This page is intentionally left blank.
Contents

About This Guide .................................................................................................................................................. 9
Introduction ......................................................................................................................................................... 9
 Chapters and Appendixes in This Guide ........................................................................................................... 9
Information about the DMM-0400 .................................................................................................................... 10
About This Release ............................................................................................................................................ 10
Viewing the PDF Version .................................................................................................................................. 10
Symbols ............................................................................................................................................................. 10
Regulatory Compliance Certification ................................................................................................................ 11
Contact Information ......................................................................................................................................... 12

Chapter 1 – Site Planning ................................................................................................................................ 13
About This Chapter ........................................................................................................................................... 13
Introduction ....................................................................................................................................................... 13
Topics ................................................................................................................................................................. 13
Customer Responsibilities .................................................................................................................................. 14
Requirements for the Installation Site ............................................................................................................... 14
Preliminary Steps ............................................................................................................................................. 14
Site Requirements ........................................................................................................................................... 15
Requirements for Receiving and Setup ........................................................................................................... 15
Specifications for Power, Communications, and Environment ..................................................................... 16
Electrical Power ................................................................................................................................................ 16
Electrical receptacle .......................................................................................................................................... 16
Grounding .......................................................................................................................................................... 16
Power Cord and Communications Cables ......................................................................................................... 17
Operating Environment .................................................................................................................................... 17
Where to Go Next ............................................................................................................................................ 18

Chapter 2 – Technical Overview ................................................................................................................... 19
About This Chapter ........................................................................................................................................... 19
Introduction ....................................................................................................................................................... 19
Topics ................................................................................................................................................................. 19
Technical Description ....................................................................................................................................... 20
Motion Control Capabilities .............................................................................................................................. 20
Description ....................................................................................................................................................... 21
Front Panel ....................................................................................................................................................... 22
Rear Panel ......................................................................................................................................................... 23
Chapter 3 – Installing the DMM-0400

About This Chapter ................................................................................................................................. 36
Introduction ............................................................................................................................................... 36
Topics ...................................................................................................................................................... 36
Installing the System ................................................................................................................................. 37
Modes of Operation ................................................................................................................................. 37
Connecting to the DMM-0400 .................................................................................................................. 38
Installing the Motor and Encoder Cables .............................................................................................. 38
Installing the I/O Cables .......................................................................................................................... 38
Installing the Power Cord and Communications Cables ........................................................................ 38
Where to Go Next .................................................................................................................................... 39

Chapter 4 – Installing the Application Software for the DMM-0400

About This Chapter ................................................................................................................................. 40
Introduction ............................................................................................................................................... 40
Topics ...................................................................................................................................................... 40
Software Description ............................................................................................................................... 41
Installing PMX-4EX-SA Software .......................................................................................................... 41
Installing PMX Drivers ............................................................................................................................. 42
Where to Go Next .................................................................................................................................... 43

Chapter 5 – Running the DMM-0400 via PMX-4EX-SA Software

About This Chapter ................................................................................................................................. 44
Introduction ............................................................................................................................................... 44
Topics ...................................................................................................................................................... 44
Connecting a USB Cable .......................................................................................................................... 45
Establishing Communication with the DMM-0400 .................................................................................. 46
Using the Main Control Screen........................................................................................................... 47
Status Screen (A)................................................................................................................................... 48
Control (B)............................................................................................................................................. 49
On-The-Fly-Speed Control (C).............................................................................................................. 51
On-The-Fly-Position Control (D)........................................................................................................... 51
Product Information (E)......................................................................................................................... 51
Sync Outputs (F).................................................................................................................................... 52
Digital Input/Output (G)........................................................................................................................... 52
Analog Inputs (H).................................................................................................................................... 52
Program File Control (I).......................................................................................................................... 53
Standalone Program Editor (J)............................................................................................................... 53
Standalone Program Control (K)............................................................................................................ 54
Standalone Program Compile/Download/Upload (L).............................................................................. 54
Setup (M)............................................................................................................................................... 55
Terminal (N).......................................................................................................................................... 56
Latches (O)............................................................................................................................................. 57
Variable Status (P).................................................................................................................................. 58
About (Q)............................................................................................................................................... 58
Where to Go Next................................................................................................................................... 59

Chapter 6 – Running the DMM-0400 via Front Panel Buttons................................................................. 60
About This Chapter............................................................................................................................... 60
Introduction........................................................................................................................................... 60
Topics .................................................................................................................................................... 60
Front Panel Buttons............................................................................................................................... 61
DMM-0400 Stored programs .................................................................................................................. 62
Executing Stored Programs................................................................................................................... 62
Where to Go Next.................................................................................................................................. 63

Chapter 7 – Editing Programs.................................................................................................................. 64
About This Chapter............................................................................................................................... 64
Introduction........................................................................................................................................... 64
Topics .................................................................................................................................................... 64
Editing a Stored Program...................................................................................................................... 65
Standalone Program Specification......................................................................................................... 67
Programming Language Specification................................................................................................... 69
Sample Programs................................................................................................................................. 90
Where to Go Next.................................................................................................................................. 94
Chapter 8 – Interactive Commands & USB Communications .......................................................... 95

About This Chapter .................................................................................................................. 95
Introduction ............................................................................................................................. 95
Topics ....................................................................................................................................... 95
USB Communication Description .......................................................................................... 96
Motion Control Overview and Interactive Command Description ........................................... 98
ASCII Programming Language ............................................................................................... 116
Error Codes ............................................................................................................................ 124
Where to Go Next ..................................................................................................................... 125

Appendix ................................................................................................................................... 126
About This Chapter .................................................................................................................. 126
Introduction ............................................................................................................................. 126
Topics ....................................................................................................................................... 126
Speed Settings .......................................................................................................................... 127
Acceleration/Deceleration Range ............................................................................................ 127
Acceleration/Deceleration Range – Positional Move ............................................................... 128
Review/Revision History .......................................................................................................... 129
List of Tables

Table 1  Weights and Measurements ................................................................. 15
Table 2  Specifications for Electrical power ....................................................... 16
Table 3  Specifications for Power and Communications Cables ......................... 17
Table 4  Specifications for the Operating Environment ...................................... 17
Table 5  Rear Panel Connectors ........................................................................ 24
Table 6  Rear Panel Switches & Jumpers ........................................................... 24
Table 7  Pinouts – Serial Communications ......................................................... 25
Table 8  Pinouts – USB Communication ............................................................ 25
Table 9  Pinouts – Encoder/Limits ..................................................................... 26
Table 10 Pinouts – Motors ................................................................................ 26
Table 11 Pinouts – Encoder Out X & Y ............................................................... 27
Table 12 Pinouts – Encoder Out Z & U ............................................................... 27
Table 13 Pinouts – Digital I/O ......................................................................... 28
Table 14 Pinouts – Analog I/O & Joystick ......................................................... 29
Table 15 Pinouts – Analog Inputs .................................................................. 29
Table 16 Current Switch Settings ..................................................................... 30
Table 17 Jumper Selection – JP1 – JP8 ............................................................... 31
Table 18 Fuse LEDs ....................................................................................... 32
Table 19 Enable LEDs ................................................................................... 32
Table 20 Standard Programs ......................................................................... 62
Table 21 Standalone Run on Boot-Up ............................................................... 68
Table 22 Storing to Flash ................................................................................. 68
Table 23 Pulse Speed .................................................................................... 99
Table 24 Motor Status .................................................................................. 100
Table 25 Polarity ............................................................................................ 107
Table 26 Latch Inputs .................................................................................... 108
Table 27 Latch Return Values ....................................................................... 108
Table 28 Digital Inputs ................................................................................ 108
Table 29 Digital Outputs .............................................................................. 109
Table 30 Enable Outputs .............................................................................. 109
Table 31 Sync Outputs ................................................................................ 110
Table 32 Joystick Control .............................................................................. 111
Table 33 Summary of Joystick Control Parameters ....................................... 112
Table 34 StepNLoop Closed Loop Control .................................................... 113
Table 35 StepNLoop Return Values ............................................................... 114
Table 36 StepNLoop Conditions .................................................................. 114
Table 37 Baud Rate Setting .......................................................................... 115
Table 38 ASCII Commands .......................................................................... 116
Table 39 Error Codes .................................................................................. 124
Table 40 Speed Settings ............................................................................ 127
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>DMM-0400 Dimensions</td>
<td>15</td>
</tr>
<tr>
<td>Figure 2</td>
<td>DMM-0400 ISO View</td>
<td>21</td>
</tr>
<tr>
<td>Figure 3</td>
<td>DMM-0400 Front View</td>
<td>22</td>
</tr>
<tr>
<td>Figure 4</td>
<td>DMM-0400 Rear View</td>
<td>23</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Digital Inputs</td>
<td>33</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Digital Outputs</td>
<td>33</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Encoder Inputs</td>
<td>34</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Analog Inputs</td>
<td>34</td>
</tr>
<tr>
<td>Figure 9</td>
<td>System Installation</td>
<td>37</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Performax Installation</td>
<td>41</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Driver Installation</td>
<td>42</td>
</tr>
<tr>
<td>Figure 12</td>
<td>DMM-0400 USB Port</td>
<td>45</td>
</tr>
<tr>
<td>Figure 13</td>
<td>PMX-2EX-SA Application</td>
<td>46</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Communication Dialog Box</td>
<td>46</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Select Control Box</td>
<td>47</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Main Control Screen</td>
<td>47</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Status Screen (A)</td>
<td>48</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Status Screen (B)</td>
<td>49</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Buffered I Move</td>
<td>49</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Arc/Circle Tool</td>
<td>50</td>
</tr>
<tr>
<td>Figure 21</td>
<td>On-The-Fly-Speed Control (C)</td>
<td>51</td>
</tr>
<tr>
<td>Figure 22</td>
<td>On-The-Fly-Position Control (D)</td>
<td>51</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Product Information (E)</td>
<td>51</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Sync Output Status (F)</td>
<td>52</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Digital I/O Status (G)</td>
<td>52</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Analog Inputs (H)</td>
<td>52</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Program File Control (I)</td>
<td>53</td>
</tr>
<tr>
<td>Figure 28</td>
<td>Standalone Program Editor (J)</td>
<td>53</td>
</tr>
<tr>
<td>Figure 29</td>
<td>Standalone Program Control (K)</td>
<td>54</td>
</tr>
<tr>
<td>Figure 30</td>
<td>Standalone Program Compile/Download/Upload (L)</td>
<td>54</td>
</tr>
<tr>
<td>Figure 31</td>
<td>Setup (M)</td>
<td>55</td>
</tr>
<tr>
<td>Figure 32</td>
<td>Terminal (N)</td>
<td>56</td>
</tr>
<tr>
<td>Figure 33</td>
<td>Latch Inputs (O)</td>
<td>57</td>
</tr>
<tr>
<td>Figure 34</td>
<td>Variables Status (P)</td>
<td>58</td>
</tr>
<tr>
<td>Figure 35</td>
<td>About Button (Q)</td>
<td>58</td>
</tr>
<tr>
<td>Figure 36</td>
<td>DMM-0400 Front Panel</td>
<td>61</td>
</tr>
<tr>
<td>Figure 37</td>
<td>Retrieving a Program</td>
<td>65</td>
</tr>
<tr>
<td>Figure 38</td>
<td>Text Window</td>
<td>66</td>
</tr>
<tr>
<td>Figure 39</td>
<td>USB Cable</td>
<td>97</td>
</tr>
<tr>
<td>Figure 40</td>
<td>Trapezoidal Velocity Profile</td>
<td>98</td>
</tr>
<tr>
<td>Figure 41</td>
<td>S-Surve Velocity Profile</td>
<td>98</td>
</tr>
<tr>
<td>Figure 42</td>
<td>Circular Interpolation Moves</td>
<td>102</td>
</tr>
<tr>
<td>Figure 43</td>
<td>Arc Interpolation Moves</td>
<td>103</td>
</tr>
<tr>
<td>Figure 44</td>
<td>Homing - Home Input Only (High Speed)</td>
<td>104</td>
</tr>
<tr>
<td>Figure 45</td>
<td>Homing - Limit Only</td>
<td>105</td>
</tr>
<tr>
<td>Figure 46</td>
<td>Homing - Home and Z-Index</td>
<td>105</td>
</tr>
<tr>
<td>Figure 47</td>
<td>Homing - Z-Index Only</td>
<td>106</td>
</tr>
<tr>
<td>Figure 48</td>
<td>Homing - Home Input Only (High Speed and Low Speed)</td>
<td>106</td>
</tr>
<tr>
<td>Figure 49</td>
<td>Joystick Control</td>
<td>111</td>
</tr>
<tr>
<td>Figure 50</td>
<td>Acceleration/Deceleration Profile</td>
<td>128</td>
</tr>
</tbody>
</table>
About This Guide

Introduction

This guide describes the installation of the DMM-0400 and the operation of it. The intended audiences are the customer, technician responsible for performing the installation and preparing the site beforehand, and the operator running the system.

This point forward, “About This Guide,” covers the following topics.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information about the DMM-0400</td>
<td>10</td>
</tr>
<tr>
<td>Symbols</td>
<td>10</td>
</tr>
<tr>
<td>Regulatory Compliance Certification</td>
<td>11</td>
</tr>
<tr>
<td>Contact Information</td>
<td>12</td>
</tr>
</tbody>
</table>

Chapters and Appendixes in This Guide

This guide contains the following chapters. It does not cover advanced topics such as debugging the system or editing configuration files.

<table>
<thead>
<tr>
<th>For</th>
<th>Go to page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chapter 1</strong> Site Planning</td>
<td>13</td>
</tr>
<tr>
<td>Describes the responsibilities of the customer and the requirements for the installation site, and includes an installation process flow and installation procedure checklist.</td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 2</strong> Technical Overview</td>
<td>19</td>
</tr>
<tr>
<td>Describes the functionality of the DMM-0400 and its main components.</td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 3</strong> Installing the DMM-0400</td>
<td>36</td>
</tr>
<tr>
<td>Describes how to install the DMM-0400 into your system.</td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 4</strong> Installing Application Software for the DMM-0400</td>
<td>40</td>
</tr>
<tr>
<td>Explains how to install application software on your PC.</td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 5</strong> Running the DMM-0400 via PMX-4EX-SA Software</td>
<td>44</td>
</tr>
<tr>
<td>Explains how to run the DMM-310 using only the GUI on your PC.</td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 6</strong> Running the DMM-0400 via Front Panel Buttons</td>
<td>60</td>
</tr>
<tr>
<td>Explains how to run the DMM-0400 using only the front panel buttons.</td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 7</strong> Editing Programs</td>
<td>64</td>
</tr>
<tr>
<td>Explains how to edit stored programs.</td>
<td></td>
</tr>
<tr>
<td><strong>Chapter 8</strong> Interactive Commands &amp; USB Communications</td>
<td>95</td>
</tr>
<tr>
<td>Describes the USB communications and ASCII commands to interactively communicate to the DMM-0400.</td>
<td></td>
</tr>
<tr>
<td><strong>Appendix</strong></td>
<td>126</td>
</tr>
<tr>
<td>Supporting documentation for the DMM-0400 manual</td>
<td></td>
</tr>
</tbody>
</table>
Information about the DMM-0400

About This Release

This version of the operations manual supports the DMM-0400 GEN1 motion controller.

Viewing the PDF Version

The PDF version of this guide provides a number of ways to navigate through the content. Blue text indicates links to related topics. You can also do the following:

- Click the Bookmarks tab in the left pane to view the list of bookmarks and click any bookmark to navigate to that topic. If the Bookmarks tab is not visible, choose View > Navigation > Bookmarks from the menu bar.
- Click on the Contents bookmark. Pick any entry in the Contents and click either the entry text or its page number to go to that topic.
- Scroll to the end of the Contents to the List of Figures. Pick any entry in the list and click either the figure number and title or its page number to go to that topic.
- Scroll to the end of the List of Figures to the List of Tables. Pick any entry in the list and click either the table number and title or its page number to go to that topic.
- Click on the Index bookmark. Pick any index entry and click its page number to go to that topic.

Symbols

This guide uses the following symbols.

Note: Indicates neutral or positive information that emphasizes or supplements important points of the main text.

Caution: Advises users that failure to take or avoid a specified action could result in loss of data.

Warning: Advises users that failure to take or avoid a specified action could result in physical harm to the user or hardware.
Regulatory Compliance Certification

[TESTING FOR COMPLIANCE IS IN PROCESS.]

The DMM-0400 hardware and its components meet or exceed the requirements of the following regulatory agencies. Applicable labels indicating compliance with these requirements appear on the hardware.
Contact Information

DMM-0400 Contact Information

The Business Unit of Dover Motion in Boxborough, MA is the manufacturer of the DMM-0400. The location of the unit is:

Dover Motion
159 Swanson Rd.
Boxborough, MA 01719

The Dover Motion Website is located at:

http://www.Dovermotion.com

If you have questions or comments, contact:

Phone: 508-475-3400
Email: sales@dovermotion.com

Or for Sales, contact:

Eastern U.S. & Canada
Phone: 508-475-3400
Email: eastsales@dovermotion.com

Western U.S. & Canada
Phone: 508-475-3400
Email: westsales@dovermotion.com
Chapter 1
Site Planning

About This Chapter
Introduction

This chapter describes the responsibilities of the customer and the requirements for the installation site.

Topics

This chapter covers the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Responsibilities</td>
<td>14</td>
</tr>
<tr>
<td>Site Requirements</td>
<td>15</td>
</tr>
<tr>
<td>Specifications for Power, Communications, and Environment</td>
<td>16</td>
</tr>
<tr>
<td>Where to Go Next</td>
<td>18</td>
</tr>
</tbody>
</table>
Customer Responsibilities

Requirements for the Installation Site

For installation of the DMM-0400, you are responsible for supplying the following:

- An electrical receptacle available for power to the system. See “Electrical Receptacle” on page 16 for specifications.

  Warning: If you use a different type of power cord, plug, or receptacle, other than those specified in the guide, you are responsible for having a qualified electrician install them.

- A suitable bench area for the DMM-0400. See bench specifications in the Table 1.
- One 2.0 USB cables (optional).

  Note: The USB cable is not required for the initial installation since the system software is already installed on the DMM-0400 when shipped. This cable is for customer use.

Preliminary Steps

Perform the following preliminary steps to prepare for the installation:

1. Complete the Site Requirements Checklist. See Site Requirements section.

  Note: Do not attempt to perform an installation unless the requirements specified in this guide have been carried out.
Site Requirements

Requirements for Receiving and Setup

Receiving

When you receive your DMM-0400, do the following:

1. Inspect the package for damage.

   **Caution:** If you suspect damage, take pictures of the location where you suspect there is damage. A representative of the shipping carrier may need to be present during the unpacking process for insurance purposes.

Setup

Select a location where you want the DMM-0400 to be unpacked and installed. Please ensure the following parameters are met:

**Table 1  Weights and Measurements**

<table>
<thead>
<tr>
<th>Specification</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DMM-0400</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>4.88 Kg (approx. 10.75 lbs)</td>
<td></td>
</tr>
<tr>
<td>Dimensions (maximum):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>38.56 cm (approx. 15.18 inches)</td>
<td></td>
</tr>
<tr>
<td>Depth</td>
<td>25.40 cm (approx. 10.00 inches)</td>
<td></td>
</tr>
<tr>
<td>Height</td>
<td>15.24 cm (approx. 6.00 inches)</td>
<td></td>
</tr>
</tbody>
</table>

![DMM-0400 Dimensions](image)
Specifications for Power, Communications, and Environment

**Electrical Power**

The DMM-0400 requires a main power source, as specified in Table 2. Ensure that the AC power source is easily accessible.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>90-264VAC ± 10%</td>
</tr>
<tr>
<td>Frequency</td>
<td>47-63 Hz</td>
</tr>
<tr>
<td>Current (at nominal voltage, 110VAC)</td>
<td>Typical, maximum, inrush surge:</td>
</tr>
<tr>
<td></td>
<td>No more than 11 amps.</td>
</tr>
</tbody>
</table>

**Electrical receptacle**

The DMM-0400 requires a NEMA 5-15 receptacle in North America. The electrical receptacle must be located within 2.5 meters (approx. 8 feet) of the power cord input on the system.

**Note:** The electrical receptacle must have a dedicated 1.2kVA power line and ground.

**Grounding**

Certain types of electrical noise are greatly exaggerated by poor or improper electrical ground connections. To prevent these problems, it is desirable to have a dedicated circuit and ground for use by the DMM-0400.

**Note:** For grounding wiring, the DMM-0400 uses internal AC ground connections with green/yellow 16AWG wiring.

**Warning:** Do not connect the DMM-0400 to the same dedicated line and ground that is used to power a device with a high current.

**Warning:** Use qualified personnel for installation of all electrical fixtures, and ensure that all installations follow local bylaws.
Power Cord and Communications Cables

Table 3 lists the suggested lengths for power cord and communications cables for the DMM-0400. The DMM-0400 shipment includes only the power cord.

<table>
<thead>
<tr>
<th>Cable Type</th>
<th>Length(^1)</th>
<th>Maximum Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMM-0400 power cord (supplied):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEMA5-15 (North America-specific)</td>
<td>2.4 meters (8 feet)</td>
<td>3.6 meters (12 feet)</td>
</tr>
<tr>
<td>Communications cables (not supplied):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USB 2.0 cable A-B</td>
<td>3 meters (10 feet)</td>
<td>30 meters (100 feet)</td>
</tr>
</tbody>
</table>

---

**Warning:** If you use a different type of power cord, plug, or receptacle, other than those specified in this guide, you are responsible for having a qualified electrician install them.

