MMC120 Linear Motion Control Module

and

DCS120-Win Software

User Manual

Version 2.04.1 August 8, 2012

www.deltacompsys.com

Table of Contents

Disclaimer

Although great effort has been taken to ensure the accuracy of the information in this documentation, it is intended to be used only as a guide. Knowledge of motion control, hydraulic servos, electric servos, magnetostrictive displacement transducers, and safety rules is required. Delta Computer Systems, Inc. cannot accept responsibility for problems resulting from omissions in this documentation. The information in this documentation is subject to change without notice.

Neither Delta Computer Systems, Inc. nor anyone else involved in the creation, production, or delivery of this product shall be liable for any direct, indirect, consequential injuries and or damages arising out of the use, the results of use, or the inability to use this product.

All brand names and trademarks referenced in this manual are the property of their respective holders.

Introducing the MMC120

MMC120 Overview

The MMC120 Motion Control Module is a complete two-axis position control subsystem for the Modicon TSX Quantum Automation Series® family of Programmable Controllers. An onboard processor controls the axes, providing complete independent PID motion control loops and allowing on-the-fly motion profile changes. The module has two optically isolated magnetostrictive transducer interfaces and two optically isolated ±10 volt outputs.

The MMC120 occupies one slot of the Quantum rack. The MMC120 and the Programmable Controller communicate over the back plane through four output and four input registers. The MMC120 relieves the Programmable Controller of the overhead needed for servo control. The MMC120 updates the axis position and drive output 1024 times each second, assuring precise positioning even at high speeds.

If more than two axes of control are needed, additional MMC120 modules can be installed.

Features

- ModConnect© Certified
- Quantum Compatible
- Two axes of control
- Powerful tuning/diagnostic software using RS-232 diagnostic port
  See DCS120-Win

- Direct connection to Magnetostrictive Displacement Transducers and proportional/servo valve
- Isolated inputs and outputs
- One millisecond control loop
- Full PID loop control
- Motion profiles can be changed on the fly
- Velocity and Acceleration Feed-Forward
- Deterministic Event Control
  See Event Control
- Synchronization of 2-8 axes
  See Synchronizing Axes.
- FLASH memory for parameter storage
  See FLASH Memory

Applications

- Forest Industry machinery
- Pinch Roller positioning
- Hydraulic actuators
- Palletizers/Stackers
- Laser Positioning
- Tube forging machines
- Cyclic Testing
- Mechanical Animation

Position Transducer Interface

- Resolution to 0.001 inch with a single circulation
- Compatible with Start/Stop and Gated magnetostrictive transducers
- Differential or single ended transducer interface
- Maximum speeds up to 200 in/sec (0.004” resolution)
- Transducer lengths up to 240 inches (0.004” resolution)
- 2500 VAC isolation

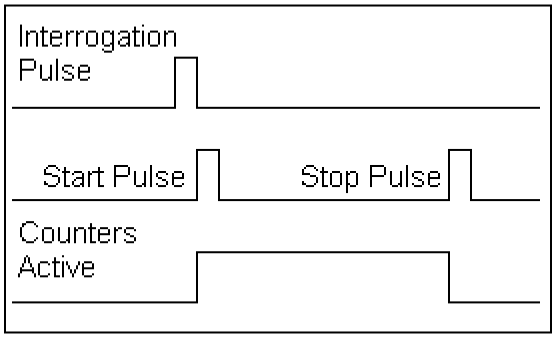
Drive Outputs

- ±10 volts
- 12 bit resolution
- Current output available with optional VC2100 module

Principle of Operation

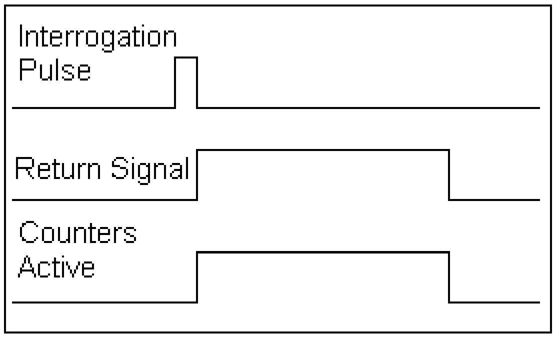
Position Measurement

Each Motion Control Module has interface circuitry for multiple magnetostrictive transducers. Each axis can be configured for a Start/Stop transducer or a Pulse Width Modulated transducer by changing the axis's Configuration Word. To make a measurement with a Start/Stop transducer, the Motion Controller sends an interrogation pulse to the transducer. The transducer responds by returning 2 pulses -- a *Start* pulse and a *Stop* pulse. The counters on the Motion Controller are active between the two pulses. The time between the pulses is proportional to the transducer position.



