Serial Port Setup

If the MMC-203 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 *MMC-203 Drivers* or can be downloaded from:

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

Below are the virtual RS-232 configuration settings necessary for correct communication setup:

Software Parameter	Setting
Data Bits	8
Stop Bits	1
Parity	No
Handshake	No

Baud rate | 38400

3.3 RJ11 RS485 Bus

The RS485 Intermodular RJ11 connector connects directly to the same Serial bus as the FTDI interface above. The RS485 line needs a terminating resistor of 22kΩ or higher. This connector can be used to communicate with the MMC-100 in the place of the USB connection. For more on the RS-485 Intermodular RJ11 connector see the Appendix 6.4.

4. Operation

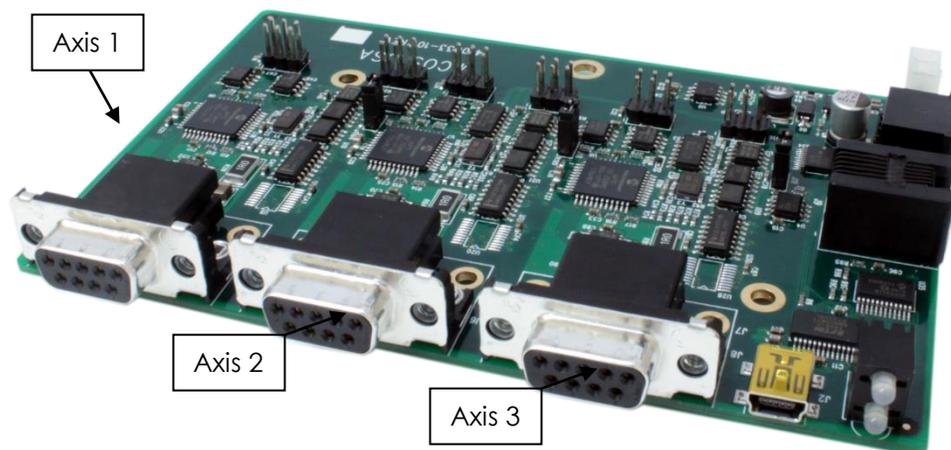
4.1 Axis Addressing

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.



4.2 Feedback Control

The MMC-203 has four different movement modes of operation. When executing a move command, the controller will drive a stage differently when set to different modes. The FBK command is used to switch between these modes.

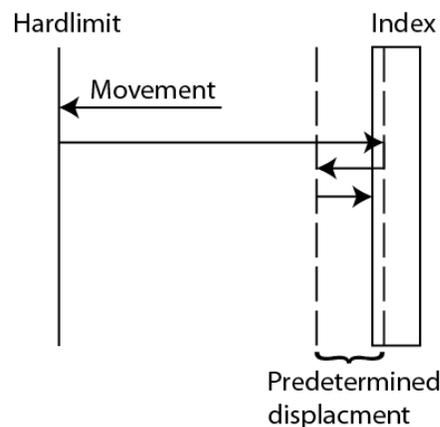
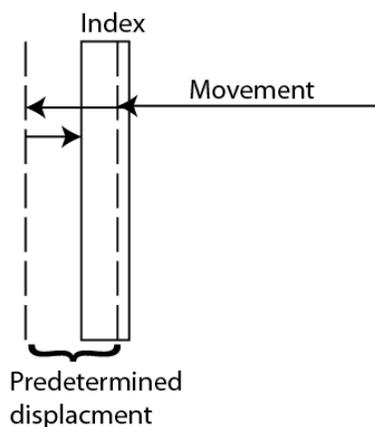
The first mode (nFBK0) is a traditional Open Loop. It follows a standard trapezoidal velocity characteristic. It bases the transition between acceleration, constant velocity and deceleration on the resolution settings (nREZx) or the distance it travels in one pulse. This is entirely theoretical and does not guarantee a set trajectory or end point.

The third mode (nFBK2) is a version of closed loop; meaning it takes position data from an attached encoder and uses it to ensure that it stops at the desired position. In this mode the controller runs in the second open loop mode (nFBK1) until it reaches the deceleration point. At this point it constantly reads from encoder and corrects its position to arrive at the correct position. This, unlike the first two modes can guarantee position within the specified deadband (DBN Command). However, this mode cannot guarantee a known trajectory.

The fourth mode (nFBK3) is a more traditional closed loop. The controller will constantly try to achieve an ideal trapezoidal velocity characteristic. Like the previous mode it too can guarantee position final within the specified deadband.