Operating Environment

Table 4 lists the specifications for the environment in which the DMM-0400 will be fully operational and compliant with its performance specifications.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Acceptable Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>15-30(^\circ) C</td>
</tr>
<tr>
<td></td>
<td>59-86(^\circ) F</td>
</tr>
<tr>
<td>Relative humidity</td>
<td>20-80% non-condensing</td>
</tr>
<tr>
<td>Altitude</td>
<td>0 to 2,000 meters above sea level</td>
</tr>
<tr>
<td></td>
<td>0 to 6,600 feet above sea level</td>
</tr>
</tbody>
</table>

\(^1\) The cord and cable lengths specified in this table are suggested lengths. You must supply the cord or cable if you need to use anything longer than these (see the column, Maximum length).
Where to Go Next

Continue with the next chapter, *Chapter 2 - Technical Overview*. The next section will allow you to become familiar with the functionalities, capabilities, and main components of the DMM-0400.
Chapter 2
Technical Overview

About This Chapter

Introduction

This chapter describes the functionalities, capabilities, and main components of the DMM-0400 from a technical perspective. This chapter should provide the user with more detail and familiarity regarding the operation DMM-0400.

Topics

This chapter covers the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Description</td>
<td>20</td>
</tr>
<tr>
<td>Front Panel</td>
<td>22</td>
</tr>
<tr>
<td>Rear Panel</td>
<td>23</td>
</tr>
<tr>
<td>Rear Panel Connector Pinouts</td>
<td>25</td>
</tr>
<tr>
<td>Rear Panel Jumper and Switch Selection</td>
<td>30</td>
</tr>
<tr>
<td>Rear Panel LEDs</td>
<td>32</td>
</tr>
<tr>
<td>Interface Circuitry</td>
<td>33</td>
</tr>
<tr>
<td>Where to Go Next</td>
<td>35</td>
</tr>
</tbody>
</table>
Technical Description

Motion Control Capabilities

The DMM-0400 controller (along with the DMM-0200) is a family of programmable, motion controllers. It is highly customizable and can control up to four axis of motion.

A wide range of applications using stepper motors can be controlled using the DMM-0400. The versatile yet low cost and highly efficient design of the DMM-0400 allows the user to satisfy a wide range of motion control requirements.

See below for part number information:

Ordering Information:

<table>
<thead>
<tr>
<th>Number of Step Axes (1 - 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>DMM - 0420</td>
</tr>
</tbody>
</table>

The DMM-0400 can be ordered with 1 – 4 step axes per controller. Please contact Dover Motion Sales group for further information.
Description

The DMM-0400 can control stepper motors. The figures in this section will describe the axis configurations for the DMM-0400 as well as interfacing with the controller.

Caution: To avoid overheating, always operate the DMM-0400 with the enclosure cover on to allow for proper air flow through it.

Note: Removing the enclosure cover is only necessary for troubleshooting.

Figure 2 DMM-0400 ISO View
Front Panel

Figure 3 DMM-0400 Front View

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>POWER</td>
<td>On/Off power switch</td>
</tr>
<tr>
<td>2</td>
<td>START</td>
<td>Start motion program</td>
</tr>
<tr>
<td>3</td>
<td>STOP</td>
<td>Stop motion program</td>
</tr>
<tr>
<td>4</td>
<td>PROGRAM SELECT</td>
<td>Selects the motion program number</td>
</tr>
</tbody>
</table>
There are 4 axis of motion – X, Y, Z, and U. The connectors for axis X are shown in detail here. The connectors and LEDs for Y, Z, and U are identical to the X axis.
Rear Panel Connectors & Switches

The following table lists the connector positions on the rear panel and the type of connector for each.

Table 5 Rear Panel Connectors

<table>
<thead>
<tr>
<th>Connector Position</th>
<th>Description</th>
<th>Connector Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>J4-SERIAL X</td>
<td>Serial data communications, Axis 1, X-axis</td>
<td>DE-9-M</td>
</tr>
<tr>
<td>J21-SERIAL Y</td>
<td>Serial data communications, Axis 2, Y-axis</td>
<td>DE-9-M</td>
</tr>
<tr>
<td>J17-SERIAL Z</td>
<td>Serial data communications, Axis 3, Z-axis</td>
<td>DE-9-M</td>
</tr>
<tr>
<td>J33-SERIAL U</td>
<td>Serial data communications, Axis 4, U-axis</td>
<td>DE-9-M</td>
</tr>
<tr>
<td>J10-ENC/LIM X</td>
<td>Encoder/Limits, Axis 1 (positions and limits for X-axis)</td>
<td>HD-15-F</td>
</tr>
<tr>
<td>J24-ENC/LIM Y</td>
<td>Encoder/Limits, Axis 2 (positions and limits for Y-axis)</td>
<td>HD-15-F</td>
</tr>
<tr>
<td>J35-ENC/LIM Z</td>
<td>Encoder/Limits, Axis 3 (positions and limits for Z-axis)</td>
<td>HD-15-F</td>
</tr>
<tr>
<td>J37-ENC/LIM U</td>
<td>Encoder/Limits, Axis 4 (positions and limits for U-axis)</td>
<td>HD-15-F</td>
</tr>
<tr>
<td>J11-MOTOR X</td>
<td>Motor, Axis 1 (motor for X-axis)</td>
<td>DE-9-F</td>
</tr>
<tr>
<td>J26-MOTOR Y</td>
<td>Motor, Axis 2 (motor for Y-axis)</td>
<td>DE-9-F</td>
</tr>
<tr>
<td>J23-MOTOR Z</td>
<td>Motor, Axis 3 (motor for Z-axis)</td>
<td>DE-9-F</td>
</tr>
<tr>
<td>J25-ENC OUT X/Y</td>
<td>Encoder out, Axis 1 &amp; 2, XY axis</td>
<td>DA-15-M</td>
</tr>
<tr>
<td>J36-ENC OUT U/Z</td>
<td>Encoder out, Axis 3 &amp; 4, U/Z axis</td>
<td>DA-15-M</td>
</tr>
<tr>
<td>J18-DIGITAL I/O</td>
<td>Opto-isolated inputs and outputs</td>
<td>DB-25-F</td>
</tr>
<tr>
<td>J19-ANALOG I/O</td>
<td>Joystick connection</td>
<td>DA-15-F</td>
</tr>
<tr>
<td>J13-ANALOG IN</td>
<td>Analog inputs</td>
<td>DE-9-M</td>
</tr>
<tr>
<td>USB</td>
<td>USB communications, all axis</td>
<td>USB</td>
</tr>
<tr>
<td>AC IN</td>
<td>AC Power</td>
<td>NEMA-5-15P</td>
</tr>
</tbody>
</table>

The following table lists the switches and jumpers on the rear panel.

Table 6 Rear Panel Switches & Jumpers

<table>
<thead>
<tr>
<th>Switch/Jumper Position</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW1 CUR X</td>
<td>Current limit DIP switch for X axis stepper</td>
</tr>
<tr>
<td>SW2 CUR Y</td>
<td>Current limit DIP switch for Y axis stepper</td>
</tr>
<tr>
<td>SW3 CUR Z</td>
<td>Current limit DIP switch for Z axis stepper</td>
</tr>
<tr>
<td>SW4 CUR U</td>
<td>Current limit DIP switch for U axis stepper</td>
</tr>
<tr>
<td>JP1</td>
<td>Rear panel SPDT relay enable/disable</td>
</tr>
<tr>
<td>JP2</td>
<td>N/A</td>
</tr>
<tr>
<td>JP3</td>
<td>Front panel PROGRAM SELECT enable/disable</td>
</tr>
<tr>
<td>JP4</td>
<td>Front panel PROGRAM SELECT enable/disable</td>
</tr>
<tr>
<td>JP5</td>
<td>Front panel PROGRAM SELECT enable/disable</td>
</tr>
<tr>
<td>JP6</td>
<td>Front panel PROGRAM SELECT enable/disable</td>
</tr>
<tr>
<td>JP7</td>
<td>Front panel START button enable/disable</td>
</tr>
<tr>
<td>JP8</td>
<td>Front panel STOP button enable/disable</td>
</tr>
</tbody>
</table>
Rear Panel Connector Pinouts

Pinouts – Serial Communications (J4-SERIAL X, J21-SERIAL Y, J17-SERIAL Z, J33-SERIAL U)

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Function</th>
<th>Description</th>
<th>DE-9-M Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RX</td>
<td>Receive data</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TX</td>
<td>Transmit data</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7 Pinouts – Serial Communications

**Note:** The four serial connectors on the rear panel are used for setting up the individual drivers for each axis of motion. **This is already done at Dover Motion.**

Under normal circumstances, the user should not need to communicate to the individual drivers and therefore should not need to connect to these four serial ports.

Should you need to communicate to the individual drivers, please contact the Dover Motion Service Department for assistance.

Pinouts – USB Communication

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Function</th>
<th>Description</th>
<th>USB-B Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VCC</td>
<td>+5VDC</td>
<td>Pin 2 Pin 1</td>
</tr>
<tr>
<td>2</td>
<td>D-</td>
<td>Data-</td>
<td>Pin 3 Pin 4</td>
</tr>
<tr>
<td>3</td>
<td>D+</td>
<td>Data+</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
</tbody>
</table>

Table 8 Pinouts – USB Communication

**Note:** The user should only need to communicate to the DMM-0400 using the USB connection. The USB connection provides full control of the DMM-0400.

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Function</th>
<th>Description</th>
<th>HD-15-F Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+VLIM</td>
<td>Limit power (+5VDC)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+LIM</td>
<td>Positive limit</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-LIM</td>
<td>Negative limit</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CHASSIS</td>
<td>Earth ground</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
<td>Logic ground</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>+5VENC</td>
<td>Encoder power (+5VDC)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ENC A</td>
<td>Channel A</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ENC /A</td>
<td>Complement of channel A</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ENC B</td>
<td>Channel B</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ENC /B</td>
<td>Complement of channel B</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ENC Z</td>
<td>Channel Z</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ENC /Z</td>
<td>Complement of channel Z</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>HOME</td>
<td>Home switch</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>Logic ground</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>CHASSIS</td>
<td>Earth ground</td>
<td></td>
</tr>
</tbody>
</table>

Table 9 Pinouts – Encoder/Limits

### Pinouts – Motors (J11-MOTOR X, J26-MOTOR Y, J23-MOTOR Z, J39-MOTOR U)

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Function</th>
<th>Description</th>
<th>DE-9-F Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>STP A</td>
<td>Stepper motor coil A</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>STP B</td>
<td>Stepper motor coil B</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>STP /A</td>
<td>Stepper motor coil /A</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>FAULT</td>
<td>Motor fault</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>STP B/</td>
<td>Stepper motor coil /B</td>
<td></td>
</tr>
</tbody>
</table>

Table 10 Pinouts – Motors
## Pinouts – Encoder Out X & Y (J25-ENC OUT X/Y)

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Function</th>
<th>Description</th>
<th>DA-15-M Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ENC AY</td>
<td>Y axis – Encoder A channel</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ENC /AY</td>
<td>Y axis – Encoder /A channel</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ENC BY</td>
<td>Y axis – Encoder B channel</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ENC /BY</td>
<td>Y axis – Encoder /B channel</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ENC ZY</td>
<td>Y axis – Encoder Z channel</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ENC /ZY</td>
<td>Y axis – Encoder /Z channel</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ENC AX</td>
<td>X axis – Encoder A channel</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ENC /AX</td>
<td>X axis – Encoder /A channel</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ENC BX</td>
<td>X axis – Encoder B channel</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ENC /BX</td>
<td>X axis – Encoder /B channel</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ENC ZX</td>
<td>X axis – Encoder Z channel</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ENC /ZX</td>
<td>X axis – Encoder /Z channel</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
</tbody>
</table>

Table 11 Pinouts – Encoder Out X & Y

## Pinouts – Encoder Out Z & U (J36-ENC OUT Z/U)

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Function</th>
<th>Description</th>
<th>DA-15-M Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ENC AU</td>
<td>U axis – Encoder A channel</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ENC /AU</td>
<td>U axis – Encoder /A channel</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ENC BU</td>
<td>U axis – Encoder B channel</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ENC /BU</td>
<td>U axis – Encoder /B channel</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ENC ZU</td>
<td>U axis – Encoder Z channel</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ENC /ZU</td>
<td>U axis – Encoder /Z channel</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ENC AZ</td>
<td>Z axis – Encoder A channel</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ENC /AZ</td>
<td>Z axis – Encoder /A channel</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ENC BZ</td>
<td>Z axis – Encoder B channel</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ENC /BZ</td>
<td>Z axis – Encoder /B channel</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ENC ZZ</td>
<td>Z axis – Encoder Z channel</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ENC /ZZ</td>
<td>Z axis – Encoder /Z channel</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
</tbody>
</table>

Table 12 Pinouts – Encoder Out Z & U
## Pinouts – Digital I/O (J18-DIGITAL I/O)

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Function</th>
<th>Description</th>
<th>DB-25-F Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+OPTO LIM</td>
<td>Opto limit power (+5VDC)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+OPTO I/O</td>
<td>Opto I/O power (+5VDC)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>+12V INT</td>
<td>+12VDC</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>OPTO GND</td>
<td>Logic ground</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>OPTO GND</td>
<td>Logic ground</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>DIG IN1</td>
<td>Digital Input 1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DIG IN2</td>
<td>Digital Input 2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DIG IN3</td>
<td>Digital Input 3</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>DIG IN4</td>
<td>Digital Input 4</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DIG IN5</td>
<td>Digital Input 5</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>DIG IN6</td>
<td>Digital Input 6</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>DIG IN7</td>
<td>Digital Input 7</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>DIG IN8</td>
<td>Digital Input 8</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>DIG OUT1</td>
<td>Digital Output 1</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>DIG OUT2</td>
<td>Digital Output 2</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>DIG OUT3</td>
<td>Digital Output 3</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>DIG OUT4</td>
<td>Digital Output 4</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>DIG OUT5</td>
<td>Digital Output 5</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>DIG OUT6</td>
<td>Digital Output 6</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>DIG OUT7</td>
<td>Digital Output 7</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>DIG OUT8</td>
<td>Digital Output 8&lt;sup&gt;2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>RELAY N.C.</td>
<td>Relay N.O. connection&lt;sup&gt;2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>RELAY N.O.</td>
<td>Relay N.C. connection&lt;sup&gt;2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>RELAY COM</td>
<td>Relay common&lt;sup&gt;2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>SHIELD</td>
<td>Chassis ground</td>
<td></td>
</tr>
</tbody>
</table>

<sup>2</sup> Jumper JP1 enables and disables the relay function through digital output 8. Please see Jumper section for detailed description of the functionality.
## Pinouts – Analog I/O & Joystick (J19-ANALOG I/O JOYSTICK)

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Function</th>
<th>Description</th>
<th>DA-15-F Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+VANALOG</td>
<td>Analog power (+5VDC)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>STICK P/B</td>
<td>Joystick - Stick pushbutton, Digital input 1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ANALOG X</td>
<td>Joystick X analog input, Analog input 1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>GND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ANALOG Y</td>
<td>Joystick Y analog input, Analog input 2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>LEFT P/B</td>
<td>Joystick - Left pushbutton(^3), Digital input 2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ANALOG Z</td>
<td>Joystick Z analog input, Analog input 3</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>RIGHT P/B</td>
<td>Joystick - Right pushbutton(^4), Digital input 3</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 14 Pinouts – Analog I/O & Joystick

### Pinouts – Analog Inputs (J13-ANALOG IN)

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Function</th>
<th>Description</th>
<th>DE-9-M Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ANALOG 8</td>
<td>Analog Input 8</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ANALOG 6</td>
<td>Analog Input 6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ANALOG 4</td>
<td>Analog Input 4</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>N/C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ANALOG 7</td>
<td>Analog Input 7</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ANALOG 5</td>
<td>Analog Input 5</td>
<td></td>
</tr>
</tbody>
</table>

Table 15 Pinouts – Analog Inputs

---

\(^3\) Jumper JP8 enables and disables the left pushbutton control from the joystick. Please see Jumper section for detailed description of the functionality.

\(^4\) Jumper JP7 enables and disables the right pushbutton control from the joystick. Please see Jumper section for detailed description of the functionality.
Rear Panel Jumper and Switch Selection

Current Switch Settings – SW1 CUR X, SW2 CUR Y, SW3 CUR Z, SW4 CUR U

<table>
<thead>
<tr>
<th>DS4</th>
<th>DS3</th>
<th>DS2</th>
<th>DS1</th>
<th>Current (A)</th>
<th>DIP Switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.2</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3.0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2.8</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2.6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2.4</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2.2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1.8</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1.6</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.4</td>
<td></td>
</tr>
</tbody>
</table>

Table 16 Current Switch Settings

These DIP switches set the maximum current rating for the stepper motor. Please refer to your motor data sheet to set these DIP switches.
## Jumper Selection – JP1 – JP8

<table>
<thead>
<tr>
<th>Jumper</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JP1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1-2 for Output 8 to rear panel</td>
<td>Installing a jumper on JP1 1-2 will <strong>disable</strong> the SPDT Relay and Output 8 will be routed to J18</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2-3 for SPDT Relay</td>
<td>Installing a jumper on JP1 2-3 will <strong>enable</strong> the SPDT Relay and Output 8 will be used to switch the relay</td>
</tr>
<tr>
<td>JP2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1-2 for Input 8 to rear panel</td>
<td>Installing a jumper on JP2 1-2 will <strong>route</strong> Input 8 to J18</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2-3 to disable Input 8</td>
<td>Installing a jumper on JP2 2-3 will <strong>disable</strong> Input 8</td>
</tr>
<tr>
<td>JP3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1-2 for Input 7 to rear panel</td>
<td>Installing a jumper on JP3 1-2 will <strong>disable</strong> the front panel THUMBWHEEL button and Input 7 will be routed to J18</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2-3 for front panel thumbwheel</td>
<td>Installing a jumper on JP3 2-3 will <strong>enable</strong> the front panel THUMBWHEEL button and Input 7 will not be available on J18</td>
</tr>
<tr>
<td>JP4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1-2 for Input 6 to rear panel</td>
<td>Installing a jumper on JP4 1-2 will <strong>disable</strong> the front panel THUMBWHEEL button and Input 6 will be routed to J18</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2-3 for front panel thumbwheel</td>
<td>Installing a jumper on JP4 2-3 will <strong>enable</strong> the front panel THUMBWHEEL button and Input 6 will not be available on J18</td>
</tr>
<tr>
<td>JP5</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1-2 for Input 5 to rear panel</td>
<td>Installing a jumper on JP5 1-2 will <strong>disable</strong> the front panel THUMBWHEEL button and Input 5 will be routed to J18</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2-3 for front panel thumbwheel</td>
<td>Installing a jumper on JP5 2-3 will <strong>enable</strong> the front panel THUMBWHEEL button and Input 5 will not be available on J18</td>
</tr>
<tr>
<td>JP6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1-2 for Input 4 to rear panel</td>
<td>Installing a jumper on JP6 1-2 will <strong>disable</strong> the front panel THUMBWHEEL button and Input 4 will be routed to J18</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2-3 for front panel thumbwheel</td>
<td>Installing a jumper on JP6 2-3 will <strong>enable</strong> the front panel THUMBWHEEL button and Input 4 will not be available on J18</td>
</tr>
<tr>
<td>JP7</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1-2 for Input 3 to rear panel</td>
<td>Installing a jumper on JP7 1-2 will <strong>disable</strong> the front panel STOP button and Input 3 will be routed to J18 &amp; J19</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2-3 for front panel Stop button</td>
<td>Installing a jumper on JP7 2-3 will <strong>enable</strong> the front panel STOP button and Input 3 will not be available on J18 &amp; J19</td>
</tr>
<tr>
<td>JP8</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1-2 for Input 2 to rear panel</td>
<td>Installing a jumper on JP8 1-2 will <strong>disable</strong> the front panel START button and Input 2 will be routed to J18 &amp; J19</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2-3 for front panel Start button</td>
<td>Installing a jumper on JP8 2-3 will <strong>enable</strong> the front panel START button and Input 2 will not be available on J18 &amp; J19</td>
</tr>
</tbody>
</table>

Table 17: Jumper Selection – JP1 – JP8
Rear Panel LEDs

Fuse LEDs

<table>
<thead>
<tr>
<th>LED</th>
<th>Description</th>
<th>Possible States</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuse X</td>
<td>Indicates that the fuse on X axis is ok</td>
<td>Not illuminated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Illuminated green</td>
</tr>
<tr>
<td>Fuse Y</td>
<td>Indicates that the fuse on Y axis is ok</td>
<td>Not illuminated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Illuminated green</td>
</tr>
<tr>
<td>Fuse Z</td>
<td>Indicates that the fuse on Z axis is ok</td>
<td>Not illuminated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Illuminated green</td>
</tr>
<tr>
<td>Fuse U</td>
<td>Indicates that the fuse on U axis is ok</td>
<td>Not illuminated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Illuminated green</td>
</tr>
</tbody>
</table>

Table 18  Fuse LEDs

Enable LEDs

<table>
<thead>
<tr>
<th>LED</th>
<th>Description</th>
<th>Possible States</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENA X</td>
<td>Indicates that the axis is enabled</td>
<td>Not illuminated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Illuminated green</td>
</tr>
<tr>
<td>ENA Y</td>
<td>Indicates that the axis is enabled</td>
<td>Not illuminated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Illuminated green</td>
</tr>
<tr>
<td>ENA Z</td>
<td>Indicates that the axis is enabled</td>
<td>Not illuminated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Illuminated green</td>
</tr>
<tr>
<td>ENA U</td>
<td>Indicates that the axis is enabled</td>
<td>Not illuminated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Illuminated green</td>
</tr>
</tbody>
</table>

Table 19  Enable LEDs
Interface Circuitry

Limit, Home, and Digital Input

To trigger the opto-isolated digital inputs, sink the digital input signal to the ground of the corresponding opto-supply.

Note: Alarm input for TB9 version is 5V TTL type.