Start/Stop Pulse Transducer

To make a measurement with a Pulse Width Modulated transducer, the Motion Controller sends an interrogation pulse to the transducer. The transducer responds with a return signal that is high while the transducer is determining its position. The counters on the Motion Controller are active while the return signal is high. The width of the return signal is proportional to the transducer position.

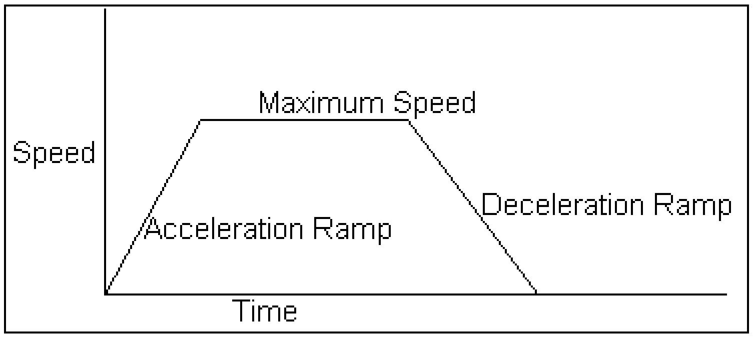


Pulse Width Modulated Transducer

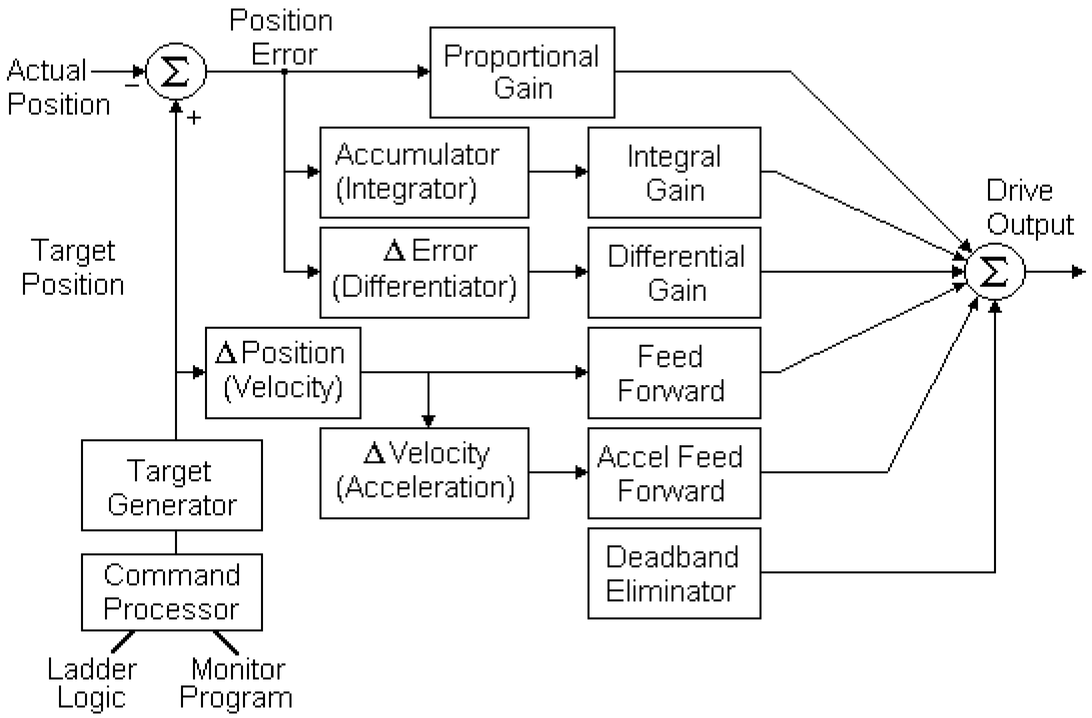
The Motion Controller converts the Transducer Counts read from the counters to an ACTUAL POSITION in user-defined Position Units (usually 0.001 inch) for use by the Programmable Controller.