4.3 HOM, MLN, and MLP

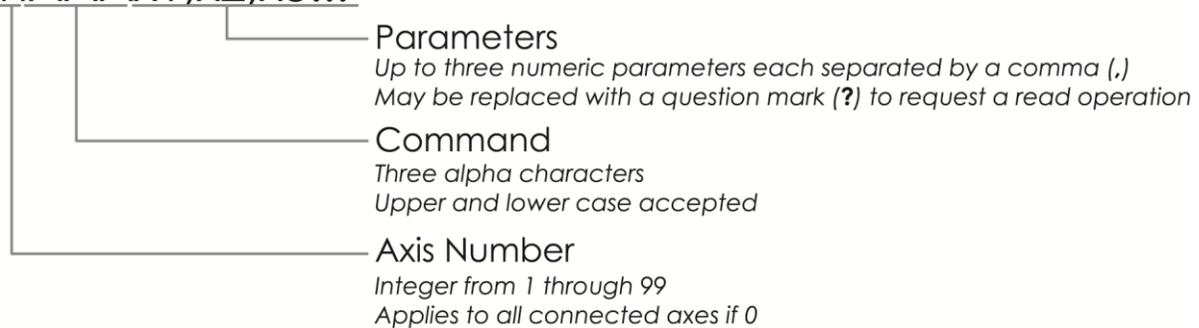
The HOM command all requires the attached stage to have an encoder. The MLN and MLP commands require either an attached encoder, or limit switches. HCG, LCG, LDR and LPL are all commands that affect the operation of either HOM or MLN and MLP. The HOM command will move negative direction by default. This can be changed using the HCG command. If the stage is above the index, it will move until it reaches the index then move a predetermined distance out of the index in the negative direction. The stage will then travel in the positive direction at a slower speed stopping at the edge of the index. If the stage is below the index it will move until it reaches a hard limit or the maximum travel. It then reverses direction and proceeds until it reaches the index. It will then travel a predetermined distance out of the index in the negative direction and finally travel toward the index at a slower velocity finally resting on the edge of the index. The HOM command will always home to the negative side of the limit.



5. Commands

5.1 Command Line Syntax

nAAAx1,x2,x3...



There are three components to every command prompt. The first is the "Axis Number" 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 "Command", 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 "Parameter". 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 (spaces and tabs) are ignored in the command format. The following are examples of equivalent commands:

```
4DBD13,45
4 DBD 13 , 45
```

5.2 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:

```
1MVR16;3MVR12 | Axis 1, Move 16 mm [16 degrees]; Axis 3, Move 12 mm [12 degrees]
```

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.

5.3 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/s² [degrees/s²].

5.4 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".

5.5 Synchronous Move

It is possible to move multiple motion devices at the same time, or extremely close to, by setting up and executing a synchronous move. To set up a synchronous move, use the MSA and MSR commands on the same command line (up to 8 allowed) or on separate lines followed by a line terminator. To execute the move, use the RUN command on the proceeding command line followed by a line terminator. For example;

1MSA4 ; 2MSA4 ; 3MSA4	Axis 1, Move 4mm; Axis 2, Move 4mm; Axis 3 Move 4mm
ORUN	Run Synchronous Move
Or	
1MSA4	Axis 1, Move 4mm
2MSA4	Axis 2, Move 4mm
3MSA4	Axis 3 Move 4mm
ORUN	Run Synchronous Move

5.6 Program Mode

A program may be used to save time when repeatedly using a sequence of commands. Each controller or axis must be programmed individually; however, multiple controllers may execute the same program at the same time.

A list of available program numbers may be viewed with the PGM? command. Existing program numbers may not be overridden unless previously erased using the ERA command.

To record a program sequence, enter the PGM command on a unique line followed by a line terminator. End a program sequence by entering the END command on a unique line followed by a line terminator. When you want to execute this program, use the EXC command. See the *Summary of Commands* page for a list of program compatible commands and more information about the PGM, END and EXC commands.