![Figure 5 Digital Inputs](image)

Digital Outputs

For the opto-isolated outputs, the digital output signal will source from VS when the signal is turned on.

![Figure 6 Digital Outputs](image)
Encoder Input Connection

Both single-ended and differential quadrature encoder inputs are accepted.

When using single-ended encoders, use the /A, /B, and /Z inputs.

+5V supply and Ground signals are available to power the encoder. Make sure that the total current usage is less than 200mA for the +5V.

The maximum encoder frequency is 5MHz.

![Encoder Inputs](image)

Figure 7 Encoder Inputs

Analog Inputs

8 x 10-bit analog inputs are available on the DMM-0400. Use AI[1-8] command to read the analog input value. Range is from 0-5000 mV.

![Analog Inputs](image)

Figure 8 Analog Inputs
Where to Go Next

Continue with the next chapter, *Chapter 3 - Installing the DMM-0400*. Now that user has become familiar with the technical capabilities of the DMM-0400, the next section will outline the steps needed to install the DMM-0400 into your system.
Chapter 3
Installing the DMM-0400

About This Chapter
Introduction

This chapter describes how to hook up the DMM-0400 to your Dover Motion stages.

Topics

This chapter covers the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing the System</td>
<td>37</td>
</tr>
<tr>
<td>Modes of Operation</td>
<td>37</td>
</tr>
<tr>
<td>Connecting to the DMM-0400</td>
<td>38</td>
</tr>
<tr>
<td>Where to Go Next</td>
<td>39</td>
</tr>
</tbody>
</table>
Installing the System

Figure 9  System Installation

Modes of Operation

1. **PC Controlled** – The DMM-0400 can be controlled using a PC and USB communications. Dover Motion provides a GUI for running and programming the DMM-0400. The software is called PMX-4EX-SA. See Chapter 5.

2. **Stand Alone Operation** – The DMM-0400 can be operated stand alone. No external communications are required. Motion can be initiated using the front panel START & STOP buttons and the PROGRAM SELECT thumbwheel. See Chapter 6.
Connecting to the DMM-0400

Installing the Motor and Encoder Cables

**Caution:** Before plugging in any motor and encoder cables, verify the mechanical operation of your Dover Motion stage. Verify that the stage moves freely as intended and that there is no potential for damage or personal injury.

**Note:** If you have any questions regarding the mechanical operation of your stage or the connecting to the DMM-0400, please call the Dover Motion customer service department:

Phone: (508) 475 – 3400
Email: sales@dovermotion.com

Plug in the motor and encoder cables to the appropriate rear panel connectors. Plug in your motor and encoder connections before plugging in the power cable.

Installing the I/O Cables

**Caution:** Before plugging in the I/O cables, verify that your I/O devices operate as intended and there is no potential for damage or personal injury.

Plug in the I/O cables to the appropriate rear panel connectors. Plug in your I/O connections before plugging in the power cable.

Installing the Power Cord and Communications Cables

The DMM-0400 comes with a 2.4 meter (approx. 8 foot) power cord. You must supply the USB 2.0 cable.

To install the power cord, insert the female connector on the power cord into the male receptacle on the rear of the system enclosure. The USB 2.0 cable plugs into the USB connector on the rear panel. See the Rear Panel section for the locations of these connectors.

**Note:** After plugging in the power cord, the DMM-0400 is ready to be turned on.
Where to Go Next

Continue with the next chapter, *Chapter 4 - Installing the Application Software for the DMM-0400*. After hooking up the DMM-0400 to the stages, the user should now be able to install the application software, **PMX-4EX-SA Software** and start communicating to the DMM-0400. The next section describes the steps needed to properly install the application software.

Note: Using the PMX-4EX-SA Software is not required in order to communicate to the DMM-0400. The user may use Labview, Matlab, or a number of other software to communicate to the DMM-0400 through USB connection.
Chapter 4
Installing Application Software for the DMM-0400

About This Chapter

Introduction

This chapter describes the Application GUI software you need to run the DMM-0400 via a terminal.

Topics

This chapter covers the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Description</td>
<td>41</td>
</tr>
<tr>
<td>Installing PMX-4EX-SA Software</td>
<td>41</td>
</tr>
<tr>
<td>Installing PMX Drivers</td>
<td>42</td>
</tr>
<tr>
<td>Where to Go Next</td>
<td>43</td>
</tr>
</tbody>
</table>
Software Description

The software that is used to communicate and to run the DMM-0400 is the PMX-4EX-SA software. This software provides the user with an intuitive graphical user interface for running and programming the DMM-0400. The user also needs to install the PMX drivers to complement the software.

Note: Using the PMX-4EX-SA Software is not required in order to communicate to the DMM-0400. The user may use Labview, Matlab, or a number of other software to communicate to the DMM-0400 through USB. See the USB Communications section for communication details and associated ASCII commands.

Installing PMX-4EX-SA Software

GUI (PMX-4EX-SA) Install

Locate the CD that was shipped with your DMM-0400. From your Windows folder double click the “Performax_Installation_1.54” and follow the on screen instructions. See Figure 10.
Installing PMX Drivers

Locate the CD that was shipped with your DMM-0400. From your Windows folder double click the “Arcus_Drivers_and_Tools_Setup_1.06” and follow the on screen instructions. See Figure 11.
Where to Go Next

Continue with the next chapter, *Chapter 5 - Running the DMM-0400 via PMX-4EX-SA Software*. This next section describes how to use the software that was installed in section 4. Details and functions about the various screens are described.
Chapter 5
Running the DMM-0400 via PMX-4EX-SA Software

About This Chapter

Introduction

This chapter describes how to communicate with the DMM-0400 via USB connection to the user's PC and how to use the PMX-4EX-SA software to control motion.

Topics

This chapter covers the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connecting a USB Cable</td>
<td>45</td>
</tr>
<tr>
<td>Establishing Communication with the DMM-0400</td>
<td>46</td>
</tr>
<tr>
<td>Using the Main Control Screen</td>
<td>47</td>
</tr>
<tr>
<td>Where to Go Next</td>
<td>59</td>
</tr>
</tbody>
</table>
Connecting a USB Cable

To communicate with the DMM-0400 via a terminal you will need a USB 2.0 cable not supplied with this DMM. See Table 3 for details of cable specifications.

Once you have located a USB cable, plug the “B” end into the DMM-0400 USB port. See Figure 12. Then plug the “A” end in any open USB port of your PC.

Figure 12  DMM-0400 USB Port
Establishing Communication with the DMM-0400

Locate the “SOFT-PMX-4EX-SA-131” icon on your desktop or locate it in your START menu.

Select the device you want to communicate with in the shown dialog box. Choose USB for USB communications for the DMM-0400. See Figure 14.
In the Select Control Box, the user can select one of 3 programs. Select the Program & Control button for the Main Control Screen.

Using the Main Control Screen

With the DMM-0400 GUI, you can control all motion via one "Main Control Screen", see Figure 16. For addition information not found in this manual reference the "PMX-4EX-SA Manual" which is included on the shipping CD and installed onto your PC with this GUI.
1. Current pulse position (X,Y,Z,U axes). If StepNLoop is enabled, this shows the real-time target position.
2. Current encoder position (X,Y,Z,U axes)
3. Current speed (X,Y,Z,U axes) pulse/sec. If StepNLoop is enabled, the speed is in encoder counts/sec, unless an interpolation move is in process.
4. Motor status (X,Y,Z,U axes)
   i. Idle – motor is not moving.
   ii. Accel – motor is accelerating
   iii. Const – motor is running in constant speed
   iv. Decel – motor is decelerating
   v. +LimError – plus limit error
   vi. -LimError – minus limit error
5. StepNLoop status - valid only when StepNLoop is enabled and displays current StepNLoop status by displaying one of the following:
   NA – StepNLoop is disabled
   IDLE – motor is not moving
   MOVING – target move is in progress
   JOGGING – jog move is in progress
   HOMING – homing is in progress
   Z-HOMING – homing using Z-index channel in progress
   ERR-STALL – StepNLoop has stalled.
   ERR-LIM – plus/minus limit error
6. StepNLoop delta status
7. +Limit, -Limit, Home and Alarm input status (X,Y,Z,U axes)
8. Move mode status
   i. ABS – absolute move
   ii. INC – incremental move
9. Timer register status (counts down)
10. Clears any limit or StepNLoop error
11. Buffer move enable status
12. Buffer start: This is the current index of the buffer. Note that the buffer is a 36 position ring buffer. (Used for buffer move mode only)
13. Buffer end: This is the current end of the buffer. Note that the buffer is a 36 position ring buffer. (Used for buffer move mode only)
14. Provides the available empty positions of the buffer (Used for buffer move mode only)
Control (B)

1. Global High speed, low speed, and acceleration. To give each axis individual speed parameters, enter HS[axis], LS[axis] and ACC[axis] commands via the command line.
2. Select X/Y/Z/U axis to control.
3. Target Position (X,Y,Z,U axes)
4. Enable – motor power is turned on or off by clicking on these circles (X,Y,Z,U axes)
5. Set absolute move mode
6. Set incremental move mode
7. RP - Reset pulse counter for the specified axis. Not allowed if StepNLoop is enabled.
8. RE - Reset encoder counter for the specified axis.
9. SP - Set pulse counter for the specified axis.
10. SE - Set encoder counter for the specified axis.
11. ISTOP – the motion is immediately stopped without deceleration.
12. RSTOP – the motion is stopped with deceleration.
13. Z+/Z-: Home the axis using only encoder index channel.
14. H+/H-: Home the axis at high speed using only the home sensor.
15. Buffer move Tool – Clicking on this button will provide the user with an interface to load buffer move commands to the PMX-4EX-SA.

Figure 18 Status Screen (B)

Figure 19 Buffered I Move
a. Buffer Array List – Enter the desired list of buffer commands here. Once the list is loaded, if the number of commands is greater than 36 (max buffer size), the program will automatically send the remaining commands to the PMX-4EX-SA as spaces clears up in the buffer.
b. Buffer enable – Enable/Disable buffer move mode
c. Load Move Array – Once the buffer array list is created, click here to load the array list to the program.
d. Start – Once the array has been loaded, click here to begin sending the buffered commands to the PMX-4EX-SA. Note that after the “START” button is clicked, the buffer commands will not begin to be sent to the PMX-4EX-SA until the Buffer I Move window is closed.
e. Abort – Stop sending buffer commands to the PMX-4EX-SA. Also disables buffer move mode.
f. Open/Save/New – Allows users to save/open or create new buffer array lists
g. I Accel – Enable/Disable buffered I move acceleration.

16. ZH+/ZH-: Home sensor and encoder index channel is used to home.
17. Arc/Circle Tool – Clicking on this button will provide the user with an interface to perform Arc/Circle XY moves.

![Figure 20 Arc/Circle Tool](image)

a. Perform Arc Move – Once the arc center/degree/move direction parameters are set, clicking on this button will begin the arc move.
b. Perform Circle Move – Once the circle center/move direction parameters are set, clicking on this button will begin the circle move.
   Note that after an arc or circle move is started, the position/speed values of the main control window will not begin to update until the above window is closed.
18. JOG+/JOG-: Jogs the motor in positive and negative direction.
19. HL+/HL-: Home the axis at high speed and low speed using only the home sensor.
20. L+/L-: Home the axis using the limit sensor.
21. ABS: Perform absolute move. If more than one axis is selected, an interpolated move will result.
22. DAT: Return to 0 position. If more than one axis is selected, an interpolated move will result.
On-The-Fly-Speed Control (C)

1. Select X/Y/Z/U axis.
2. Select destination speed of the axis.
3. Select the acceleration used during an on-the-fly speed change.
4. Select the SSPDM mode for the axis. See On-The-Fly Speed section for details.
5. Set the SSPDM mode for the axis.
6. Set on-the-fly speed change. Acceleration will be taken from the "Acc" field. Make sure that the SSPDM mode has been set before issuing the on-the-fly speed operation.

On-The-Fly-Position Control (D)

1. Set the new target position for the specified axis.
2. SP - Perform and on-the-fly position change.

Product Information (E)
Sync Outputs (F)

Figure 24 Sync Output Status (F)

   i. OFF
   ii. WAITING
   iii. TRIGGERED

Digital Input/Output (G)

Figure 25 Digital I/O Status (G)

1. Digital Input - DI1-DI8
2. Digital Outputs - DO1-DO8. To turn on/off a digital output, click on the corresponding circle.

Analog Inputs (H)

Figure 26 Analog Inputs (H)

1. Analog input status of AI1-AI8. Units are in mV.
Program File Control (I)

1. Open – Open standalone program
2. Save – Save standalone program
3. New – Clear the standalone program editor
4. View - View the compiled program

Standalone Program Editor (J)

1. Text Program – Text box for writing and editing a standalone program.
2. Opens a larger Program Editor window for easier programming.
3. Clear Code Space – Clear the code space on the PMX-4EX-SA.
Standalone Program Control (K)

1. Run – Standalone program is run.
2. Stop – Program is stopped.
3. Pause – Program that is running can be paused.
4. Cont – Program that is paused can be continued.
5. XThread - Open the Standalone Program Control for all standalone programs.
6. Index – Current line of low-level code that is being executed.
7. Status of standalone program:
   i. Idle – Program is not running.
   ii. Running – Program is running.
   iii. Paused – Program is paused.
   iv. Error – Program is in an error state.

Standalone Program Compile/Download/Upload (L)

1. Compile – Compile the standalone program
2. Download – Download the compiled program
3. Upload – Upload the standalone program from the controller
Setup (M)

1. Polarity/S-curve:
   a. Set home/alarm/dir polarity for X/Y/Z/U axes. Note that limit polarity is fixed either high/low for ALL axes.
   b. Set s-curve enable/disable for each axes.
   c. DOP - Set the digital output polarity
   d. DIP - Set the digital input polarity
   e. EOP - Set the enable output polarity
   f. SA Err - Set the return jump line for standalone error handling

2. Set StepNLoop parameters
3. Set joystick parameters
4. Communication settings:
   a. Set device name: [4EX00-4EX99]
   b. Set baud rate (used for RS-485 communication)

5. Bootup Parameters
   a. Auto Run - Click and perform store to flash to have the specified standalone program run on boot-up
   b. DOBOOT/EOBOOT - Set the digital and enable output configuration status on boot-up.

6. Miscellaneous Settings:
   a. IERR - Enable/disable the ignore limit/alarm error feature
   b. IACC - Enable/disable I move acceleration (used with buffered I commands)
   c. EDEC - Enable/disable unique deceleration.
7. Set Sync output parameters (Note that sync output parameters are not stored to flash memory)
8. Open/Save parameters to file.
9. Upload/Download parameters to and from RAM
10. Store parameters to flash memory

Terminal (N)

1. Response Box – Displays sent command as well as corresponding response
2. Device Name – Device name of the PMX-4EX-SA. In USB communication mode, the field is fixed. In RS-485 communication mode, this field can be modified so that the user can communicate with a specific device.
3. Command line – ASCII command line
4. Save – Save the current contents of the Response Box to file
Latches (O)

1. Enable Latch – Enable latch support for X/Y/Z/U axis
2. Latch Status:
   a. ON: Latch has been triggered
   b. OFF: Latch is disabled
   c. WAITING: Latch is enabled and is waiting to be triggered.
3. Position – Latched position value. Note that this is reset to 0 each time that the latch is enabled.
4. Encoder – Latched encoder value. Note that this is reset to 0 each time that the latch is enabled.
Variable Status (P)

1. Volatile Variables – Status of volatile variable V0-V49
3. Command line – Set variables using V[0-99]=[value] syntax

About (Q)

Displays the current Software and Firmware versions.
Where to Go Next

Continue with the next chapter, *Chapter 6 - Running the DMM-0400 via Front Panel Buttons*. The next section describes how to run the DMM-0400 in standalone mode using the front panel buttons to control motion. Descriptions of the default motion programs installed on the DMM-0400 are also described.
Chapter 6
Running the DMM-0400 via Front Panel Buttons

About This Chapter

Introduction

This chapter describes how to run the DMM-0400 via the Front Panel Thumb Wheel and Start/ Jog+, Stop/ Jog- buttons for standalone operation. The sample motion programs installed from the factory are also described in greater detail.

Topics

This chapter covers the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front Panel Buttons</td>
<td>61</td>
</tr>
<tr>
<td>DMM-0400 Stored programs</td>
<td>62</td>
</tr>
<tr>
<td>Executing Stored Programs</td>
<td>62</td>
</tr>
<tr>
<td>Where to Go Next</td>
<td>63</td>
</tr>
</tbody>
</table>
Front Panel Buttons

The DMM-0400 front panel is made up of a Thumb wheel, Start/ Jog+, and Stop/ Jog- buttons, see Figure 36. These buttons can be used to run stored programs in the DMM-0400s memory. No USB communication to the DMM-0400 is required for standalone operation. The DMM-0400 will function as a completely standalone motion controller.

Figure 36 DMM-0400 Front Panel
DMM-0400 Stored programs

The DMM-0400 is shipped with 7 standard programs that can be executed via the Thumbwheel and Start/Stop buttons. See Table 20.

**Note:** The programs shipped with the DMM-0400 are **SAMPLE** programs. These programs are not intended to be the customer’s final solution. They are standard Dover Motion programs that may not be exactly suitable for the customer’s application. The user should create their own motion programs for specific motion profiles and control.

<table>
<thead>
<tr>
<th>Program Number</th>
<th>Function</th>
<th>Pushing the START Button will…</th>
<th>Pushing the STOP Button will…</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Joystick Control</td>
<td>Enable external joystick control</td>
<td>Disable external joystick control</td>
<td>When the joystick control is enabled, starting another motion program will automatically disable the joystick and run the selected motion program. To enable the joystick again, the user must start program 0 again.</td>
</tr>
<tr>
<td>1</td>
<td>X Axis Jogging</td>
<td>Jog the X Axis positive</td>
<td>Jog the X Axis negative</td>
<td>Holding the START (or STOP) button will jog the axis. Releasing the START (or STOP) button will stop the jogging.</td>
</tr>
<tr>
<td>2</td>
<td>Y Axis Jogging</td>
<td>Jog the Y Axis positive</td>
<td>Jog the Y Axis negative</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Z Axis Jogging</td>
<td>Jog the Z Axis positive</td>
<td>Jog the Z Axis negative</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>U Axis Jogging</td>
<td>Jog the U Axis positive</td>
<td>Jog the U Axis negative</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>XYZU Homing</td>
<td>Homes all axis</td>
<td>Stop the homing routing</td>
<td>The homing program will home all axis sequentially one after another. The axis will jog to the negative limit, jog off the negative limit by 1 mm, and zero the position counter. The resulting position is the home or zero position.</td>
</tr>
<tr>
<td>6</td>
<td>XYZU Cycling</td>
<td>Cycles all axis</td>
<td>Stops the cycling</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Available for customer use</td>
<td>Start program 7</td>
<td>Stop program 7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Available for customer use</td>
<td>Start program 8</td>
<td>Stop program 8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Available for customer use</td>
<td>Start program 9</td>
<td>Stop program 9</td>
<td></td>
</tr>
</tbody>
</table>

Table 20  Standard Programs

**Executing Stored Programs**

To execute one of the stored programs, select that program number on the thumbwheel and then depress the **Start** button.

To stop the program press the **Stop** button.
Where to Go Next

Continue with the next chapter, *Chapter 7 - Editing Programs*. Using either front panel buttons or the application software to control the DMM-0400, the DMM-0400 calls motion programs to run the stages. The next section describes how to edit those programs.
Chapter 7
Editing Programs

About This Chapter

Introduction

This chapter describes how to edit motion programs with the GUI. This chapter also describes the programming language for programs in the DMM-0400

Topics

This chapter covers the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Editing a Stored Program</td>
<td>65</td>
</tr>
<tr>
<td>Standalone Program Specification</td>
<td>67</td>
</tr>
<tr>
<td>Programming Language Specification</td>
<td>69</td>
</tr>
<tr>
<td>Sample Programs</td>
<td>90</td>
</tr>
<tr>
<td>Where to Go Next</td>
<td>94</td>
</tr>
</tbody>
</table>
Editing a Stored Program

To edit a stored program with the GUI, you first need to open the program. You can either Upload the program from the controller’s memory or open a program saved on your PC. Within the GUI you can hit the Open or Upload button. See Figure 37.

Figure 37 Retrieving a Program
The program will now be in the Text window. The user can now edit the program as needed. After editing the programs, hit the Compile button and then the Download button to compile and download the new programs into the DMM-0400 memory. To run the new programs, hit the run button. See Figure 38.

**Figure 38 Text Window**

**Note:** When you upload the saved programs from the controller's memory, you will be uploading Dover Motion's standard motion programs described in the DMM-0400 Stored Programs section.

If you want to keep Dover Motion's standard programs, the user will have to edit the uploaded file (which contains Dover Motion's standard programs) to include the user's new motion programs.

If you do not want to keep Dover Motion's standard programs, Clear Code Space, create your own motion programs, and download to the controller's memory.
Standalone Program Specification

Standalone Program Specification:
Memory size: 1,275 assembly lines.
Note: Each line of pre-compiled code equates to 1-4 lines of assembly lines.

WAIT Statement
When writing a standalone program, it is generally necessary to wait until a motion is completed before moving on to the next line. In order to do this, the WAIT statement must be used. See the examples below:

In the example below, the variable V1 will be set immediately after the X10000 move command begins; it will not wait until the controller is idle.

```plaintext
X10000 ;* Move to position 0
V1 = 100
```

Conversely, in the example below, the variable V1 will not be set until the motion has been completed. V1 will only be set once the controller is idle.

```plaintext
X10000 ;* Move to position 0
WAITX ;* Wait for the move to complete
V1 = 100
```

Multi-Threaded
PMX-4EX-SA supports the simultaneous execution of up to 4 standalone programs. Programs 0,1,2,3 are controlled via the SR0, SR1, SR2 and SR3 commands respectively. For examples of multi-threading, please refer to the Example Stand-alone Programs section.