Control Loop

This motion controller is a targeting controller; each millisecond the onboard microprocessor updates the TARGET POSITION and TARGET SPEED values. For point-to-point moves, TARGET POSITIONS are generated so the target speed follows a profile. The MODE, ACCELERATION, DECELERATION, SPEED, and COMMAND VALUE (requested position) are used to generate the profile. They are specified by the user, and can be changed while the axis is moving. A trapezoidal profile is shown here.



The ACTUAL POSITION measured by the magnetostrictive transducer is compared with the TARGET POSITION to determine the position error. Every millisecond the position error is used to calculate the closed loop components of the drive output. It is multiplied by the PROPORTIONAL GAIN to calculate the proportional component of the drive output. The *accumulated* position error is used, along with the INTEGRAL GAIN, to calculate the integral portion of the drive output. The *change* in position error, along with the DIFFERENTIAL GAIN, is used to calculate the differential portion of the drive output.



In addition to the closed loop drive, this motion controller has two feed forward terms, made up of EXTEND and RETRACT FEED FORWARD, and EXTEND and RETRACT ACCELERATION FEED FORWARD. These feed forward terms give approximately the drive needed to make the axis follow the target, freeing the PID loop to correct for non-linearity in the system and changes in system load.

Drive Output

The drive generated by the motion controller is sent through optical isolation to a 12-bit digital-to-analog converter (DAC). The output from the DAC is amplified to provide a ±10 volt output to the hydraulic valve. Servo valves that need current input require a voltage-to-current converter (Delta part number VC2100). Proportional valves work directly with the voltage signal.

Quantum Interface

The MMC120 communicates with the Quantum controller over the back plane. The module is I/O mapped as a **DCS MMC 120 0x** with four input and four output registers. Commands and status for both axes are transferred across the back plane in groups of four 16-bit words.

Programming

Commands to the Motion Controller are sent by writing to the Programmable Controller's output registers. The first two registers send commands to axis 1 while the next two registers send commands to axis 2. Programming details are presented in Communicating with the MMC120.

Event Control

Sequences of commands can be stored and executed by the Motion Controller with little intervention by the Programmable Controller. This allows a 1-millisecond response time by the Motion Controller to internal events such as move done or elapsed time. This is discussed in Event Control.

FLASH Memory

You can store parameters, profiles, and Event Control steps in the Motion Controller’s non-volatile FLASH Memory. This reduces the memory requirements in the Programmable Controller and eliminates the need to transfer initialization parameters back and forth.

NOTE: Since data is stored in the module, when you replace one module with another you must transfer the parameters and profiles to the new module. Because of this, you must store all parameters and profiles either in the control program or in a monitor program file so they can be transferred to a module when needed.

Starting Up the MMC120

Step-by-Step MMC120 Startup

TIP: Delta’s SSS/10 Servo System Simulator and PPS/14 Position/Pressure

Simulator provide a simple way to test your program before connecting the module to a real system.

1. Connect the MMC120 to the DCS120-Win software

Many of the following steps assume the DCS120-Win software package is used in conjunction with the MMC120 to set up the system. Therefore, one must first connect DCS120-Win with the MMC120 using a serial cable.

- For information on the cable required, see Wiring Notes.
- To change your serial port, see Selecting a Serial Port to Use.

2. Wire the Transducers and Drive Outputs

The MMC120 supports single ended and differential transducers. For more information see Wiring Notes.

Caution: Leave power to the motors and/or hydraulics *disabled* until instructed to power them on later in this procedure.

3. Configure the Transducers

The MMC120 supports a wide range of Magnetostrictive Displacement Transducers. To select the type of the transducer and polarity of the drive output, the following settings can be changed:

- Support for Pulse-Width Modulated or Start-Stop MDTs.
- Support for between 1 and 15 recirculations on PWM MDTs.
- Support for using either edge of a Start-Stop pulse. This is required to support some Balluff transducers.
- Support for reversing the drive output in software.

Each of these settings can be changed using the Configuration word axis parameter; no jumpers are required. You need to set this parameter for each axis used; refer to the Configuration word topic for instructions on using this parameter.