5.7 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:

```
1VEL0.005 \n\r      | Axis 1, Set velocity to .005 mm/s [degrees/s2] [New line, Carriage Return]
```

5.8 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.
15	Move Limit Requires Encoder	The MLN and MLP commands require 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	<p>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.</p> <p>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.</p>
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.
32	Incorrect Jog Velocity Request	The jog velocity can only be changed during motion by using a new JOG command. If the VEL command is used to change the velocity, this error will be triggered. The VEL command can only be used to change velocity during motion initiated by the move commands [MVR, MVA, MSR, MSA].
33	Not In Jog Mode	Sending a JOG command during motion initiated by a move command will trigger this error. To initiate Jog Mode, the controller should be at stand-still. To change velocity during a move, use the VEL command.
34	Trace Already In Progress	This error is triggered when a new trace command is received after a trace is already in progress. Trace settings may be modified only if the trace hasn't started recording data. Otherwise, wait until the trace has finished before modifying the trace settings.
35	Trace Did Not Complete	An error occurred while recording trace data. Try the operation again.

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.
37	Move Outside Soft Limits	If a requested move will take the controller outside of the preset travel limits, then the command will not be executed.
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.
39	Program Number Out of Range	The number entered for the program number was either less than 1 or greater than 16.
40	Program Size Limit Exceeded	The program has exceeded the character limit of 4 Kb.
41	Program failed to Record	Error in recording program. Erase program and try operation again.
42	End Command Must Be on its Own Line	The End command used to end a program must be on a separate line from all other instructions.
43	Failed to Read Program	An error occurred while trying to read a program. Try the Operation again.
44	Command Only Valid Within Program	The command that triggered this error is only suitable for use within a program.
45	Program Already Exists	A program already exists for the indicated program parameter. The program must be erased with the ERA command before being written again.
46	Program Doesn't Exist	The indicated program does not exist. This error can occur when you try to execute a program number that has not had a program assigned to it.
47	Read Operations Not Allowed Inside Program	Read Operations are not permitted in programs.
48	Command Not Allowed While Program in Progress	The command that triggered this error was given while a program was executing.
50	Limit Activated	Motion in the direction of the activated limit switch is disallowed if limit switches are enabled.

51	End of Travel Limit	The requested move will take the controller outside of its valid travel range, therefore the move is disallowed.
52	Home In Progress	A Home or a Move To Limit Procedure is in progress. Motion commands are disallowed during this time. A STP or EST command can be used to terminate the Home, and then a motion command can be sent.
53	IO Function Already In Use	The I/O Function in question is already assigned to another I/O pin. Some Functions can only be assigned to one pin at a time. See the documentation for each function for more details.
54	Invalid Resolution	The parameters entered for Gear Ratio, Lead-Screw Pitch, and Full Steps Per Revolution result in a resolution that cannot be handled by the controller.
55	Limits Are Not Configured Properly	Both Limit Switches are active, so motion is disallowed in both directions. Most likely the LPL(Limit Polarity command) setting should be switched.
80	Command Not Available in this Version	The command entered is not supported in this version of the firmware.
81	Analog Encoder Not Available In this Version	The current version of firmware installed does not support Analog Encoders.

6. Appendix

6.1 Encoder Input Pin-out

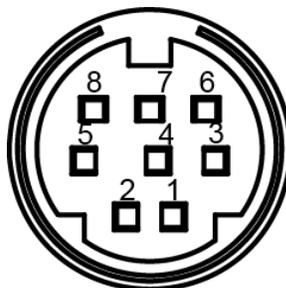
Pin	Color	Description
1	Brown	A+/Cos+
2	Red	B+/Sin+
3	Orange	Index +
4	Yellow	Ground
5	Green	+5V
6	Blue	A-/Cos-
7	Purple	B-/Sin-
8	Grey	Index -
9	Black	Not In Use

6.2 Motor Pin-out

Pin	Description
1	Phase A+
2	Phase A-
3	Phase B+
4	Phase B-
5	Ground
6	LSN
7	LSP
8	+5V
9	Ground

6.3 8-Pin Din IO connector

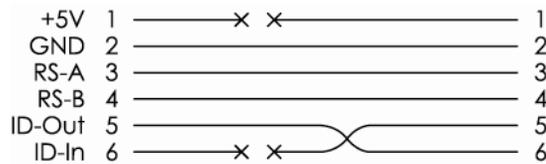
Pin8 - +5V (output)
 Pin7 - IO1 (output only)
 Pin6 - IO2
 Pin5 - IO3
 Pin3 - IO4
 Pin4 - GND



6.4 RS-485 Intermodular Connector Cable Pin-out

The RS-485 Intermodular Connector Cable is used to daisy chain two MMC 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. This connector can also be used to communicate with the controller in place of the USB connection by connecting your RS-485 Bus to pins 2, 3 and 4 as they appear below.

RS-485 Intermodular Connector Cable Pin-out:



6.5 Power supply pin out