Note: Sub-routines can be shared by different threads.

Error Handling
If an error occurs during standalone execution (i.e. limit error), the program automatically jumps to SUB 31. If SUB 31 is NOT defined, the program will cease execution and go to error state. If SUB 31 is defined by the user, the code within SUB 31 will be executed. The return jump line will be determined by value of the SAP register. If the value is 0, the return jump line will be the line that caused the error. Otherwise, the return jump line will be line 0.

Calling subroutines over communication
Once a subroutine is written into the flash, they can be called via USB communication using the GS command. The subroutines are referenced by their subroutine number [0-31]. If a subroutine number is not defined, the controller will return with an error.

Standalone Run on Boot-Up
Standalone can be configured to run on boot-up using the SLOAD command. See description below:
<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Standalone Program 0</td>
</tr>
<tr>
<td>1</td>
<td>Standalone Program 1</td>
</tr>
<tr>
<td>2</td>
<td>Standalone Program 2</td>
</tr>
<tr>
<td>3</td>
<td>Standalone Program 3</td>
</tr>
</tbody>
</table>

Table 21 Standalone Run on Boot-Up

Timer Register
PMX-4EX-SA comes with a timer register. Once the timer register is set, it begins to count down to 0. Read and write to the timer register using the TR command. The units are in milliseconds.

Communication Time-out Feature (Watchdog)
PMX-4EX-SA allows for the user to trigger an alarm if the master has not communicated with the device for a set period of time. When an alarm is triggered bit 11 of the MSTX parameter is turned on. The time-out value is set by the TOC command. Units are in milliseconds. This feature is usually used in stand-alone mode. Refer to the Example Stand-alone Programs section for an example.

In order to disable this feature set TOC=0.

Storing to Flash
The following items are stored to flash:

<table>
<thead>
<tr>
<th>ASCII Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DN</td>
<td>Device name</td>
</tr>
<tr>
<td>DB</td>
<td>Serial communication baud rate</td>
</tr>
<tr>
<td>DOBOOT</td>
<td>DO configuration at boot-up</td>
</tr>
<tr>
<td>EDEC</td>
<td>Unique deceleration enable</td>
</tr>
<tr>
<td>EOB OOT</td>
<td>EO configuration at boot-up</td>
</tr>
<tr>
<td>IACC</td>
<td>Automatic I-move acceleration/deceleration enable</td>
</tr>
<tr>
<td>IERR</td>
<td>Ignore limit error enable</td>
</tr>
<tr>
<td>POL[axis]</td>
<td>Polarity settings</td>
</tr>
<tr>
<td>SAP</td>
<td>Jump return line select (Stand-alone error handling)</td>
</tr>
<tr>
<td>EOP</td>
<td>EO polarity</td>
</tr>
<tr>
<td>DIP</td>
<td>DI polarity</td>
</tr>
<tr>
<td>DOP</td>
<td>DO polarity</td>
</tr>
<tr>
<td>SCV[axis]</td>
<td>S-curve enable</td>
</tr>
<tr>
<td>SL[axis], SLR[axis], SLE [axis], SLT[axis]</td>
<td>StepNLoo parameters</td>
</tr>
<tr>
<td>JO, JF, JV[1-12], JL[1-]</td>
<td>Joystick settings</td>
</tr>
<tr>
<td>SLOAD</td>
<td>Standalone program run on boot-up parameter</td>
</tr>
<tr>
<td>TOC</td>
<td>Time-out counter reset value</td>
</tr>
<tr>
<td>V50-V99</td>
<td>Note that on boot-up, V0-V49 are reset to value 0</td>
</tr>
</tbody>
</table>

Table 22 Storing to Flash

Note: When a standalone program is downloaded, the program is immediately written to flash memory.
Programming Language Specification

; Description:
Comment notation. In programming, comment must be in its own line.
Syntax:
; [Comment Text]
Examples:
; ***This is a comment
  JOGX+ ;***Jogs X axis to positive direction
  DELAY= 1000 ;***Wait 1 second
  ABORT ;***Stop immediately all axes including X axis

ABORT
Description:
Motion: Immediately stops all axes if in motion without deceleration.
Syntax:
ABORT
Examples:
  JOGX+ ;***Jogs X axis to positive direction
  DELAY= 1000 ;***Wait 1 second
  ABORT ;***Stop immediately all axes including X axis

ABORT[axis]
Description:
Motion: Immediately stops individual axis without deceleration.
Syntax:
ABORT[axis]
Examples:
  JOGX+ ;***Jogs X axis to positive direction
  JOGY+ ;***Jogs Y axis to positive direction
  JOGZ+ ;***Jogs Z axis to positive direction

ABS
Description:
Motion: Changes all move commands to absolute mode.
Syntax:
ABS
Examples:
  ABS ;***Change to absolute mode
  PX=0 ;***Change X position to 0
  X1000 ;***Move X axis to position 1000
  WAITX
  X2000 ;***Move X axis to position 2000
  WAIT

ACC
Description:
Read: Get acceleration value
Write: Set acceleration value. Value is in milliseconds.
Syntax:
Read:  [variable] = ACC
Write:  ACC = [value]
       ACC = [variable]
Conditional:  IF ACC=[variable]
               ENDIF
            IF ACC=[value]
               ENDIF

Examples:
ACC=300 ;***Sets the acceleration to 300 milliseconds
V3=500  ;***Sets the variable 3 to 500
ACC=V3  ;***Sets the acceleration to variable 3 value of 500

**ACC[axis]**
Description:
Read:  Get individual acceleration value
Write:  Set individual acceleration value.
       Value is in milliseconds.
Syntax:
Read:  [variable] = ACC[axis]
Write:  ACC[axis] = [value]
       ACC[axis] = [variable]
Conditional:  IF ACC[axis]=[variable]
               ENDIF
            IF ACC[axis]=[value]
               ENDIF

Examples:
ACCX=300 ;***Sets the X acceleration to 300 milliseconds
V3=500   ;***Sets the variable 3 to 500
ACCX=V3  ;***Sets the X acceleration to variable 3 value of 500

**AI[1-8]**
Description:
Read:  Gets the analog input value. PMX-4EX-SA has 8 analog inputs. Range is from 0-5000 mV
Syntax:
Read:  [variable] = AI[1-8]
Conditional:  IF AI[1-8]=[variable]
               ENDIF
            IF AI[1-8]=[value]
               ENDIF

Examples:
IF AI1 < 500
   DO=1 ;***If analog input 1 is less than 500, set DO=1
   ENDIF

**ARC**
Description:
Motion: Perform arc move using X and Y axis.
       Specify clockwise or counter-clockwise, center location, and the absolute and relative angle. Angle is in whole number in thousandth. For example, 45 degrees is 45,000.
Syntax:
ARC[P for CW, N for CCW] [Center X][Center Y][AngleA][AngleR]
Examples:

;*** move 90 degrees CW to absolute angle 90 degrees
ARCP1000:0:90000:90000
WAITX
;*** move 180 degrees CCW to absolute angle 0 degrees
ARCN0: 1000:0:180000 WAITX

**CIR**
Description:

**Motion:** Perform circle move using X and Y axis. 
Specify clockwise or counter-clockwise and the center location.

**Syntax:**

CIR[P for CW, N for CCW] [Center X].[Center Y]

**Examples:**

CIRP1000:1000;***Using X 1000 and Y 1000 perform circular move (CW) WAITX
CIRN0:2000;***Using X 0 and Y 2000 perform circular move (CCW) WAITX

**DEC**
Description:

**Read:** Get deceleration value
**Write:** Set deceleration value. Value is in milliseconds.

**Syntax:**

Read: [variable] = DEC
Write: DEC = [value]  
DEC = [variable]

**Examples:**

DEC=300;***Sets the deceleration to 300 milliseconds
V3=500;***Sets the variable 3 to 500
DEC=V3;***Sets the deceleration to variable 3 value of 500

**DEC[axis]**
Description:

**Read:** Get individual deceleration value
**Write:** Set individual deceleration value. Value is in milliseconds.

**Syntax:**

Read: [variable] = DEC[axis]
Write: DEC[axis] = [value]  
DEC[axis] = [variable]

**Conditional:**

IF ACC[axis]=[variable]
ENDIF
IF ACC[axis]=[value]
ENDIF

**Examples:**

DECX=300;***Sets the X deceleration to 300 milliseconds
V3=500;***Sets the variable 3 to 500
DECX=V3;***Sets the X deceleration to variable 3 value of 500
**DELAY**

Description:
Set a delay (1 ms units)

Syntax:

```
Delay=[Number] (1 ms units) Examples:
JOGX+ ;***Jogs X axis to positive direction
DELAY=10000 ;***Wait 10 second
ABORT ;***Stop with deceleration all axes including X axis
EX=0 ;***Sets the current X encoder position to 0
EY=0 ;***Sets the current Y encoder position to 0
EZ=0 ;***Sets the current Z encoder position to 0
EU=0 ;***Sets the current U encoder position to 0
```

**DI**

Description:
```
Read: Gets the digital input value
Performax 4EX has 8 digital inputs
```

Syntax:

```
Read: [variable] = DI
Conditional: IF DI=[variable]
            ENDIF
            IF DI=[value]
            ENDIF
```

Examples:
```
IF DI=255
    DO=1 ; * * *If no digital inputs are triggered, set DO= 1
ENDIF
```

**DI[1-8]**

Description:
```
Read: Gets the digital input value
Performax 4EX has 8 digital inputs
```

Syntax:

```
Read: [variable] = DI[1-8]
Conditional: IF DI[1-8]=[variable]
            ENDIF
            IF DI[1-8]=[0 or 1]
            ENDIF
```

Examples:
```
IF DI1=1
    DO=1 ; * * *If digital input 1 is triggered, set DO= 1
ENDIF
```

**DO**

Description:
```
Read: Gets the digital output value
Write: Sets the digital output value
Performax 4EX has 8 digital outputs
```

Syntax:

```
Read: [variable] = DO
Write: DO = [value]
```
DO = [variable]

Conditional: IF DO=[variable]
ENDIF
IF DO=[value]
ENDIF

Examples:
DO=7 ;***Turn first 3 bits on and rest off

**DO[1-8]**

Description:
Read: Gets the individual digital output value
Write: Sets the individual digital output value Performax 4EX has 8 digital outputs

Syntax:
Read: [variable] = DO[1-8]
Write: DO[1-8] = [0 or 1]
DO[1-8] = [variable]
Conditional: IF DO[1-8]=[variable]
ENDIF
IF DO[1-8]=[0 or 1]
ENDIF

Examples:
DO7=1 ;***Turn DO7 on
DO6=1 ;***Turn DO6 on

**E[axis]**

Description:
Read: Gets the current encoder position
Write: Sets the current encoder position

Syntax:
Read: [variable] = E[axis]
Write: E[axis] = [value]
E[axis] = [variable]
Conditional: IF E[axis]=[variable]
ENDIF
IF E[axis]=[value]
ENDIF

Examples:
JOGX+ ;***Jogs X axis to positive direction
DELAY= 1000 ;***Wait 1 second
ABORT ;***Stop with deceleration all axes including X axis
EX=0 ;***Sets the current X encoder position to 0
EY=0 ;***Sets the current Y encoder position to 0
EZ=0 ;***Sets the current Z encoder position to 0
EU=0 ;***Sets the current U encoder position to 0

**ECLEAR[axis]**

Description:
Write: Clears error status. Also clears StepNLoop error.

Syntax:
Write: ECLEAR[axis]

Examples:
ECLEARX ;***Clears error of axis X
ECLEARY ;***Clears error of axis Y
ECLEARZ ;***Clears error of axis Z
ECLEARU ;***Clears error of axis U

**ELSE**

Description: Perform ELSE condition check as a part of IF statement

Syntax: ELSE

Examples:

IF V1=1
    X1000 ;***If V1 is 1, then move to 1000
    WAITX
ELSE
    X-1000 ;***If V1 is not 1, then move to -1000
    WAITX
ENDIF

**ELSEIF**

Description: Perform ELSEIF condition check as a part of the IF statement

Syntax: ELSEIF [Argument 1] [Comparison] [Argument 2]

[Argument] can be any of the following:
Numerical value
Pulse or Encoder Position
Digital Output
Digital Input
Enable Output
Motor Status

[Comparison] can be any of the following
=   Equal to
>   Greater than
<   Less than
>=  Greater than or equal to
<=  Less than or equal to
!=  Not Equal to

Examples:

IF V1=1
    X1000
    WAITX
ELSEIF V1=2
    X2000
    WAITX
ELSEIF V1=3
    X3000
    WAITX
ELSE
    X0
    WAITX
ENDIF
**END**

Description:
Indicate end of program.
Program status changes to idle when END is reached.

*Note:* Subroutine definitions should be written AFTER the END statement

Syntax:
```plaintext
END
```

Examples:
```plaintext
X0
WAITX X1000
WAITX END
```

**ENDIF**

Description:
Indicates end of IF operation

Syntax:
```plaintext
ENDIF Examples:
IF V1=1
    X1000
    WAITX
ENDIF
```

**ENDSUB**

Description:
Indicates end of subroutine
When ENDSUB is reached, the program returns to the previously called subroutine.

*Note:* Subroutine 31 is reserved for error handling

Syntax:
```plaintext
ENDSUB
```

Examples:
```plaintext
GOSUB 1
END
SUB 1
    X0
    WAITX
    X1000
    WAITX ENDSUB
```

**END WHILE**

Description:
Indicate end of WHILE loop

Syntax:
```plaintext
ENDWHILE
```

Examples:
```plaintext
WHILE V1=1 ;***While V1 is 1 continue to loop
    X0
    WAITX X1000
    WAITX
ENDWHILE ;***End of while loop so go back to WHILE
```
**EO**

Description:
- **Read:** Gets the enable output value
- **Write:** Sets the enable output value

Performax 4EX has 4 enable outputs.

Syntax:
- **Read:** [variable] = EO
- **Write:** EO = [value]
- **Conditional:** IF EO= [variable] ENDIF
  IF EO= [value] ENDIF

Examples:
- EO=3 ;***Turn first 2 bits of enable outputs
- IF V1=1
  - EO=V2 ;***Enable output according to variable 2
  - ENDIF

**EO[1-4]**

Description:
- **Read:** Gets the individual enable output value
- **Write:** Sets the individual enable output value

Performax 4EX has 4 enable outputs.

Syntax:
- **Read:** [variable] = EO[1-4]
- **Write:** EO[1-4] = [0 or 1]
  - EO[1-4] = [variable] **Conditional:** IF
  - EO= [variable] ENDIF
  - IF EO= [value] ENDIF

Examples:
- EO1=31 ;***Turn enable output 1 on
- IF V1=1
  - EO2=V2 ;***Enable output 2 according to variable 2
  - ENDIF

**GOSUB**

Description:
- Perform go to subroutine operation
- Subroutine range is from 0 to 31.

**Note:** Subroutine definitions should be written AFTER the END statement Subroutine 31 is reserved for error handling

Syntax:
- GOSUB [subroutine number]
  - [Subroutine Number] range is 0 to 31

Examples:
- GOSUB 1
- END
- SUB 1
  - X0
WAITX X1000
WAITX ENDSUB

**HLHOME[axis][+ or -]**

Description:
- **Command**: Perform low speed homing using current high speed, low speed, and acceleration.

Syntax:
- HLHOME[Aaxis][+ or -]

Examples:
- HLHOMEX+ ;***Low speed homes X axis in positive direction
- WAITX
- HLHOMEZ- ;***Low speed homes Z axis in negative direction
- WAITZ

**HOME[axis][+ or -]**

Description:
- **Command**: Perform homing using current high speed, low speed, and acceleration.

Syntax:
- HOME[Aaxis][+ or -]

Examples:
- HOMEX+ ;***Homes X axis in positive direction
- WAITX
- HOMEZ- ;***Homes Z axis in negative direction

**HSPD**

Description:
- **Read**: Gets high speed. Value is in pulses/second
- **Write**: Sets high speed. Value is in pulses/second.
  - Range is from 1 to 6M

Syntax:
- **Read**: [variable] = HSPD
- **Write**: HSPD = [value]
  - HSPD = [variable]
  - IF HSPD = [variable]
    - ENDIF
  - IF HSPD = [value]
    - ENDIF

Examples:
- HSPD=10000 ;***Sets the high speed to 10,000 pulses/sec
- V 1=2500 ;***Sets the variable 1 to 2,500
- HSPD=V1 ;***Sets the high speed to variable 1 value of 2500

**HSPD[axis]**

Description:
- **Read**: Gets individual high speed. Value is in pulses/second
- **Write**: Sets individual high speed. Value is in pulses/second.
  - Range is from 1 to 6M

Syntax:
- **Read**: [variable] = HSPD[axis]
- **Write**: HSPD[axis] = [value]
HSPD[\text{axis}] = \text{variable}

\textbf{Conditional:}

\begin{align*}
\text{IF} \ HSPD[\text{axis}] &= \text{variable} \\
\text{ENDIF} \\
\text{IF} \ HSPD[\text{axis}] &= \text{value} \\
\text{ENDIF}
\end{align*}

\textbf{Examples:}

\begin{align*}
\text{HSPDY} &= 10000 ;*** \text{Sets the Y high speed to 10,000 pulses/sec} \\
V1 &= 2500 ;*** \text{Sets the variable 1 to 2,500} \\
\text{HSPDY} &= V1 ;*** \text{Sets the Y high speed to variable 1 value of 250}
\end{align*}

\textbf{IF}

\textbf{Description:}

Perform IF condition check

\textbf{Syntax:}

\begin{align*}
\text{IF} \ [\text{Argument 1}] \ [\text{Comparison}] \ [\text{Argument 2}] \\
[\text{Argument}] \text{ can be any of the following:} \\
\text{Numerical value} \\
\text{Pulse or Encoder Position} \\
\text{Digital Output Digital Input} \\
\text{Enable Output Motor Status} \\
[\text{Comparison}] \text{ can be any of the following} \\
= \text{ Equal to} \\
> \text{ Greater than} \\
< \text{ Less than} \\
\geq \text{ Greater than or equal to} \\
\leq \text{ Less than or equal to} \\
\neq \text{ Not Equal to}
\end{align*}

\textbf{Examples:}

\begin{align*}
\text{IF} \ V1 &= 1 \\
X1000 \\
\text{WAITX} \\
\text{ENDIF}
\end{align*}

\textbf{INC}

\textbf{Description:}

\textbf{Command:} Changes all move commands to incremental mode.

\textbf{Syntax:}

\begin{align*}
\text{INC} \\
\text{Examples:}
\end{align*}

\begin{align*}
\text{INC} ;*** \text{Change to increment mode} \\
PX &= 0 ;*** \text{Change X position to 0} \\
X1000 ;*** \text{Move X axis to position 1000 (0+1000)} \\
\text{WAITX} \\
X2000 ;*** \text{Move X axis to position 3000 (1000+2000)} \\
\text{WAITX} \\
\text{ABORT} ;*** \text{Stop immediately all axes including X axis}
\end{align*}

\textbf{JOG[\text{axis}]}

\textbf{Description:}

\textbf{Command:} Perform jogging using current high speed, low speed, and acceleration.