4. Test the Transducer and Drive Connections of Each Axis

Caution: Open loop operation, which this procedure uses, ignores all limits! Be prepared to remove drive power. Great care must be taken to avoid accidents when starting the motion control module for the first time. The most common accident is a runaway, where the motion controller tries to go to a position beyond the physical limits of an axis.

A. Ensure that the hydraulic or motor power is off, and all drive output connectors on the MMC120 are disconnected.

B. Start DCS120-Win and ensure it is **Online** with the MMC120.

C. Connect the MMC120 drive output to the motor or hydraulic valve of the axis being setting up.

D. In DCS120-Win, with the cursor on the axis you want to adjust, enter ALT+R to restore the null. Enter 0 (zero) in the COMMAND VALUE field of the axis, enter ALT+SHIFT+O, and verify that the DRIVE for the axis is 0 (zero). If the NULL DRIVE is not zero, enter ALT+N to clear it.

NOTE: Make sure the Simulate Bit in the Config word is off. Otherwise, Open Loop commands (ALT+SHIFT+O) will not affect output.

E. Turn on power to the motor or hydraulics for the axis being set up (the axis may drift due to valve null errors).

F. Next, we will output 500mV to the axis drive output. Enter 500 in the COMMAND VALUE field, enter ALT+SHIFT+O, then enter 0 (zero) in the COMMAND VALUE field. Verify that the DRIVE for the axis is 500. The axis should extend. If the axis retracts, check the drive wiring polarity, hydraulic plumbing (if applicable), and valve null.

NOTE: The extend direction is defined as the direction in which the transducer counts increase. Watch the Transducer Counts field in DCS120-Win to see that the counts increase. On MDT axes, the extend direction is away from the head of the MDT. The retract direction is opposite from the extend direction.

G. Ensure that the COMMAND VALUE field is still set to 0 (zero), and enter ALT+SHIFT+O, then ALT+P to stop the axis.

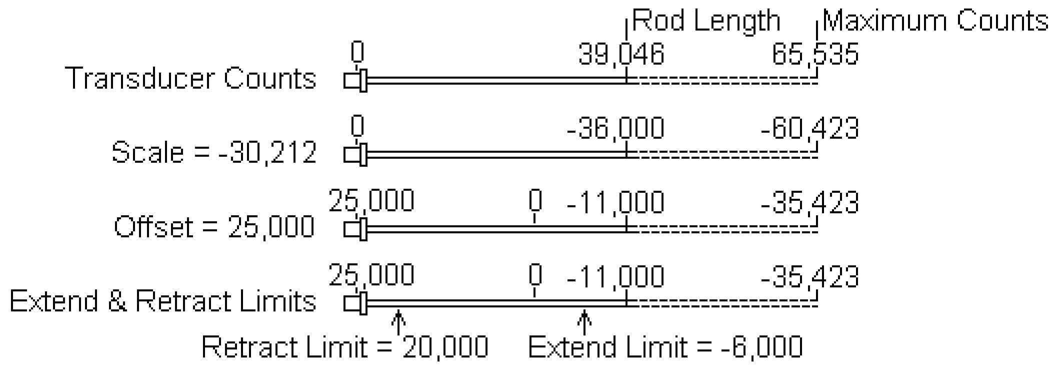
H. Next, we will output -500mV to the axis drive output. Enter -500 in the COMMAND VALUE field, enter ALT+SHIFT+O, then enter 0 (zero) in the COMMAND VALUE field. Verify that the DRIVE for the axis is -500. The axis should retract. If the axis extends, check the drive wiring polarity, hydraulic plumbing (if applicable), and valve null.

I. Ensure that the COMMAND VALUE field is still set to 0 (zero), and enter ALT+SHIFT+O, then ALT+P to stop the axis.

J. Repeat steps C through I for each axis in use.

5. Set the SCALE and OFFSET of Each Axis

The Scale and Offset parameters are used to convert raw transducer counts to user-definable position units. Below is shown an example MDT and the affect of the SCALE, OFFSET, EXTEND and RETRACT LIMITS:



There are two ways to set these fields:

- Use one of the Scale/Offset Calibration utilities. This is the recommended method; it is described in Using the Scale/Offset Calibration Utilities.

- Manually calculate and enter the Scale and Offset parameters. Refer to the Scale and Offset topics for details on calculating these fields.