\textbf{Syntax:}
JOG [Axis] [+ or -]
Examples:
JOGX+ ;*** Jogs X axis in positive direction
JOGY- ;*** Jogs Y axis in negative direction

JOYENA
Description:
Write: Enable joystick feature for axis
Range is from 0 or 15
Syntax:
Write: JOYENA=[value]
Examples:
JOYENA=1 ;*** Enable joystick feature on X axis only
JOYENA=3 ;*** Enable joystick feature on X and Y axis

JOYHS[axis]
Description:
Write: Set high speed setting for joystick control
Syntax:
Write: JOYHS [axis] = [value]
JOYHS [axis] = [variable]
Examples:
JOYHSX= 10000 ;*** High speed of X axis is set to 10,000 pps
JOYHSU=20000 ;*** High speed of U axis is set to 20,000 pps

JOYDEL[axis]
Description:
Write: Set maximum delta value of change in speed for joystick control
Syntax:
Write: JOYDEL[axis] = [value]
JOYDELaxis = [variable]
Examples:
JOYDELX= 100 ;*** Speed delta of X axis is set to 100 pps
JOYDELU=200 ;*** Speed delta of Y axis is set to 200 pps

JOYNO[axis]
Description:
Write: Set negative outer limit for joystick control
Syntax:
Write: JOYNO[axis] = [value]
JOYNO[axis] = [variable]
Examples:
JOYNOX= 10000 ;*** Negative outer limit of X-axis set to -10000
JOYNIX=9000 ;*** Negative inner limit of X-axis set to -9000
JOYPIX=9000 ;*** Positive inner limit of X-axis set to 9000
JOYPOX= 10000 ;*** Positive outer limit of X-axis set to 10000

JOYNi[axis]
Description:
**Write:** Set negative inner limit for joystick control

Syntax:

```
Write: JOYNi[axis] = [value]
       JOYNi[axis] = [variable]
```

Examples:

See JOYN0[axis]

---

**JOYPi[axis]**

Description:

**Write:** Set positive inner limit for joystick control

Syntax:

```
Write: JOYPi[axis] = [value]
       JOYPi[axis] = [variable]
```

Examples:

See JOYN0[axis]

---

**JOYPO[axis]**

Description:

**Write:** Set positive outer limit for joystick control

Syntax:

```
Write: JOYPO[axis] = [value]
       JOYPO[axis] = [variable]
```

Examples:

See JOYN0[axis]

---

**LHOME[axis][+ or -]**

Description:

**Command:** Perform limit homing using current high speed, low speed, and acceleration.

Syntax:

```
LHOME[Axis][+ or -]
```

Examples:

```
LHOMEX+ ;***Limit homes X axis in positive direction
WAITX
LHOMEZ- ;***Limit homes Z axis in negative direction
```

---

**LSPD**

Description:

**Read:** Get low speed. Value is in pulses/second.

**Write:** Set low speed. Value is in pulses/second. Range is from 1 to 6M

Syntax:

```
Read: [variable] = LSPD
Write: LSPD = [long value]
       LSPD = [variable]

Conditional: IF LSPD = [variable]
              ENDIF
              IF LSPD = [value]
              ENDIF
```

Examples:

```
LSPD = 1000 ;***Sets the start low speed to 1,000 pulses/sec
V 1 = 500    ;***Sets the variable 1 to 500
```
LSPD[V1] ;***Sets the start low speed to variable 1 value of 500

**LSPD[axis]**

Description:

- **Read:** Get individual low speed. Value is in pulses/second.
- **Write:** Set individual low speed. Value is in pulses/second. Range is from 1 to 6M

Syntax:

- **Read:** `[variable] =LSPD[axis]`
- **Write:** `LSPD[axis] = [long value]`
- `LSPD[axis] = [variable]`
- **Conditional:**
  - `IF LSPD[axis] = [variable] ENDIF`
  - `IF LSPD[axis] = [value] ENDIF`

Examples:

- `LSPDZ = 1000` ;***Sets the Z low speed to 1,000 pulses/sec
- `V1 = 500` ;***Sets the variable 1 to 500
- `LSPDZ = V1` ;***Sets the Z low speed to variable 1 value of 500

**MST[axis]**

Description:

- **Command:** Get motor status of axis

Syntax:

- `MST[Axis]`

Examples:

- `IF MSTX = 0 DIO = 6 ELSEIF MSTY = 0 DIO = 3 ELSEIF MSTZ = 0 DIO = 2 ELSEIF MSTU = 0 DIO = 1 ENDIF`

**P[axis]**

Description:

- **Read:** Gets the current pulse position
- **Write:** Sets the current pulse position

Syntax:

- **Read:** `Variable = P[axis]`
- **Write:** `P[axis] = [value]`
- `P[axis] = [variable]`
- **Conditional:**
  - `IF P[axis] = [variable] ENDIF`
  - `IF P[axis] = [value] ENDIF`

Examples:

- `JOGX+` ;***Jogs X axis to positive direction
- `DELAY = 1000` ;***Wait 1 second
- `ABORT` ;***Stop without deceleration all axes including X axis
PX=0 ;***Sets the current pulse position to 0

**PRG**

*Description:*
Indicates the start of a program
When END is reached, the program is concluded

*Syntax:*
PRG [program number] Examples:
PRG 0 ;***Program 0
X8000
END
PRG 1 ;***Program 1
Y1000
END

**PS[axis]**

*Description:*
Read: Get the current pulse position of an axis

*Syntax:*
Read: Variable = PS[Axis]
Conditional: IF PS[axis]=[variable]
ENDIF
IF PS[axis]=[value]
ENDIF

*Examples:*
JOGX+ ;***Jogs X axis to positive direction
DELAY= 1000 ;***Wait 1 second
ABORT ;***Stop without deceleration all axes including X axis
V1=PSX ;***Sets variable 1 to pulse X
JOGY+ ;***Jogs Y axis to positive direction
V2=PSY ;***Sets variable 2 to pulse Y

**SCV[axis]**

*Description:*
Read: Get individual s-curve enable. Value is 0 or 1.
Write: Set individual s-curve enable.
Range is from 0 or 1

*Syntax:*
Read: [variable]=SCV[axis]
Write: SCV[axis]=[0 or 1]
SCV[axis]=[variable]

*Note: If s-curve is enabled for an axis, on-the-fly speed feature cannot be used for the corresponding axis.*

*Examples:*
SCVX=1 ;***Sets X axis to use s-curve acceleration: on-the-fly speed
; change is NOT allowed for this axis.
SCVY=0 ;***Sets Y axis to use s-curve acceleration: on-the-fly speed
; change is allowed for this axis.
SCVZ=1 ;***Sets Z axis to use s-curve acceleration: on-the-fly speed
; change is NOT allowed for this axis.
SCVU=0 ;***Sets U axis to use s-curve acceleration: on-the-fly speed
; change is allowed for this axis.

**SL[axis]**
Description: Write: Set individual StepNLoop enable. Range is from 0 or 1
Syntax: Write: SL[axis]=[0 or 1]
Examples:
SLX=1 ;***Enables StepNLoop control for the X axis.
SLY=0 ;***Disables StepNLoop control for the Y axis.

**SLS[axis]**
Description: Command: Get the StepNLoop status of axis
Syntax:
SLS [Axis]
V[Value] = SLS [Axis]
Examples:
IF SLSX=0
    DIO=6
ELSEIF SLSY=0
    DIO=3
ENDIF

**SR[0,3]**
Description: Write: Set the standalone control for the specified standalone program
Syntax: Write: SR[0-3] = [0-3]
Examples:
IF DI1=1 ; If digital input 1 is on
    SR0=0 ; Turn off standalone program 0
ENDIF

**SSPD[axis]**
Description: Write: Set on-the-fly speed change for an individual axis.
Range is from 1 to 6,000,000 PPS
Syntax:
Write: SSPD[axis]=[value]
SSPD[axis]=[variable]
Note: If s-curve is enabled for an axis, on-the-fly speed feature cannot be used for the corresponding axis.
Examples:
SCVX=0 ;***Disable s-curve acceleration for X-axis
HSPDX= 1000 ;***X-axis high speed
LSPDX=100 ;***Set X-axis low speed
ACX= 100 ;***Set X-axis acceleration
JOGX+ ;***Jogs X axis to positive direction
DELAY= 1000 ;***Wait 1 second
SSPDX=3000 ;***Change speed on X-axis on-the-fly to 3000 PPS
**SSPDM[axis]**

**Description:**
Set individual on-the-fly speed change mode. Range is from 0 to 7.

**Syntax:**
- **Write:** SSPDM[axis]=0-7
- SSPDM[axis]=[variable]

**Examples:**
- SCVX=0;***Disable s-curve acceleration for X-axis
- HSPDX=1000;***X-axis high speed
- LSPDX=100;***Set X-axis low speed
- ACCX=100;***Set X-axis acceleration
- JOGX+;***Jogs X axis to positive direction
- DELAY=1000;***Wait 1 second
- SSPDMX=1;***Set on-the-fly speed change mode to 1
- ACCX=20000;***Set acceleration to 20 seconds
- SSPDX=190000;***Change speed on X-axis on-the-fly to 190000 PPS

**STOP**

**Description:**

**Command:** Stop all axes if in motion with deceleration. Previous acceleration value is used for deceleration.

**Syntax:**
- STOP

**Examples:**
- JOGX+;***Jogs X axis to positive direction
- DELAY=1000;***Wait 1 second
- STOP;***Stop with deceleration all axes including X axis

**STOP[axis]**

**Description:**
Stop individual axis if in motion with deceleration. Previous acceleration value is used for deceleration.

**Syntax:**
- STOP[axis]

**Examples:**
- JOGX+;***Jogs X axis to positive direction
- DELAY=1000;***Wait 1 second
- JOGY+;***Jogs Y axis to positive direction
- DELAY=1000;***Wait 1 second
- STOPX;***Stop with deceleration X axis only

**STORE**

**Description:**

**Command:** Store all values to flash

**Syntax:**
- STORE

**Examples:**
- V80=EX;***Put encoder value in V80
- DELAY=1000;***Wait 1 second
- STORE;***Store V80 to non-volatile flash
**SUB**

Description:
Indicates start of subroutine

Note: Subroutine definitions should be written AFTER the END statement Subroutine 31 is reserved for error handling

Syntax:
```
SUB [subroutine number]
```

[Subroutine Number] range is 0 to 31

Examples:
```
GOSUB 1
END
SUB 1
```
```
X0
WAITX X1000
WAITX ENDSUB
```

**SYNCFG[axis]**

Description:
Write: Set sync output configuration for axis

Syntax:
```
Write: SYNCFG[axis]=[value]
SYNCFG[axis]=[variable]
```

Examples:
```
SYNCFGX=1 ;*** Set sync output configuration to 1 for x-axis
SYNPOSX=3000 ;*** Set sync output position to 3000 for x-axis
SYNTIMEX=10 ;*** Set sync output pulse time to 10 ms for x-axis
SYNONX ;*** Turn on sync output for x-axis
V1=1 ;*** Wait until sync output is triggered for x-axis
WHILE V1 != 2
  V1 =SYNSTATX
ENDWHILE
SYNOFFX ;*** Disable sync output for x-axis
```

**SYNOFF[axis]**

Description:
Write: Disable sync output for axis

Syntax:
```
Write: SYNOFF[axis]
```

Examples:
```
See SYNCFG[axis]
```

**SYNON[axis]**

Description:
Write: Enable sync output for axis

Syntax:
```
Write: SYNON[axis]
```

Examples:
```
See SYNCFG[axis]
```
**SYNPOS[axis]**
Description: Write: Set sync output position for axis. 28-bit signed number
Syntax:
Write: SYNPOS [axis]=[value]
Write: SYNPOS [axis]=variable
Examples: See SYNCFG[axis]

**SYNSTAT[axis]**
Description: Read: Get status for sync output of axis
Syntax:
Read: variable = SYN[axis]S
Examples: See SYNCFG[axis]

**SYNTIME[axis]**
Description: Write: Set pulse output width time for sync output of axis
Syntax:
Write: SYNTIME[axis]=value
Examples: See SYNCFG[axis]

**TOC**
Description: Sets the communication time-out parameter. Value is in milli-seconds.
Syntax:
TOC=[long value]
Examples: TOC=10000 ;***Sets time-out parameter to 10 seconds

**TR**
Description: Read: Get count status of timer register
Write: Set timer register

Once TR is set, it begins to count down to 0. Units ms.
Syntax:
Read: [variable] =TR
Write: TR=[value]
Conditional: IF TR= [variable]
ENDIF
Examples:
TR= 1000
WHILE 1=1
IF TR>8000
X0
WAITX ELSEIF TR>5000
X3000
WAITX
ELSE
  X8000
  WAITX
ENDIF
ENDWHILE

**U**

Description:

**Command:** Perform U axis move to target location
With other Axis moves in the same line, linear interpolation move is done.

Syntax:

U[value]
U[variable]

Examples:

U10000 ;***Move U Axis to position 10000
WAITU
V10 = 1200 ;***Set variable 10 value to 1200
UV10 ;***Move U Axis to variable 10 value
WAITU

**V[index]**

Description:

Assign to variable.
Performax 4EX has 100 variables [V0-V99]

Syntax:

V[Variable Number] = [Argument]
V[Variable Number] = [Argument 1] [Operation] [Argument2]

Special case for BIT NOT:

V[Variable Number] = ~[Argument]

[Argument] can be any of the following:

- Numerical value
- Pulse or Encoder Position
- Digital Output
- Digital Input
- Enable Output
- Motor Status

[Operation] can be any of the following:

- Addition
- Subtraction
- Multiplication
- Division
- Modulus
- Bit Shift Right
- Bit Shift Left
- Bit AND
- Bit OR
- Bit NOT

Examples:

V1=12345 ;***Set Variable 1 to 123
V2=V1+1 ;***Set Variable 2 to V1 plus 1
V3=DI ;***Set Variable 3 to digital input value
V5=¬EO ;***Sets Variable 5 to bit NOT of enable output value
**WAIT[axis]**

Description:
Tell program to wait until move on the certain axis is finished before executing next line.

Syntax:
- `WAIT[axis]`
- `X[variable]`

Examples:
- `X10000
  WAITX
  DO=5
  Y3000
  WAITY`  
  /// Move X Axis to position 10000
  /// Wait until X Axis move is done
  /// Set digital output
  /// Move Y Axis to 3000
  /// Wait until Y Axis move is done

**WHILE**

Description:
Perform WHILE loop

Syntax:
- `WHILE [Argument 1] [Comparison] [Argument 2]`
  - `[Argument]` can be any of the following:
    - Numerical value
    - Pulse or Encoder Position
    - Digital Output Digital Input
    - Enable Output Motor Status
  - `[Comparison]` can be any of the following:
    - `=` Equal to
    - `>` Greater than
    - `<` Less than
    - `>=` Greater than or equal to
    - `<=` Less than or equal to
    - `!=` Not Equal to

Examples:
- `WHILE V1=1
  X0
  WAITX X1000
  ENDWHILE`

**X**

Description:
**Command:** Perform X axis move to target location
With other Axis moves in the same line, linear interpolation move is done.

Syntax:
- `X[value] X[variable]`

Examples:
- `X10000
  WAITX
  X2000Y3000
  V10 = 1200
  XV10
  WAITX`  
  /// Move X Axis to position 10000
  /// Move X to 2000 and Y to 3000 in linear interpolation move
  /// Set variable 10 value to 1200
  /// Move X Axis to variable 10 value
Y
Description:
Command: Perform Y axis move to target location
With other Axis moves in the same line, linear interpolation move is done.
Syntax:
Y[variable] Y[value]
Examples:
Y10000 ;***Move Y Axis to position 10000
WAITY
Y2000Z3000 ;***Move Y to 2000 and Z to 3000 in linear interpolation move WAITY
V10 = 1200 ;***Set variable 10 value to 1200
YV10 ;***Move Y Axis to variable 10 value
WAITY

Z
Description:
Command: Perform Z axis move to target location
With other Axis moves in the same line, linear interpolation move is done.
Syntax:
Z[variable] Z[value]
Examples:
Z10000 ;***Move X Axis to position 10000
WAITZ
Y1000Z2000U3000 ;***Move Y to 1000, Z to 2000, U to 3000
WAITY
V10 = 1200 ;***Set variable 10 value to 1200
ZV10 ;***Move Z Axis to variable 10 value
WAITZ

ZHOME[axis][+ or -]
Description:
Command: Perform Z-homing using current high speed, low speed, and acceleration.
Syntax:
ZHOME[axis][+ or -]
Examples:
ZHOMEX+ ;***Z Homes X axis in positive direction
ZHOMEZ- ;***Z Homes Z axis in negative direction

ZOME[axis][+ or -]
Description:
Command: Perform Zoning using current high speed, low speed, and acceleration.
Syntax:
ZOME[axis][+ or -]
Examples:
ZOMEX+ ;***Homes X axis in positive direction
ZOMEZ- ;***Homes Z axis in negative direction
Sample Programs

Standalone Example Program 1 – Single Thread
Task: Set the high speed and low speed and move the motor to 1000 and back to 0.

HSPD=20000 ;* Set the high speed to 20000 pulses/sec
LSPD= 1000 ;* Set the low speed to 1000 pulses/sec
ACC=300 ;* Set the acceleration to 300 msec
EO=1 ;* Enable the motor power
X1000 ;* Move to 1000
WAITX ;*Wait for X-axis move to complete
X0 ;* Move to zero
WAITX ;*Wait for X-axis move to complete
END ;* End of the program

Standalone Example Program 2 – Single Thread
Task: Move the motor back and forth indefinitely between position 1000 and 0.

HSPD=20000 ;* Set the high speed to 20000 pulses/sec
LSPD= 1000 ;* Set the low speed to 1000 pulses/sec
ACC=300 ;* Set the acceleration to 300 msec
EO=1 ;* Enable the motor power
WHILE 1=1 ;* Forever loop
  X0 ;* Move to zero
  WAITX ;*Wait for X-axis move to complete
  X1000 ;* Move to 1000
  WAITX ;*Wait for X-axis move to complete
ENDWHILE ;* Go back to WHILE statement
END

Standalone Example Program 3 – Single Thread
Task: Move the motor back and forth 10 times between position 1000 and 0.

HSPD=20000 ;* Set the high speed to 20000 pulses/sec
LSPD= 1000 ;* Set the low speed to 1000 pulses/sec
ACC=300 ;* Set the acceleration to 300 msec
EO=1 ;* Enable the motor power
V1=0 ;* Set variable 1 to value 0
WHILE V1<10 ;* Loop while variable 1 is less than 10
  X0 ;* Move to zero
  WAITX ;*Wait for X-axis move to complete
  X1000 ;* Move to 1000
  WAITX ;*Wait for X-axis move to complete
  V1=V1+1 ;* Increment variable 1
ENDWHILE ;* Go back to WHILE statement
**Standalone Example Program 4 – Single Thread**

Task: Move the motor back and forth between position 1000 and 0 only if the digital input 1 is turned on.

```plaintext
HSPD=20000 ;* Set the high speed to 20000 pulses/sec
LSPD= 1000 ;* Set the low speed to 1000 pulses/sec
ACC=300 ;* Set the acceleration to 300 msec
EO=1 ;* Enable the motor power
WHILE 1=1 ;* Forever loop
  IF DI1=1 ;* If digital input 1 is on, execute the statements
    X0 ;* Move to zero
    WAITX ;* Wait for X-axis move to complete
    X1000 ;* Move to 1000
    WAITX ;* Wait for X-axis move to complete
  ENDIF
ENDWHILE ;* Go back to WHILE statement
END
```

**Standalone Example Program 5 – Single Thread**

Task: Using a subroutine, increment the motor by 1000 whenever the DI1 rising edge is detected.

```plaintext
HSPD=20000 ;* Set the high speed to 20000 pulses/sec
LSPD= 1000 ;* Set the low speed to 1000 pulses/sec
ACC=300 ;* Set the acceleration to 300 msec
EO=1 ;* Enable the motor power
V1=0 ;* Set variable 1 to zero
WHILE 1=1 ;* Forever loop
  IF DI1=1 ;* If digital input 1 is on, execute the statements
    GOSUB 1 ;* Move to zero
  ENDIF
ENDWHILE ;* Go back to WHILE statement
END

SUB 1
XV1 ;* Move to V1 target position
WAITX ;* Wait for X-axis move to complete
V1=V1+1000 ;* Increment V1 by 1000
WHILE DI1=1 ;* Wait until the DI1 is turned off so that
  1000 increment is not continuously done
ENDWHILE
ENDSUB
```

**Standalone Example Program 6 – Single Thread**

Task: If digital input 1 is on, move to position 1000. If digital input 2 is on, move to position 2000. If digital input 3 is on, move to position 3000. If digital input 5 is on, home the motor in negative direction. Use digital output 1 to indicate that the motor is moving or not moving.

```plaintext
HSPD=20000 ;* Set the high speed to 20000 pulses/sec
LSPD= 1000 ;* Set the low speed to 1000 pulses/sec
ACC=300 ;* Set the acceleration to 300 msec
EO=1 ;* Enable the motor power
WHILE 1=1 ;* Forever loop
  IF DI1=1 ;* If digital input 1 is on
    X1000 ;* Move to 1000
    WAITX ;* Wait for X-axis move to complete
    X2000 ;* Move to 2000
    WAITX ;* Wait for X-axis move to complete
    X3000 ;* Move to 3000
    WAITX ;* Wait for X-axis move to complete
    HOMING ;* Home the motor in negative direction
  ENDIF
ENDWHILE ;* Go back to WHILE statement
END
```
**Standalone Example Program 7 – Multi Thread**

Task: Program 0 will continuously move the motor between positions 0 and 1000. Simultaneously, program 1 will control the status of program 0 using digital inputs.

```
PRG 0
HSPD=20000
LSPD=500
ACC=500
WHILE 1=1
  X0
  WAITX
  X1000
  WAITX
ENDWHILE
END

PRG 1
WHILE 1=1
  IF DI1=1
    ABORTX
    SR0=0
  ELSE
    SR0=1
  ENDIF
ENDWHILE
END
```

```
X1000
WAITX
ELSEIF DI2=1
  X2000
  WAITX
ELSEIF DI3=1
  X3000
  WAITX
ELSEIF DI5=1
  HOMEX-
  WAITX
ENDIF
V1=MSTX
V2=V1&7
IF V2!=0
  DO1=1
ELSE
  DO1=0
ENDIF
ENDWHILE
```

```
V1=MSTX
V2=V1&7
DO1=1
ELSE
  DO1=0
ENDIF
ENDWHILE
```

```
PRG 0
HSPD=20000
LSPD=500
ACC=500
WHILE 1=1
  X0
  WAITX
  X1000
  WAITX
ENDWHILE
END
```

```
V1=MSTX
V2=V1&7
DO1=1
ELSE
  DO1=0
ENDIF
ENDWHILE
```

```
PRG 1
WHILE 1=1
  IF DI1=1
    ABORTX
    SR0=0
  ELSE
    SR0=1
  ENDIF
ENDWHILE
END
```

```
V1=MSTX
V2=V1&7
DO1=1
ELSE
  DO1=0
ENDIF
ENDWHILE
```

```
PRG 0
HSPD=20000
LSPD=500
ACC=500
WHILE 1=1
  X0
  WAITX
  X1000
  WAITX
ENDWHILE
END
```

```
V1=MSTX
V2=V1&7
DO1=1
ELSE
  DO1=0
ENDIF
ENDWHILE
```

```
PRG 1
WHILE 1=1
  IF DI1=1
    ABORTX
    SR0=0
  ELSE
    SR0=1
  ENDIF
ENDWHILE
END
```

```
V1=MSTX
V2=V1&7
DO1=1
ELSE
  DO1=0
ENDIF
ENDWHILE
```
Standalone Example Program 8 – Multi Thread

Task: Program 0 will continuously move the motor between positions 0 and 1000. Simultaneously, program 1 will monitor the communication time-out parameter and triggers digital output 1 if a time-out occurs. Program 1 will also stop all motion, disable program 0 and then re-enable it after a delay of 3 seconds when the error occurs.

PRG 0
HSPD= 1000 ;* Set high speed to 1000 pps
LSPD=500 ;* Set low speed to 500pps
ACC=500 ;* Set acceleration to 500ms
TOC=5000 ;* Set time-out counter alarm to 5 seconds
EO=1 ;* Enable motor
WHILE 1=1 ;* Forever loop
  X0
  WAITX
  X1000
  WAITX
ENDWHILE
END

PRG 1
WHILE 1=1 ;* Forever loop
  V1=MSTX&2048 ;* Get bit time-out counter alarm variable
  IF V1 = 2048 ;* If time-out counter alarm is on
    SR0=0 ;* Stop program 0
    ABORTX
    DO=0 ;* Set DO=0
    DELAY=3000 ;* Delay 3 seconds
    SR0=1 ;* Turn program 0 back on
    DO=1 ;* Set DO=1
  ENDIF
ENDWHILE
END

* Start of Program 0
* Set high speed to 1000 pps
* Set low speed to 500pps
* Set acceleration to 500ms
* Set time-out counter alarm to 5 seconds
* Enable motor
* Move loop
* Move to position 0
* Wait for the move to complete
* Move to position 1000
* Wait for the move to complete
* Go back to WHILE statement
* End Program 0
* Start of Program 1
* Forever loop
* Get bit time-out counter alarm variable
* If time-out counter alarm is on
* Stop program 0
* Abort the motor
* Set DO=0
* Delay 3 seconds
* Turn program 0 back on
* Set DO=1
* Go back to WHILE statement
* End Program 1
Where to Go Next

Continue with the next chapter, Chapter 8 - Interactive Commands & USB Communications. Previous sections described how to use the PMX-4EX-SA software to communicate to the DMM-0400. Section 8 describes the basic ASCII commands and USB communications when using some other software such as Matlab or Labview.
Chapter 8
Interactive Commands & USB Communications

About This Chapter

Introduction

This chapter describes the USB communications and ASCII commands in detail to communicate to the DMM-0400. Previous sections explain the communication with the PMX-4EX-SA software. The user does not necessarily need to use the PMX-4EX-SA software to communicate to the DMM-0400. Visual BASIC, Visual C++, Labview, Matlab or other software can communicate to the DMM-400 through the USB connection. The command set described in this section are interactive commands that are sent through the USB port.

Topics

This chapter covers the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB Communication Description</td>
<td>96</td>
</tr>
<tr>
<td>Motion Control Overview and Interactive Command Description</td>
<td>98</td>
</tr>
<tr>
<td>ASCII Programming Language</td>
<td>116</td>
</tr>
<tr>
<td>Error Codes</td>
<td>124</td>
</tr>
<tr>
<td>Where to Go Next</td>
<td>125</td>
</tr>
</tbody>
</table>
USB Communication Description

PMX-4EX-SA USB communication is USB 2.0 compliant.

Communication between the PC and PMX-4EX-SA is done using Windows compatible DLL API function calls as shown below. Windows programming language such as Visual BASIC, Visual C++, LAB View, or any other programming language that can use DLL can be used to communicate with the Performax module.

Typical communication transaction time between PC and PMX-4EX-SA for sending a command from a PC and getting a reply from PMX-4EX-SA using the `fnPerformaxComSendRecv()` API function is in single digit milliseconds. This value will vary with CPU speed of PC and the type of command.

USB Communication API Functions

For USB communication, following DLL API functions are provided.

```c
BOOL fnPerformaxComGetNumDevices(OUT LPDWORD lpNumDevices);
- This function is used to get total number of all types of Performax and Performax USB modules connected to the PC.

BOOL fnPerformaxComGetProductString(IN DWORD dwNumDevices,
                                      OUT LPVOID lpDeviceString,
                                      IN DWORD dwOptions);
- This function is used to get the Performax or Performax product string. This function is used to find out Performax USB module product string and its associated index number. Index number starts from 0.

BOOL fnPerformaxComOpen(IN DWORD dwDeviceNum,
                         OUT HANDLE* pHandle);
- This function is used to open communication with the Performax USB module and to get communication handle. dwDeviceNum starts from 0.

BOOL fnPerformaxComClose(IN HANDLE pHandle);
- This function is used to close communication with the Performax USB module.

BOOL fnPerformaxComSetTimeouts(IN DWORD dwReadTimeout,
                                DWORD dwWriteTimeout);
- This function is used to set the communication read and write timeout. Values are in milliseconds. This must be set for the communication to work. Typical value of 1000 msec is recommended.

BOOL fnPerformaxComSendRecv(IN HANDLE pHandle,
                             IN LPVOID wBuffer,
                             IN DWORD dwNumBytesToWrite,
                             IN DWORD dwNumBytesToRead,
                             OUT LPVOID rBuffer);
- This function is used to send command and get reply. Number of bytes to read and write must be 64 characters.

BOOL fnPerformaxComFlush(IN HANDLE pHandle)
- Flushes the communication buffer on the PC as well as the USB controller. It is recommended to perform this operation right after the communication handle is opened.
```
USB Communication Issues

A common problem that users may have with USB communication is that after sending a command from the PC to the device, the response is not received by the PC until another command is sent. In this case, the data buffers between the PC and the USB device are out of sync. Below are some suggestions to help alleviate this issue.

1) **Buffer Flushing**: If USB communication begins from an unstable state (i.e. your application has closed unexpectedly), it is recommended to first flush the USB buffers of the PC and the USB device. See the following function prototype below:

   ```
   BOOL fnPerformaxComFlush(IN HANDLE pHandle)
   ```

   **Note**: fnPerformaxComFlush is only available in the most recent PerformaxCom.dll which is not registered by the standard USB driver installer. A sample of how to use this function along with this newest DLL is available for download from the following website:


2) **USB Cable**: Another source of USB communication issues may come from the USB cable. Confirm that the USB cable being used has a noise suppression choke. See photo below:

   ![USB Cable](image_url)

   **Figure 39** USB Cable
Motion Control Overview and Interactive Command Description

All the commands described in this section are interactive commands and are not analogous to stand-alone commands. Refer to the “Standalone Language Specification” section for details regarding stand-alone commands.

**Motion Profile**

By default, the PMX-4EX-SA uses trapezoidal velocity profile. See Figure 40.

![Figure 40 Trapezoidal Velocity Profile](image)

S-curve velocity profile can also be achieved by using the `SCV[axis]` command. See Figure 41.

![Figure 41 S-Curve Velocity Profile](image)

High speed and low speed are in pps (pulses/second). Use `HS[axis]` and `LS[axis]` to set/get individual high speed and low speed settings. To set/get the global high speed and low speed values use the `HS` and `LS` commands.

Acceleration and deceleration time are in milliseconds and are symmetrical. Use the `ACC[axis]` command to set/get individual acceleration/deceleration values. To set/get the global acceleration value, use the `ACC` command.

---

**Note:** By default, moves by a single axis use global speed settings, unless individual high speed, low speed and acceleration values for that axis are non-zero.
Example: To set the high-speed of the X-axis to 1500 pulses/second, and the Y-axis to 2000 pulses/second, issue the following speed setting commands:

\[
\begin{align*}
\text{HSX} & = 1500 & \text{set high speed for x-axis only} \\
\text{HSY} & = 2000 & \text{set high speed for y-axis only} \\
\text{LSX} & = 300 & \text{other parameters for the axis MUST be set as well for} \\
\text{LSY} & = 300 & \text{the controller to use the individual speed settings instead} \\
\text{ACCX} & = 100 & \text{of the global speed settings} \\
\text{ACCY} & = 100
\end{align*}
\]

It is possible to have unique acceleration and deceleration times. In order to decelerate using the value set in the \text{DEC[axis]} or \text{DEC} parameter, set \text{EDEC} to 1.

The minimum and maximum acceleration values depend on the high speed and low speed settings. Refer to Table A.0 and Figure A.0 in Appendix A for details.

**Pulse Speed**

Current pulse rate can be read using the \text{PS} command. For units, see Table 23

<table>
<thead>
<tr>
<th>Operation Mode</th>
<th>Speed Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>StepNLoop disabled</td>
<td>Pulse / sec</td>
</tr>
<tr>
<td>ALL interpolated moves</td>
<td>Pulse / sec</td>
</tr>
<tr>
<td>StepNLoop enabled and non-interpolated move</td>
<td>Encoder counts / sec</td>
</tr>
</tbody>
</table>

Table 23 Pulse Speed

This command returns the current speed of all axes. The \text{PS} return value has the following format:

\[[\text{Speed X}]:[\text{Speed Y}]:[\text{Speed Z}]:[\text{Speed U}]\]

**On-The-Fly Speed Change**

On-the-fly speed change can be achieved with the \text{SSPD[axis]} command. In order to use the \text{SSPD[axis]} command, s-curve velocity profile must be disabled.

**SSPD Mode**

The correct speed window must be selected in order to use the SSPD command. To select a speed window, use the \text{SSPDM[axis]} command. Refer to Appendix for details.

During on-the-fly speed change operation, you must keep the initial and destination speeds within the speed window.

For non on-the-fly speed change moves, set \text{SSPDM[axis]} to 0.
**Motor Status**

Motor status can be read anytime using **MST** command. Value of the motor status is replied as an integer with following bit assignment:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Motor in acceleration</td>
</tr>
<tr>
<td>1</td>
<td>Motor in deceleration</td>
</tr>
<tr>
<td>2</td>
<td>Motor running at constant speed</td>
</tr>
<tr>
<td>3</td>
<td>Alarm input status</td>
</tr>
<tr>
<td>4</td>
<td>Plus limit input switch status</td>
</tr>
<tr>
<td>5</td>
<td>Minus limit input switch status</td>
</tr>
<tr>
<td>6</td>
<td>Home input switch status</td>
</tr>
<tr>
<td>7</td>
<td>Plus limit error. This bit is latched when plus limit is hit during motion. This error must be cleared using the <strong>CLR</strong> command before issuing any subsequent move commands.</td>
</tr>
<tr>
<td>8</td>
<td>Minus limit error. This bit is latched when minus limit is hit during motion. This error must be cleared using the <strong>CLR</strong> command before issuing any subsequent move commands.</td>
</tr>
<tr>
<td>9</td>
<td>Alarm error. This bit is latched when alarm is triggered during motion. This error must be cleared using the <strong>CLR</strong> command before issuing any subsequent move commands.</td>
</tr>
<tr>
<td>10</td>
<td>Reserved</td>
</tr>
<tr>
<td>11</td>
<td>TOC time-out status</td>
</tr>
</tbody>
</table>

Table 24 Motor Status

This command returns the motor status for all axes, as well as other information. The **MST** return value has the following format:

```
[Motor Stat X];
[Motor Stat Y];
[Motor Stat Z];
[Motor Stat U];
[Buffer enabled];
[Buffer start];
[Buffer end];
[Available Buffer];
[Move mode]
```

- **Motor Stat [X/Y/Z/U]** – Provide motor status of the axis
- **Buffer enabled** – Buffer linear interpolated move status (0: off, 1: on)
- **Buffer start** – The index of the current command in the buffer
- **Buffer end** – The index of the last command in the buffer
- **Available Buffer** – The amount of empty spaces in the buffer
- **Move mode** – move mode (0: ABS, 1: INC)
Individual/Linear Interpolation Moves

For individual axis control use X, Y, Z, and U command followed by the target position value. A single move command can consist of up to 4 target positions (one for each axis). If more than one axis is specified, the motion will be linearly interpolated.

Individual/Linear Move Examples:

[X1000]: Move X-axis to position 1000.
[X1000 Y1000]: Move X-axis to position 1000, Y-axis to position 1000 using linear interpolation.
[X1000 Y1000 Z100]: Move X-axis to position 1000, Y-axis to position 1000, Z-axis to position 100 using linear interpolation.
[X1000 Y1000 Z100 U800]: Move X-axis to position 1000, Y-axis to position 1000, Z-axis to position 100, U-axis to position 800 using linear interpolation.
[X1000 U800]: Move X-axis to position 1000, U-axis to position 800 using linear interpolation.

Individual/Linear Interpolation moves can be performed in two modes: incremental mode. To set move modes, use the INC and ABS commands respectively.

Move Mode Examples:

[X1000] – INC mode: The motor will move by 1000 from the current position.
[X1000] – ABS mode: The motor will move to absolute position 1000.

Circular Interpolation Moves

PMX-4EX-SA supports circular interpolation moves using the CIRP and CIRN commands. Circles are drawn using X, Y axes only.


Note: The maximum allow radius is 134,216,773 pulses on arc or circular moves. All arc or circular moves are interpreted as absolute moves.
Arc Interpolation Moves

PMX-4EX-SA supports arc interpolation moves using the **ARCP** and **ARCN** commands. Arcs are drawn using X, Y axes only.

**ARCP**[X][Y][θ][θ] – Draw arc in CW direction where [X] [Y] signifies X,Y position of the circle center, θ signifies the absolute arc angle, and θ signifies the relative arc angle.

**ARCN**[X][Y][θ][θ] – Draw arc in CCW direction where [X] [Y] signifies X,Y position of the circle center, θ signifies the absolute arc angle, and θ signifies the relative arc angle.

---

**Note:** The maximum allowable radius is 134,216,773 pulses on arc or circular moves. All arc or circular moves are interpreted as absolute moves.

---

Angle values are whole number in thousandths. For example, 45 degrees is 45,000.

θA: The absolute angle standard is depicted in Figure 6.3. For example, to move to position A in Figure 6.3, θA should always be 90,000 (90°). Likewise, to move to position B in Figure 6.3, the absolute θA should always be 180,000 (180°). This is independent of the starting position and move direction.

θR: The relative angle is calculated by finding the total degrees that the move will take.

**Arc Move Examples**

**Example 1:**
- Arc start position: (0,1000)
- Arc start absolute angle: 90,000 (90°)
- Arc end position: (1000,0) in CW direction
- Move amount (degrees): 90,000 (90°)
- Move command: ARCP0:0: 180000:90000

**Example 2:**
- Arc start position: (-1000,0)
Arc start absolute angle: 0 (0°)
Arc end position: (0,1000) in CCW direction
Move amount (degrees): 270,000 (270°)
Move command: ARCN0:0:450000:270000

Figure 43  Arc Interpolation Moves

**Buffered Linear Interpolation Moves**

PMX-4EX-SA supports buffered linear coordinated motions for X, Y, and Z-axes using the I command. Each move has its own constant speed setting.

**Example:** To move to location X, Y, Z to 1000, 2000, 3000 position with speed of 250 pps, use the following command - I1000:2000:3000:250.

**Notes:**

**Manual Acceleration Control:** When IACC=0, acceleration/deceleration must be done manually. To control the acceleration or deceleration manually, gradually increase or decrease the speed value for each interpolated move.

**Automatic Acceleration Control:** When IACC=1, acceleration/deceleration is processed automatically. In this case, the speed acceleration profile will be automatically generated between sequential buffered moves. The acceleration/deceleration value used for automatic acceleration control is found in the global acceleration value (ACC).

Linear interpolation buffer move size is 36 points. To turn on and off buffer move, use the **BO** and **BF** command respectively. When enabled, as soon as the first I command is issued the motion will start.

Buffered moves apply only to X, Y and Z axes.
Buffered move operation cannot be used while StepNLoop is enabled.
For command information for buffer move status, see **Motor Status** section.
On-The-Fly Target Position Change

On-the-fly target position change can be achieved using the \texttt{T[axis][value]} command. While the motor is moving, \texttt{T[axis][value]} will change the final destination of the motor. If the motor has already passed the new target position, it will reverse direction once the target position change command is issued.

\textbf{Note:} If a \texttt{T} command is sent while the controller is not performing a target move, the command is not processed. Instead, an error response is returned.

Homing

Home search sequence involves moving the motor towards the home or limit switches and then stopping when the relevant input is detected.

The PMX-4EX-SA has five different homing routines. Use the \texttt{H} command to perform them.

\texttt{H[axis][direction + or -][homing mode 0,1,2,3,4]}

Four homing modes:
- 0 – Home Input Only (High speed only)
- 1 – Limit Input Only
- 2 – Home Input and Z-Index
- 3 – Z-Index Only
- 4 – Home Input Only (High speed and low speed)

\textbf{MODE 0 : Home Input Only (High Speed Only)}

Figure 44 shows the homing routine.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{homing.png}
\caption{Homing - Home Input Only (High Speed)}
\end{figure}

A. Starts the motor from low speed and accelerates to high speed.
B. As soon as the home input is triggered, the position counter is reset to zero and the motor begins to decelerate to low speed. As the motor decelerates, the position counter keeps counting with reference to the zero position.
C. Once low speed is reached, the motor stops. The position is non-zero.
**MODE 1 : Limit Only**

Figure 45 shows the homing routine.

![Figure 45 Homing - Limit Only](image)

A. Issuing a limit home command starts the motor from low speed and accelerates to high speed.

B. The corresponding limit is triggered, the motor stops immediately and the position is set to zero.

**MODE 2 : Home and Z-index**

Figure 46 shows the homing routine.

![Figure 46 Homing - Home and Z-Index](image)

A. Issuing the command starts the motor from low speed and accelerates to high speed.

B. As soon as the home input is triggered, the motor decelerates to low speed.

C. Once low speed is reached, the motor begins to search for the z-index pulse.

D. Once the z-index pulse is found, the motor stops and the position is set to zero.
MODE 3 : Z-index only
Figure 47 shows the homing routine.

Figure 47 Homing - Z-Index Only

A. Issuing the command starts the motor at low speed.
B. Once the z-index pulse is found, the motor stops and the position is set to zero.

MODE 4 : Home Input Only (High Speed and Low Speed)
Figure 48 shows the homing routine.

Figure 48 Homing - Home Input Only (High Speed and Low Speed)

A. Starts the motor from low speed and accelerates to high speed.
B. As soon as the home input is triggered, the position counter is reset to zero and the motor decelerates to low speed.
C. Once low speed is reached, the motor reverses direction to search for the home switch.
D. Once the home switch is reached, it will continue past the home switch until the home switch is off.
E. The motor is now past the home input. The motor now moves back towards the home switch at low speed.
F. The home input is triggered again, the position counter is reset to zero and the motor stops immediately.

Jogging

Use J command for jogging the motor. Use the following format:

\[ J[\text{axis}][\text{direction} + \text{ or } -] \]

Stopping

When the motor is moving, the ABORT[\text{axis}] command will immediately stop an individual axis. Use the ABORT command to immediately stop ALL axes.

To employ deceleration on a stop, use the STOP[\text{axis}] to stop an individual axis. Use the STOP command to stop ALL axes.
Note: If an interpolation operation is in process when a STOP[axis] or ABORT[axis] command is entered, all axes involved in the interpolation operation will stop.

Polarity

Using the PO[axis] command to get and set polarity of the signal below. The format is as an integer with the following bit assignment:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Home</td>
</tr>
<tr>
<td>1</td>
<td>Alarm</td>
</tr>
<tr>
<td>2</td>
<td>Limit (X-axis limit input setting controls limit switch polarity for all axes)</td>
</tr>
<tr>
<td>3</td>
<td>Direction</td>
</tr>
</tbody>
</table>

Table 25 Polarity

Motor Position

Motor positions can be read using the PP command which returns the pulse position of all 4 axes. The return value has the following format:

[Pulse X]:[Pulse Y]:[Pulse Z]:[Pulse U]

Encoder positions can be read using PE command which returns the encoder position of all 4 axes. Encoders are set to 4X reading. The return value has the following format:

[Encoder X]:[Encoder Y]:[Encoder Z]:[Encoder U]

To manually set/get the pulse position of an individual axis, use the P[axis] command. Note that setting the pulse position is not allowed if StepNLoop is enabled.

To manually set/get the encoder position of an individual axis, use the E[axis] command.

Limits and Alarm

If positive limit switch is triggered while moving in positive direction, the motor will immediately stop and the motor status bit for positive limit error is set. The same is for the negative limit while moving in the negative direction.

If the alarm input for an axis is triggered during movement in either direction, the motor will immediately stop and the motor status bit for alarm error is set.

Once the limit or alarm error is set, use the CLR[axis] command to clear the error.

The limit and alarm error states can be ignored by setting IERR=1. In this case, the motor will still stop when the appropriate switch is triggered; however, it will not enter an error state.
Latch Inputs

The PMX-4EX-SA module provides the following high speed position latch inputs.

These inputs perform high speed position capture of both pulse and encoder positions but does not reset the pulse or encoder position counters.

See corresponding latch input for each axis below:

<table>
<thead>
<tr>
<th>Axis</th>
<th>Synchronization Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>DI4</td>
</tr>
<tr>
<td>Y</td>
<td>DI6</td>
</tr>
<tr>
<td>Z</td>
<td>DI5</td>
</tr>
<tr>
<td>U</td>
<td>DI8</td>
</tr>
</tbody>
</table>

Table 26 Latch Inputs

Note: When StepNLooP mode is enabled, the position value should be ignored.

Use the LT[Axis] command to enable and disable latch feature. To read the latch status, use LT[Axis]S command.

Following are return value description for LT[Axis]S command:

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Latch off</td>
</tr>
<tr>
<td>1</td>
<td>Latch on and waiting for latch trigger</td>
</tr>
<tr>
<td>2</td>
<td>Latch Triggered</td>
</tr>
</tbody>
</table>

Table 27 Latch Return Values

Once the latch is triggered, the triggered position can be retrieved using LT[Axis]P (latched pulse position) and LT[Axis]E (latched encoder position) commands.
Digital Inputs/Outputs and Enable Outputs

PMX-4EX-SA module comes with 8 digital inputs and 8 digital outputs and 4 enable outputs.