6. Set the Extend and Retract Limits of Each Axis

The DCS120-Win has software-enforced Extend and Retract Limits. This procedure describes setting these limits by moving the axis to each limit and entering the appropriate value in each field.

Repeat the following steps for each axis:

A. Using one of the following methods, move the axis to the extend limit:
   - Use a control box (diddle box) that can electrically drive the valve or motor.
   - Manually position the motor or cylinder.
   - Use the Drive Test procedure described in step 4 above.

B. In DCS120-Win, enter the value in the Actual Position field for the axis into the axis's Extend Limit parameter.

C. Using any of the above three methods, move the axis to the retract limit.

D. In DCS120-Win, enter the value in the Actual Position field for the axis into the axis's Retract Limit parameter.

7. Tune Each Axis

NOTE: The monitor program is extremely useful; we recommend you use it to configure, tune, and troubleshoot the system. Refer to Using Graphs of Axis Moves for information about creating plots of moves.

The next step is to tune the position control of each axis. For details see Tuning a Position Axis. At this point AUTO STOP should be set to 0xE0E0 so any transducer error on the axis will cause it to stop, but other errors will not. Check the STATUS word for errors after each move.

8. Save Your Configuration Settings

There are several possible places to store the settings to the MMC120 module:

- **FLASH Memory**

  All MMC120 settings can be saved in the MMC120's FLASH by issuing a single command. Refer to FLASH Memory for details.

- **DCS120-Win Disk Files**

  All parameters and tables can be saved to and loaded from disk. The table editors and main screen each has a **Save** command under the **File** menu that can be used to save settings.

- **Programmable Controller Memory**

  It is highly recommended that the MMC120 settings be stored in the Quantum Controllers memory. This allows the MMC120 module to be replaced without losing parameters, provided that the Quantum downloads the settings to the MMC120 on each power-up.

Setup Details

Wiring Notes

Use shielded twisted pairs for all connections to inputs and outputs. Route the transducer wiring separate from other wiring. You must provide the power supplies needed for your transducers.

Drive Outputs

Three Pin Plug-in Terminal Block

Pin	Function
1	Axis 1 Drive
2	Drive Common
3	Axis 2 Drive

When wiring the system, it is important that the drive extends and transducer counts increase when a positive voltage is sent to the drive. The extend direction is defined as the direction that causes the transducer to return increasing counts. The extend direction of a magnetostrictive transducer is away from the head.

CAUTION: If the outputs from the MMC120 are reversed, the axis will be uncontrollable when power is connected. Confirm that your wiring is correct!