Inputs
Read digital input status using the DI command.
Digital input values can also be referenced one bit at a time by the DI[1-8] commands. Note that the indexes are 1-based for the bit references (i.e. DI1 refers to bit 0, not bit 1)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Bit-Wise Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Digital Input 1</td>
<td>DI1</td>
</tr>
<tr>
<td>1</td>
<td>Digital Input 2</td>
<td>DI2</td>
</tr>
<tr>
<td>2</td>
<td>Digital Input 3</td>
<td>DI3</td>
</tr>
<tr>
<td>3</td>
<td>Digital Input 4</td>
<td>DI4</td>
</tr>
<tr>
<td>4</td>
<td>Digital Input 5</td>
<td>DI5</td>
</tr>
<tr>
<td>5</td>
<td>Digital Input 6</td>
<td>DI6</td>
</tr>
<tr>
<td>6</td>
<td>Digital Input 7</td>
<td>DI7</td>
</tr>
<tr>
<td>7</td>
<td>Digital Input 8</td>
<td>DI8</td>
</tr>
</tbody>
</table>

Table 28 Digital Inputs

Digital Outputs
The digital output status can be controlled using the DO command. DO value must be within the range of 0-255.

Digital output values can also be referenced one bit at a time by the DO[1-8] commands. Note that the indexes are 1-based for the bit references (i.e. DO1 refers to bit 0, not bit 1)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Bit-Wise Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Digital Output 1</td>
<td>DO1</td>
</tr>
<tr>
<td>1</td>
<td>Digital Output 2</td>
<td>DO2</td>
</tr>
<tr>
<td>2</td>
<td>Digital Output 3</td>
<td>DO3</td>
</tr>
<tr>
<td>3</td>
<td>Digital Output 4</td>
<td>DO4</td>
</tr>
<tr>
<td>4</td>
<td>Digital Output 5</td>
<td>DO5</td>
</tr>
<tr>
<td>5</td>
<td>Digital Output 6</td>
<td>DO6</td>
</tr>
<tr>
<td>6</td>
<td>Digital Output 7</td>
<td>DO7</td>
</tr>
<tr>
<td>7</td>
<td>Digital Output 8</td>
<td>DO8</td>
</tr>
</tbody>
</table>

Table 29 Digital Outputs

The initial state of the digital outputs can be defined by setting the DOBOOT register to the desired initial digital output value. The value is stored to flash memory once the STORE command is issued.

Enable Outputs
The enable output status can be controlled using the EO command. EO value must be within the range of 0-15.

Enable output values can also be referenced one bit at a time by the EO[1-4] commands. Note that the indexes are 1-based for the bit references (i.e. EO1 refers to bit 0, not bit 1)
Bit | Description | Bit-Wise Command
--- | --- | ---
0 | Enable Output 1 [X-axis] | EO1
1 | Enable Output 2 [Y-axis] | EO2
2 | Enable Output 3 [Z-axis] | EO3
3 | Enable Output 4 [U-axis] | EO4

Table 30 Enable Outputs

The initial state of the enable outputs can be defined by setting the EOBOOT register to the desired initial enable output value. The value is stored to flash memory once the STORE command is issued.

**Sync Outputs**

PMX-4EX-SA has synchronization digital outputs for each axis. The synchronization signal output is triggered when the encoder position value meets the set condition. See synchronization output for each axis below:

<table>
<thead>
<tr>
<th>Axis</th>
<th>Synchronization Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>DO1</td>
</tr>
<tr>
<td>Y</td>
<td>DO2</td>
</tr>
<tr>
<td>Z</td>
<td>DO3</td>
</tr>
<tr>
<td>U</td>
<td>DO4</td>
</tr>
</tbody>
</table>

Table 31 Sync Outputs

**Note:** While feature is enabled for an axis, the corresponding digital output cannot be controlled by user.

Use `SYN[axis]O` to enable the synchronization output feature for an axis.

Use `SYN[axis]F` to disable the synchronization output feature for an axis.

Use `SYN[axis]P` to read and set the synchronization position value for an axis. (28-bit signed number)

Use `SYN[axis]C` to set the synchronization condition.

1 - Turn the output on when the encoder position is EQUAL to sync position. If the synchronization output is done during motion, the sync output pulse will turn on only when the encoder position and sync position are equal.
2 - Turns output on when the encoder position is LESS than the sync position.
3 - Turns output on when the encoder position is GREATER than sync position.

Use `SYN[axis]T` to set the pulse width output time (ms). This parameter is only used if the synchronization condition is set to 1. Note the maximum pulse width is 10 ms. If this parameter is set to 0, the output pulse will depend on how long the encoder value is equal to the sync position.

Use `SYN[axis]S` to read the synchronization output status for an axis

0 - Sync output feature is off
1 - Waiting for sync condition
2 - Sync condition occurred
Analog Inputs

8 x 10-bit analog inputs are available on PMX-4EX-SA. Use **AI[1-8]** command to read the analog input value. Range is from 0-5000 mV.

Joystick Control

Joystick control is available on PMX-4EX-SA. When this mode is enabled, the pulse speed and direction output can be controlled by corresponding analog input. See the axis to analog input relationship in the table below:

<table>
<thead>
<tr>
<th>Axis</th>
<th>Analog Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>AI1</td>
</tr>
<tr>
<td>Y</td>
<td>AI2</td>
</tr>
<tr>
<td>Z</td>
<td>AI3</td>
</tr>
<tr>
<td>U</td>
<td>AI4</td>
</tr>
</tbody>
</table>

Table 32 Joystick Control

Maximum joystick speed is set using the **JV1, JV2, JV3** and **JV4** variables.

Maximum speed change (delta) is set using the **JV5, JV6, JV7** and **JV8** variables.

Maximum joystick value is set using the **JMAX[axis]** variable.

Minimum joystick value is set using the **JMIN[axis]** variable.

Tolerance of the zero joystick position, use **JV9, JV10, JV11** and **JV12** variables.

During joystick operation, analog input of **JMIN[axis]** mV to **MID** mV represents negative joystick direction and analog input of **MID** mV to **JMAX[axis]** mV represents positive joystick direction.

**MID** mV represents the zero joystick position. This value is calculated with the following formula:

\[
\text{MID} = \left( \frac{\text{JMAX[axis]} - \text{JMIN[axis]}}{2} \right) + \text{JMIN[axis]}
\]

**Note:** There is no command corresponding to **MID**. This is an internally calculated value.

Joystick control also has soft limit controls. Limits are broken into: negative outer limit, negative inner limit, positive inner limit and positive outer limit.

When moving in positive direction, as soon as the positive inner limit is crossed, the speed is reduced. If the position reaches the positive outer limit, the joystick speed is set to zero. Same goes for the negative direction and negative limits.

The behavior of the limits of the joystick control is explained by the following:

![Figure 49 Joystick Control](image)
## Summary of Joystick Control Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JV1</td>
<td>X-axis maximum joystick speed at 5000 mV and 0 mV</td>
</tr>
<tr>
<td>JV2</td>
<td>Y-axis maximum joystick speed at 5000 mV and 0 mV</td>
</tr>
<tr>
<td>JV3</td>
<td>Z-axis maximum joystick speed at 5000 mV and 0 mV</td>
</tr>
<tr>
<td>JV4</td>
<td>U-axis maximum joystick speed at 5000 mV and 0 mV</td>
</tr>
<tr>
<td>JV5</td>
<td>X-axis maximum speed change</td>
</tr>
<tr>
<td>JV6</td>
<td>Y-axis maximum speed change</td>
</tr>
<tr>
<td>JV7</td>
<td>Z-axis maximum speed change</td>
</tr>
<tr>
<td>JV8</td>
<td>U-axis maximum speed change</td>
</tr>
<tr>
<td>JV9</td>
<td>X-axis zero tolerance range for analog input</td>
</tr>
<tr>
<td>JV10</td>
<td>Y-axis zero tolerance range for analog input</td>
</tr>
<tr>
<td>JV11</td>
<td>Z-axis zero tolerance range for analog input</td>
</tr>
<tr>
<td>JV12</td>
<td>U-axis zero tolerance range for analog input</td>
</tr>
<tr>
<td>JL1</td>
<td>X-axis negative outer limit</td>
</tr>
<tr>
<td>JL2</td>
<td>X-axis negative inner limit</td>
</tr>
<tr>
<td>JL3</td>
<td>X-axis positive inner limit</td>
</tr>
<tr>
<td>JL4</td>
<td>X-axis positive outer limit</td>
</tr>
<tr>
<td>JL5</td>
<td>Y-axis negative outer limit</td>
</tr>
<tr>
<td>JL6</td>
<td>Y-axis negative inner limit</td>
</tr>
<tr>
<td>JL7</td>
<td>Y-axis positive inner limit</td>
</tr>
<tr>
<td>JL8</td>
<td>Y-axis positive outer limit</td>
</tr>
<tr>
<td>JL9</td>
<td>Z-axis negative outer limit</td>
</tr>
<tr>
<td>JL10</td>
<td>Z-axis negative inner limit</td>
</tr>
<tr>
<td>JL11</td>
<td>Z-axis positive inner limit</td>
</tr>
<tr>
<td>JL12</td>
<td>Z-axis positive outer limit</td>
</tr>
<tr>
<td>JL13</td>
<td>U-axis negative outer limit</td>
</tr>
<tr>
<td>JL14</td>
<td>U-axis negative inner limit</td>
</tr>
<tr>
<td>JL15</td>
<td>U-axis positive inner limit</td>
</tr>
<tr>
<td>JL16</td>
<td>U-axis positive outer limit</td>
</tr>
<tr>
<td>JMAXX</td>
<td>X-axis maximum mV</td>
</tr>
<tr>
<td>JMAXY</td>
<td>Y-axis maximum mV</td>
</tr>
<tr>
<td>JMAXZ</td>
<td>Z-axis maximum mV</td>
</tr>
<tr>
<td>JMAXU</td>
<td>U-axis maximum mV</td>
</tr>
<tr>
<td>JMINX</td>
<td>X-axis minimum mV</td>
</tr>
<tr>
<td>JMINY</td>
<td>Y-axis minimum mV</td>
</tr>
<tr>
<td>JMINZ</td>
<td>Z-axis minimum mV</td>
</tr>
<tr>
<td>JMINU</td>
<td>U-axis minimum mV</td>
</tr>
</tbody>
</table>

Table 33  Summary of Joystick Control Parameters
To enable/disable joystick control for an axis, use the JE command. Joystick enable parameter is a 4 bit value. For example, digital output value of 15 (1111 in binary or 0xF in hex) means joystick feature is enabled on all axes.

**Note:** If joystick control is enabled, StepNLoop is automatically disabled.

**Joystick Example (X-axis):**

JMAXX = 4500 mV  
JMINX = 100 mV  
JMID = ((4500-100)/2 + 100) = 2300 mV  
JV9 = 100

In this case, maximum speed in the negative direction would be achieved when AI1 reaches 100 mV, and maximum speed in the positive direction would be achieved when AI1 reaches 4500 mV.

Since the zero tolerance (JV9) is set to 100, no motion will be performed if AI1 is between 2400 mV to 2600 mV.

**StepNLoop Closed Loop Control**

PMX-4EX-SA features a closed-loop position verification algorithm called StepNLoop (SNL). The algorithm requires the use of an incremental encoder.

SNL performs the following operations:

1) **Position Verification:** At the end of any targeted move, SNL will perform a correction if the current error is greater than the tolerance value.

2) **Delta Monitoring:** The delta value is the difference between the actual and the target position. When delta exceeds the error range value, the motor is stopped and the SNL Status goes into an error state. Delta monitoring is performed during moves — including homing and jogging. To read the delta value, use the DX command.

See Table 34 StepNLoop Closed Loop Control for a list of the SNL control parameters.

<table>
<thead>
<tr>
<th>SNL Parameter</th>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>StepNLoop Ratio</td>
<td>†Ratio between motor pulses and encoder counts. This ratio will depend on the motor type, microstepping, encoder resolution and decoding multiplier. Value must be in the range [0.001 , 999.999].</td>
<td>SLR[axis]</td>
</tr>
<tr>
<td>Tolerance</td>
<td>Maximum error between target and actual position that is considered “In Position”. In this case, no correction is performed. Units are in encoder counts.</td>
<td>SLT[axis]</td>
</tr>
<tr>
<td>Error Range</td>
<td>Maximum error between target and actual position that is not considered a serious error. If the error exceeds this value, the motor will stop immediately and go into an error state.</td>
<td>SLE[axis]</td>
</tr>
<tr>
<td>Correction Attempt</td>
<td>Maximum number of correction tries that the controller will attempt before stopping and going into an error state.</td>
<td>SLA [axis]</td>
</tr>
</tbody>
</table>

Table 34 StepNLoop Closed Loop Control

†A convenient way to find the StepNLoop ratio is to set EX=0, PX=0 and move the motor + 1000 pulses. The ratio can be calculated by dividing 1000 by the resulting EX value. Note that the value must be positive. If it is not, then
the direction polarity must be adjusted. This test can be performed on all axes that require StepNLoop. See Table 25 for details.

To enable/disable the SNL feature use the SL[axis] command. To read the SNL status, use SLS[axis] command to read the status.

See Table 35 StepNLoop Return Values for a list of the SLS[axis] return values.

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Idle</td>
</tr>
<tr>
<td>1</td>
<td>Moving</td>
</tr>
<tr>
<td>2</td>
<td>Correcting</td>
</tr>
<tr>
<td>3</td>
<td>Stopping</td>
</tr>
<tr>
<td>4</td>
<td>Aborting</td>
</tr>
<tr>
<td>5</td>
<td>Jogging</td>
</tr>
<tr>
<td>6</td>
<td>Homing</td>
</tr>
<tr>
<td>7</td>
<td>Z-Homing</td>
</tr>
<tr>
<td>8</td>
<td>Correction range error. To clear this error, use CLRS or CLR command.</td>
</tr>
<tr>
<td>9</td>
<td>Correction attempt error. To clear this error, use CLRS or CLR command.</td>
</tr>
<tr>
<td>10</td>
<td>Stall Error. DX value has exceeded the correction range value. To clear this error, use CLRS or CLR command.</td>
</tr>
<tr>
<td>11</td>
<td>Limit Error</td>
</tr>
<tr>
<td>12</td>
<td>N/A (i.e. SNL is not enabled)</td>
</tr>
<tr>
<td>13</td>
<td>Limit homing</td>
</tr>
</tbody>
</table>

Table 35 StepNLoop Return Values

See Table 36 for SNL behavior within different scenarios.

<table>
<thead>
<tr>
<th>Condition</th>
<th>SNL behavior (motor is moving)</th>
<th>SNL behavior (motor is idle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>δ &lt;= SLT</td>
<td>Continue to monitor the DX[axis]</td>
<td>In Position. No correction is performed.</td>
</tr>
<tr>
<td>δ &gt; SLT AND δ &lt; SLE</td>
<td>Continue to monitor the DX[axis]</td>
<td>Out of Position. A correction is performed.</td>
</tr>
<tr>
<td>Correction Attempt &gt; SLA</td>
<td>NA</td>
<td>Max Attempt Error. Motor stops and signals and error.</td>
</tr>
</tbody>
</table>

Table 36 StepNLoop Conditions
Key

| [δ] | Error between the target position and actual position |
| SLT | Tolerance range |
| SLE | Error range |
| SLA | Max correction attempt |

**Notes:**

Once SNL is enabled, position move commands are in term of encoder position. For example, X1000 means to move the motor to encoder 1000 position. This applies to individual as well as interpolated moves.

Once SNL is enabled, the speed is in encoder speed. For example HSPD= 1000 when SNL is enabled means that the target high speed is 1000 encoder counts per second. This only applies to individual axis moves.

**Linear Interpolation w/ StepNLoop:** If StepNLoop is used during a linear interpolation move, StepNLoop must be enabled for all axes being moved. Also note that unlike the individual axis moves, the speed during a linear interpolation is calculated as pulse/sec, NOT encoder counts/sec.

**Arc/Circular Interpolation w/ StepNLoop:** If StepNLoop is used during an arc/circular interpolation move, StepNLoop must be enabled for both X and Y axes. Also note that unlike the individual axis and linear interpolation moves, the StepNLoop ratio of X and Y MUST be the same. Also note that the speed during an arc/circular interpolation move is calculated as pulse/sec, NOT encoder counts/sec.

**Device Number**

PMX-4EX-SA module provides the user with the ability to modify the unique device number. In order to make these changes, first store the desired number using the DN command. Note that this value must be within the range [4EX00,4EX99].

To write the values to the device’s flash memory, use the STORE command. After a complete power cycle, the new device number will be written to memory. Note that before a power cycle is completed, the settings will not take effect.

By default: Device name is set to: 4EX00

**Baud Rate Setting**

PMX-4EX-SA provides the user with the ability to set the desired baud rate of the serial communication. In order to make these changes, first store the desired baud rate by using the DB command.

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9600</td>
</tr>
<tr>
<td>2</td>
<td>19200</td>
</tr>
<tr>
<td>3</td>
<td>38400</td>
</tr>
<tr>
<td>4</td>
<td>57600</td>
</tr>
<tr>
<td>5</td>
<td>115200</td>
</tr>
</tbody>
</table>

**Table 37 Baud Rate Setting**

To write the values to the device’s flash memory, use the STORE command. After a complete power cycle, the new device number will be written to memory. Note that before a power cycle is completed, the settings will not take effect.

By default: Baud rate is set to: 1 (9600 bps)
ASCII Programming Language

Invalid command is returned with ? (Error Message). Always check for proper reply when command is sent. Like the commands, all responses are in ASCII form.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>Immediately stops all the motor if in motion. Abort turns off the buffered move.</td>
<td>OK</td>
</tr>
<tr>
<td>ABORTX</td>
<td>Immediately stops individual motor if in motion. Abort turns off the buffered move.</td>
<td>OK</td>
</tr>
<tr>
<td>ABORY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABORTZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABORTU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABS</td>
<td>Turns on absolute move mode</td>
<td>OK</td>
</tr>
<tr>
<td>ACC</td>
<td>Returns current global acceleration value in milliseconds.</td>
<td>OK</td>
</tr>
<tr>
<td>ACC=[Value]</td>
<td>Sets global acceleration value in milliseconds.</td>
<td>OK</td>
</tr>
<tr>
<td>ACCX</td>
<td>Returns current individual acceleration value in milliseconds.</td>
<td></td>
</tr>
<tr>
<td>ACCY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACCZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACCU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACCX=[value]</td>
<td>Sets individual acceleration value in milliseconds.</td>
<td>OK</td>
</tr>
<tr>
<td>ACCY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACCZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACCU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI[1-8]</td>
<td>Get analog input status. Units in mV</td>
<td>[0-5000]</td>
</tr>
<tr>
<td>ARCP[X]:[Y]: [OA]:[OR]</td>
<td>XY Arc interpolation move (CW direction)</td>
<td>OK</td>
</tr>
<tr>
<td>ARCN[X]:[Y]: [Oa]:[Oz]</td>
<td>XY Arc interpolation move (CCW direction)</td>
<td>OK</td>
</tr>
<tr>
<td>BF</td>
<td>Disable buffered move</td>
<td>OK</td>
</tr>
<tr>
<td>BO</td>
<td>Enable buffer move on</td>
<td>OK</td>
</tr>
<tr>
<td>CIRP[X]:[Y]</td>
<td>XY Circular interpolation move (CW direction)</td>
<td>OK</td>
</tr>
<tr>
<td>CIRN[X]:[Y]</td>
<td>XY Circular interpolation move (CCW direction)</td>
<td>OK</td>
</tr>
<tr>
<td>CLRX</td>
<td>Clears motor limit or alarm status bit. Also clears a StepNLoop errors</td>
<td>OK</td>
</tr>
<tr>
<td>CLRY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLRZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLRU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB</td>
<td>Return baud rate</td>
<td>[1,2,3,4,5]</td>
</tr>
<tr>
<td>DB=[value]</td>
<td>Set baud rate 1 – 9600 bps 2 – 19200 bps 3 – 38400 bps 4 – 57600 bps 5 – 115200 bps</td>
<td>OK</td>
</tr>
<tr>
<td>DEC</td>
<td>Returns the current global deceleration value in milliseconds</td>
<td>OK</td>
</tr>
<tr>
<td>DEC=[Value]</td>
<td>Sets the global deceleration value in milliseconds</td>
<td>OK</td>
</tr>
<tr>
<td>DECX</td>
<td>Returns the current individual deceleration value in milliseconds</td>
<td>OK</td>
</tr>
<tr>
<td>DECY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECU</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 38 ASCII Commands
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECX=[value]</td>
<td>Sets the individual deceleration value in milliseconds</td>
<td>OK</td>
</tr>
<tr>
<td>DECY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DI</td>
<td>Returns 8 bits of general purpose digital input.</td>
<td>[0-255] See Table 28</td>
</tr>
<tr>
<td>DI[1-8]</td>
<td>Returns bit status of general purpose digital input.</td>
<td>[0,1]</td>
</tr>
<tr>
<td>DIP</td>
<td>Returns the digital input polarity</td>
<td>[0,1]</td>
</tr>
<tr>
<td>DIP=[0 or 1]</td>
<td>Sets the digital input polarity</td>
<td>OK</td>
</tr>
<tr>
<td>DO</td>
<td>Returns 8 bits of general purpose digital output value.</td>
<td>[0-255] See Table 29</td>
</tr>
<tr>
<td>DO=[value]</td>
<td>Sets 8 bits of general purpose digital output.</td>
<td>OK</td>
</tr>
<tr>
<td>DO[1-8]</td>
<td>Returns bit of general purpose digital output value.</td>
<td>[0,1]</td>
</tr>
<tr>
<td>DO[1-8]=[value]</td>
<td>Sets bit of general purpose digital output.</td>
<td>OK</td>
</tr>
<tr>
<td>DOBOOT</td>
<td>Get DO boot-up state</td>
<td>See Table 29</td>
</tr>
<tr>
<td>DOBOOT=[Value]</td>
<td>Set DO boot-up state</td>
<td>OK</td>
</tr>
<tr>
<td>DOP</td>
<td>Returns the digital output polarity</td>
<td>[0,1]</td>
</tr>
<tr>
<td>DOP=[0,1]</td>
<td>Sets the digital output polarity</td>
<td>OK</td>
</tr>
<tr>
<td>DN</td>
<td>Return device name</td>
<td>[4EX00-4EX99]</td>
</tr>
<tr>
<td>DN=[value]</td>
<td>Set device name. value must be in the range [4EX00, 4EX99]</td>
<td>OK</td>
</tr>
<tr>
<td>DXX</td>
<td>Get StepNLoop delta value of axis</td>
<td></td>
</tr>
<tr>
<td>DXY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DXZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DXU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDEC</td>
<td>Returns the enable deceleration status</td>
<td>[0,1]</td>
</tr>
<tr>
<td>EDEC=[0 or 1]</td>
<td>Sets the enabled deceleration status</td>
<td>OK</td>
</tr>
<tr>
<td>EO</td>
<td>Returns 4 bits of enable output value.</td>
<td>[0-15]</td>
</tr>
<tr>
<td>EO[1-4]</td>
<td>Returns bit of enable output value.</td>
<td>[0,1]</td>
</tr>
<tr>
<td>EO[1-4]=[value]</td>
<td>Set bit of enable outputs.</td>
<td>OK</td>
</tr>
<tr>
<td>EOBOOT</td>
<td>Get EO boot-up state</td>
<td>See Table 30</td>
</tr>
<tr>
<td>EOBOOT=[Value]</td>
<td>Set EO boot-up state</td>
<td>OK</td>
</tr>
<tr>
<td>EOP</td>
<td>Returns the enable output polarity</td>
<td>[0,1]</td>
</tr>
<tr>
<td>EOP=[0,1]</td>
<td>Sets the enable output polarity</td>
<td>OK</td>
</tr>
<tr>
<td>EX=[value]</td>
<td>Set encoder value of axis</td>
<td></td>
</tr>
<tr>
<td>EY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GS[0-3 1]</td>
<td>Call a defined subroutine</td>
<td>OK</td>
</tr>
<tr>
<td>HS</td>
<td>Returns global high speed setting</td>
<td>[1-6,000,000]</td>
</tr>
<tr>
<td>HS=[value]</td>
<td>Sets global high speed</td>
<td>OK</td>
</tr>
<tr>
<td>HSX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td>Return</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>HSX=[value]</td>
<td>Sets individual high speed</td>
<td>OK</td>
</tr>
<tr>
<td>HSY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HX[+/-][mode]</td>
<td>Homes the motor in plus [+] or minus [-] direction using different homing</td>
<td>OK</td>
</tr>
<tr>
<td>HY[+/-][mode]</td>
<td>mode.</td>
<td></td>
</tr>
<tr>
<td>HZ[+/-][mode]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HU[+/-][mode]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I[X axis]:</td>
<td>XYZ interpolated move. Target move values are separated by <code>:</code> character.</td>
<td>OK</td>
</tr>
<tr>
<td>[Y axis]:</td>
<td>Last value is the constant speed that will be used in the move.</td>
<td></td>
</tr>
<tr>
<td>[Z axis]:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[speed]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IACC</td>
<td>Get automatic acceleration during buffer interpolated move status</td>
<td>[0-1]</td>
</tr>
<tr>
<td>IACC=[0 or 1]</td>
<td>Set automatic acceleration during buffer interpolated move status</td>
<td>OK</td>
</tr>
<tr>
<td>IERR</td>
<td>Get the ignore limit/alarm error status</td>
<td>[0-1]</td>
</tr>
<tr>
<td>IERR=[0 or 1]</td>
<td>Set the ignore limit/alarm error status</td>
<td>OK</td>
</tr>
<tr>
<td>INC</td>
<td>Enable incremental move mode</td>
<td>OK</td>
</tr>
<tr>
<td>JE</td>
<td>Get joystick enable status</td>
<td>[0-15]</td>
</tr>
<tr>
<td>JE=[value]</td>
<td>Set joystick enable status</td>
<td>OK</td>
</tr>
<tr>
<td>JX[+/-]</td>
<td>Jogs the motor in plus [+] or minus [-] direction.</td>
<td>OK</td>
</tr>
<tr>
<td>JY[+/-]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JZ[+/-]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JU[+/-]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JV[1-12]</td>
<td>Get joystick speed, delta and tolerance settings</td>
<td>See Table 33</td>
</tr>
<tr>
<td>JV[1-12]=[value]</td>
<td>Set joystick speed, delta and tolerance settings</td>
<td>OK</td>
</tr>
<tr>
<td>See Table 33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JL[1-16]</td>
<td>Get joystick soft limit settings</td>
<td>See Table 33</td>
</tr>
<tr>
<td>JL[1-16]=[value]</td>
<td>Set joystick soft limit settings</td>
<td>OK</td>
</tr>
<tr>
<td>See Table 33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMAXX</td>
<td>Get joystick control maximum voltage level</td>
<td>[0-5000 mV]</td>
</tr>
<tr>
<td>JMAXY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMAXZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMAXU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMAXX=[value]</td>
<td>Set joystick control maximum voltage level</td>
<td>OK</td>
</tr>
<tr>
<td>JMAXY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMAXZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMAXU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMINX</td>
<td>Get joystick control minimum voltage level</td>
<td>[0-5000 mV]</td>
</tr>
<tr>
<td>JMINY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMINZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMINU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMINX=[value]</td>
<td>Set joystick control minimum voltage level</td>
<td>OK</td>
</tr>
<tr>
<td>JMINY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMINZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JMINU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LS</td>
<td>Returns global low speed setting</td>
<td>[1-6,000,000]</td>
</tr>
<tr>
<td>LS=[value]</td>
<td>Sets global low speed</td>
<td>OK</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td>Return</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>LSX</td>
<td>Returns individual low speed setting</td>
<td>[1-6,000,000]</td>
</tr>
<tr>
<td>LSY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSX=[value]</td>
<td>Sets individual low speed</td>
<td>OK</td>
</tr>
<tr>
<td>LSY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTX=[0 or 1]</td>
<td>Enable or disable position latch feature</td>
<td>OK</td>
</tr>
<tr>
<td>LTY=[0 or 1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTZ=[0 or 1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTU=[0 or 1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTXE</td>
<td>Returns latched encoder position</td>
<td>28-bit number</td>
</tr>
<tr>
<td>LTYE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTZE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTUE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTXP</td>
<td>Returns latched pulse position</td>
<td>28-bit number</td>
</tr>
<tr>
<td>LTYP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTZP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTUP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTXS</td>
<td>Returns latch status.</td>
<td>0 – Latch off</td>
</tr>
<tr>
<td>LTYS</td>
<td>1 – Latch on and waiting for latch trigger</td>
<td></td>
</tr>
<tr>
<td>LTZS</td>
<td>2 – Latch triggered</td>
<td></td>
</tr>
<tr>
<td>LTUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MST</td>
<td>Returns all motor status, buffer move status, and move mode status</td>
<td>See Table 24</td>
</tr>
<tr>
<td>PE</td>
<td>Returns current encoder counter values of all 4 axes</td>
<td>[X Enc Position]:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[Y Enc Position]:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[Z Enc Position]:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[U Enc Position]:</td>
</tr>
<tr>
<td>POX</td>
<td>Returns polarity setup</td>
<td>See Table 25</td>
</tr>
<tr>
<td>POY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POX=[value]</td>
<td>Sets polarity</td>
<td>OK</td>
</tr>
<tr>
<td>POY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td>Return</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>PP</td>
<td>Returns current pulse counter values of all 4 axes</td>
<td>[X Pulse Position]; [Y Pulse Position]; [Z Pulse Position]; [U Pulse Position]</td>
</tr>
<tr>
<td>PS</td>
<td>Returns current pulse speed values of all 4 axes</td>
<td>[X Speed]; [Y Speed]; [Z Speed]; [U Speed]</td>
</tr>
<tr>
<td>PX=[value]</td>
<td>Set position value of axis</td>
<td>OK</td>
</tr>
<tr>
<td>PY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SALineNumber</td>
<td>Get standalone line</td>
<td>Single line of compiled code</td>
</tr>
<tr>
<td>SA[LineNumber]=[ Value]</td>
<td>Set standalone line</td>
<td>OK</td>
</tr>
<tr>
<td>SAP</td>
<td>Get the return jump line (standalone error handling)</td>
<td>[0,1]</td>
</tr>
<tr>
<td>SAP=[0 or 1]</td>
<td>Sets the return jump line (standalone error handling)</td>
<td>OK</td>
</tr>
<tr>
<td>SASTAT</td>
<td>Get standalone program status</td>
<td>[0-4]</td>
</tr>
<tr>
<td>0 – Stopped</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 – Running</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 – Paused</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 – In Error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCVX</td>
<td>Returns the s-curve control</td>
<td>[0,1]</td>
</tr>
<tr>
<td>SCVY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCVZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCVU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCVX=[0 or 1]</td>
<td>Enable or disable s-curve. If disabled, trapezoidal acceleration/deceleration will be used.</td>
<td>OK</td>
</tr>
<tr>
<td>SCVY=[0 or 1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCVZ=[0 or 1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCVU=[0 or 1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLAX</td>
<td>Get StepNLoop maximum attempt value of axis</td>
<td></td>
</tr>
<tr>
<td>SLAY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLAZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLAU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLAX=[value]</td>
<td>Set StepNLoop maximum attempt value of axis</td>
<td>OK</td>
</tr>
<tr>
<td>SLAY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLAZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLAU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLEX</td>
<td>Get StepNLoop error range value of axis</td>
<td></td>
</tr>
<tr>
<td>SLEY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLEZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLEU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td>Return</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>SLEX=[value]</td>
<td>Set StepNLoop error range value of axis</td>
<td>OK</td>
</tr>
<tr>
<td>SLEY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLEZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLEU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLRX</td>
<td>Get StepNLoop ratio of axis (ppr / cpr)</td>
<td>[0.001-999.999]</td>
</tr>
<tr>
<td>SLRY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLRZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLRU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLRX=[value]</td>
<td>Set StepNLoop ratio of axis (ppr / cpr)</td>
<td>OK</td>
</tr>
<tr>
<td>SLRY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLRZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLRU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLSX</td>
<td>Get StepNLoop status of axis</td>
<td>[0-12] See Table 35</td>
</tr>
<tr>
<td>SLSY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLSZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLSU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLTX</td>
<td>Get StepNLoop tolerance of axis</td>
<td></td>
</tr>
<tr>
<td>SLTY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLTZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLTU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLTX=[value]</td>
<td>Set StepNLoop tolerance of axis</td>
<td>OK</td>
</tr>
<tr>
<td>SLTY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLTZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLTU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLX</td>
<td>Get StepNLoop enable of axis</td>
<td>[0,1]</td>
</tr>
<tr>
<td>SLY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLX=[value]</td>
<td>Set StepNLoop enable of axis</td>
<td>OK</td>
</tr>
<tr>
<td>SLY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLOAD</td>
<td>Returns the RunOnBoot parameter</td>
<td>[0-15]</td>
</tr>
<tr>
<td>SLOAD=[value]</td>
<td>Set the 4 bit RunOnBoot parameter</td>
<td>OK</td>
</tr>
<tr>
<td>SR[0-3]=[value]</td>
<td>Controls the specified standalone program: 0 – Stop standalone program 1 – Run standalone program 2 – Pause standalone program 3 – Continue standalone program</td>
<td>OK</td>
</tr>
<tr>
<td>SPC[0-3]</td>
<td>Returns the program counter for the specified standalone program</td>
<td>[0-1275]</td>
</tr>
<tr>
<td>SSPDX=[value]</td>
<td>PMX on-the-fly speed change. In order to use this command on a certain axis, S-curve control must be disabled for the corresponding axis. Use SCV[axis] command to enable and disable s-curve acceleration/ deceleration control.</td>
<td>OK</td>
</tr>
<tr>
<td>SSPDY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSPDZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSPDU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSPDMX</td>
<td>Get on-the-fly speed change mode for each axis</td>
<td>[0-7]</td>
</tr>
<tr>
<td>SSPDMY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSPDMZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSPDMU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td>Return</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>SSPDMX=[value]</td>
<td>Set on-the-fly speed change mode for each axis.</td>
<td>OK</td>
</tr>
<tr>
<td>SSPDMY=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSPDMZ=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSPDMU=[value]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOP</td>
<td>Performs ramp down to low speed and stop if the motor is moving. (All axes)</td>
<td>OK</td>
</tr>
<tr>
<td>STOPX</td>
<td>Performs ramp down to low speed and stop if the motor is moving. (Individual axis)</td>
<td>OK</td>
</tr>
<tr>
<td>STOPY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOPZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STORE</td>
<td>Store parameters to flash</td>
<td>OK</td>
</tr>
<tr>
<td>SYNXC</td>
<td>Read sync output configuration for each axis</td>
<td>[1-3]</td>
</tr>
<tr>
<td>SYNVC</td>
<td>1 – trigger when encoder equals position</td>
<td></td>
</tr>
<tr>
<td>SYNZC</td>
<td>2 – trigger when encoder is greater than position</td>
<td></td>
</tr>
<tr>
<td>SYNUC</td>
<td>3 – trigger when encoder is less than position</td>
<td></td>
</tr>
<tr>
<td>SYNXC=</td>
<td>Set sync output configuration for each axis</td>
<td>OK</td>
</tr>
<tr>
<td>SYNYC=</td>
<td>1 – trigger when encoder equals position</td>
<td></td>
</tr>
<tr>
<td>SYNZC=</td>
<td>2 – trigger when encoder is greater than position</td>
<td></td>
</tr>
<tr>
<td>SYNUC=</td>
<td>3 – trigger when encoder is less than position</td>
<td></td>
</tr>
<tr>
<td>SYNXF</td>
<td>Turn of sync output for each axis</td>
<td>OK</td>
</tr>
<tr>
<td>SYNYF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNZF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNUF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNXO</td>
<td>Turn on sync output for each axis</td>
<td>OK</td>
</tr>
<tr>
<td>SYNYO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNZO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNUO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNXP</td>
<td>Get trigger position for each axis</td>
<td>28 bit signed number</td>
</tr>
<tr>
<td>SYNYP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNZP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNUP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNXP=</td>
<td>Set trigger position for each axis</td>
<td>28 bit signed number</td>
</tr>
<tr>
<td>SYNYP=</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNZP=</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNUP=</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNXT</td>
<td>Get pulse width time (ms). Only applicable if sync output configuration is set to 1.</td>
<td>[0-10]</td>
</tr>
<tr>
<td>SYNYT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNZT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNXT=</td>
<td>Set pulse width time (ms). Only applicable if sync output configuration is set to 1.</td>
<td>OK</td>
</tr>
<tr>
<td>SYNYT=</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNZT=</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNUT=</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T[axis] [value]</td>
<td>Set on-the-fly target position change</td>
<td>OK</td>
</tr>
<tr>
<td>TOC</td>
<td>Returns the communication time-out parameter. Value is in milliseconds.</td>
<td>32-bit number</td>
</tr>
<tr>
<td>TOC=[value]</td>
<td>Set the communication time-out parameter.</td>
<td>OK</td>
</tr>
<tr>
<td>TR</td>
<td>Get timer register value</td>
<td>32-bit number</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td>Return</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>TR=[value]</td>
<td>Set timer register value (ms)</td>
<td>OK</td>
</tr>
<tr>
<td>V[0-99]</td>
<td>Get standalone variable value</td>
<td></td>
</tr>
<tr>
<td>V[0-99]=[Value]</td>
<td>Write standalone variable value</td>
<td>OK</td>
</tr>
<tr>
<td>VER</td>
<td>Returns controller firmware version</td>
<td>V[#]</td>
</tr>
<tr>
<td>X[target X]</td>
<td>Individual/interpolation move command</td>
<td>OK</td>
</tr>
<tr>
<td>Y[target Y]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z[target Z]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U[target U]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Error Codes

If an ASCII command cannot be processed by the PMX-4EX-SA, the controller will reply with an error code. See below for possible error responses:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Command]</td>
<td>The ASCII command is not understood by the PMX-4EX-SA</td>
</tr>
<tr>
<td>?ALARM</td>
<td>A move command is sent while the alarm is on.</td>
</tr>
<tr>
<td>?BUFFER FULL</td>
<td>An attempt to add a move to a full buffer has been made.</td>
</tr>
<tr>
<td>?ERRORED</td>
<td>Command has been issued while the controller is in error state.</td>
</tr>
<tr>
<td>?In Incremental mode</td>
<td>A circle or arc interpolation move has been issue while the PMX-4EXSA is in incremental mode.</td>
</tr>
<tr>
<td>?Index out of Range</td>
<td>The index for the command sent to the controller is not valid.</td>
</tr>
<tr>
<td>?LIMIT</td>
<td>A move command is sent while the axis has a limit error.</td>
</tr>
<tr>
<td>?PULSING</td>
<td>A move or position change command is sent while the PMX-4EX-SA is outputting pulses.</td>
</tr>
<tr>
<td>?SSPD Error</td>
<td>An on-the-fly speed change was issued without initialized the SSPDM parameter.</td>
</tr>
<tr>
<td>?Sub not Initialized</td>
<td>Call to a subroutine using the GS command is not valid because the specified subroutine has not been defined.</td>
</tr>
<tr>
<td>?Timer Running</td>
<td>An attempt to set the timer register while it is running has been made.</td>
</tr>
</tbody>
</table>

Table 39 Error Codes
Where to Go Next

The user should now have all the information required to run the DMM-0400. If any further assistance is required, please contact the service department at Dover Motion.
Appendix

About This Chapter

Introduction

This appendix contains supporting information for the DMM-0400.

Topics

This chapter covers the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed Settings</td>
<td>127</td>
</tr>
<tr>
<td>Acceleration/Deceleration Range</td>
<td>127</td>
</tr>
<tr>
<td>Acceleration/Deceleration Range – Positional Move</td>
<td>128</td>
</tr>
<tr>
<td>Review/Revision History</td>
<td>129</td>
</tr>
</tbody>
</table>
## Speed Settings

### Table 40 Speed Settings

<table>
<thead>
<tr>
<th>HSPD value [PPS] †</th>
<th>Speed Window [SSPD value]</th>
<th>Min. LSPD value</th>
<th>Min. ACC [ms]</th>
<th>δ</th>
<th>Max ACC setting [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 65K</td>
<td>0.1</td>
<td>1</td>
<td>2</td>
<td>50</td>
<td>((HSPD − LSPD) / δ) × 1000</td>
</tr>
<tr>
<td>65K - 130K</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>130K - 325K</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>325K - 650 K</td>
<td>4</td>
<td>10</td>
<td>1</td>
<td>800</td>
<td></td>
</tr>
<tr>
<td>650K - 1.3M</td>
<td>5</td>
<td>20</td>
<td>1</td>
<td>1500</td>
<td></td>
</tr>
<tr>
<td>1.3M - 3.2M</td>
<td>6</td>
<td>50</td>
<td>1</td>
<td>3800</td>
<td></td>
</tr>
<tr>
<td>3.2M - 6M</td>
<td>7</td>
<td>100</td>
<td>1</td>
<td>7500</td>
<td></td>
</tr>
</tbody>
</table>

†If StepNLoop is enabled, the [HSPD range] values needs to be transposed from PPS (pulse/sec) to EPS (encoder counts/sec) using the following formula:

EPS = PPS / Step-N-Loop Ratio

### Acceleration/Deceleration Range

The allowable acceleration/deceleration values depend on the LS and HS settings.

The minimum acceleration/deceleration setting for a given high speed and low speed is shown below.

The maximum acceleration/deceleration setting for a given high speed and low speed can be calculated using the formula:

Max ACC = ((HS − LS) / δ) × 1000 [ms]

**Note:** The ACC parameter will be automatically adjusted if the value exceeds the allowable range.

Examples:

a) If HSPD = 20,000 pps, LSPD = 10,000 pps:
   a. Min acceleration allowable: 1 ms
   b. Max acceleration allowable:
      
      
      \((20,000 \text{ – } 10000) / 50 \times 1000 \text{ ms} = 200,000 \text{ ms (200 sec)}\)

b) If HSPD = 900,000 pps, LSPD = 9,000 pps:
   a. Min acceleration allowable: 1 ms
   b. Max acceleration allowable:
      
      \((900,000 \text{ – } 9000) / 1500 \times 1000 \text{ ms} = 594,000 \text{ ms (594 sec)}\)
Acceleration/Deceleration Range – Positional Move

When dealing with positional moves, the controller automatically calculates the appropriate acceleration and deceleration based on the following rules.

1) **ACC vs. DEC 1**: If the theoretical position where the controller begins deceleration is less than L/2, the acceleration value is used for both ramp up and ramp down. This is regardless of the EDEC setting.

2) **ACC vs. DEC 2**: If the theoretical position where the controller begins constant speed is greater than L/2, the acceleration value is used for both ramp up and ramp down. This is regardless of the EDEC setting.

3) **Triangle Profile**: If either (1) or (2) occur, the velocity profile becomes triangle. Maximum speed is reached at L/2.
## Review/Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Summary</th>
<th>ECO Number</th>
<th>Writer / Reviser</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>04/13/12</td>
<td>Initial Release</td>
<td></td>
<td>Phil Li</td>
</tr>
<tr>
<td>B</td>
<td>03/11/13</td>
<td>Removed servo references</td>
<td>S04_101116</td>
<td>Phil Li</td>
</tr>
<tr>
<td>C</td>
<td>06/27/13</td>
<td>Updated rear panel views Switch Silkscreen corrections</td>
<td>S04_101185</td>
<td>Robert Winslow</td>
</tr>
</tbody>
</table>