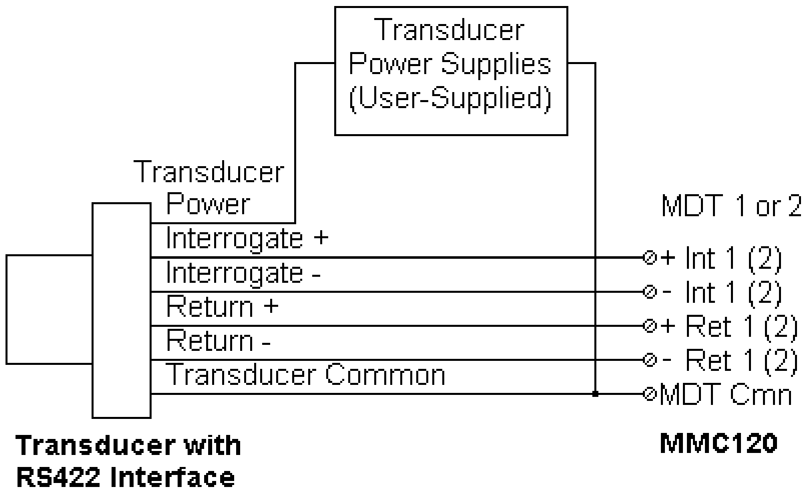
Magnetostrictive Transducer Inputs

Nine-Pin Plug-in Terminal Block

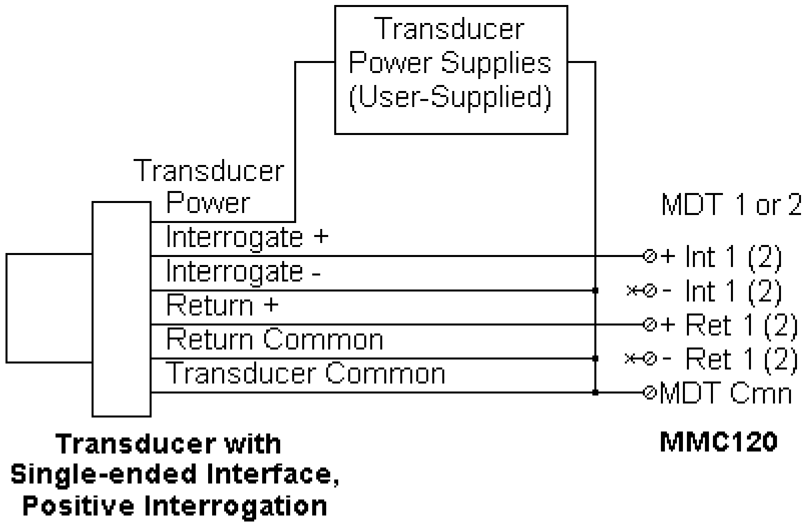
Pin	Function
1	Axis 1 + Interrogation
2	Axis 1 – Interrogation
3	Axis 1 + Return
4	Axis 1 – Return
5	MDT Common
6	Axis 2 + Interrogation
7	Axis 2 – Interrogation
8	Axis 2 + Return
9	Axis 2 – Return

NOTE: The following example schematics do not include transducer pin numbers, color codes, or power supply requirements, since these vary among different transducers. To determine your power supply needs and connector pin-outs or cable color codes, consult your transducer documentation.

The MMC120 family can interface to transducers with either single-ended (TTL) or Differential Line Driver (RS-422) interrogation signals. With RS-422 signals, connect both the '+Int' and '-Int' between the transducer and the MMC120 for the interrogate signal, and connect both the '+Ret' and '-Ret' between the transducer and the MMC120 for the return signal. Connect the transducer DC ground to MDT Common.

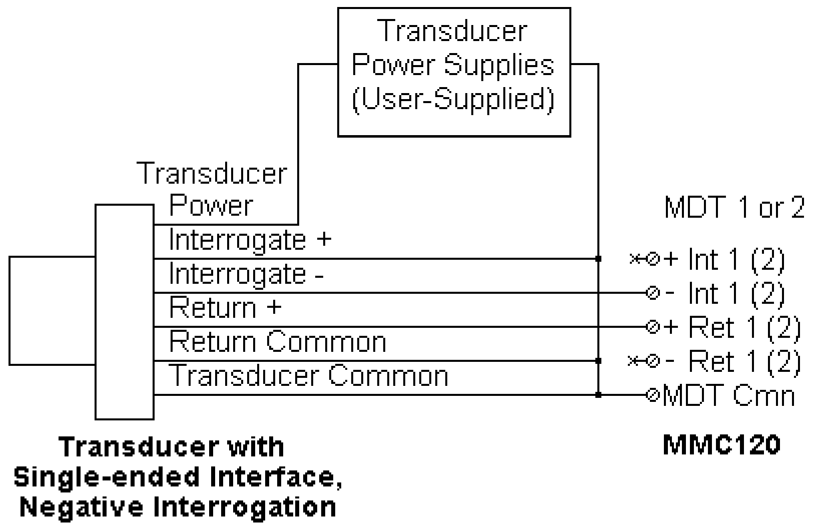


For single-ended transducer with positive interrogation, connect the transducer '-interrogation in' wire to the ‘MDT Cmn’ pin and the transducer '+ interrogation in' wire to the '+ Int' pin. CONNECT NOTHING TO THE '-Int' PIN OF THE MMC120. Connect the transducer return plus wire to the '+Ret’ pin on the MMC120 and the transducer return common wire to ‘MDT Cmn’ on the MMC120. CONNECT NOTHING TO THE '-Ret' PIN OF THE MMC120.



Temposonics I transducer users:

For the negative interrogation version of this transducer, connect the transducer '+ interrogation in' wire to the ‘MDT Cmn’ pin and the transducer '- interrogation in' wire to the '-Int' pin. CONNECT NOTHING TO THE '+Int' PIN OF THE MMC120. Connect the transducer return plus wire to the '+Ret’ pin on the MMC120 and the transducer return common wire to ‘MDT Cmn’ on the MMC120. CONNECT NOTHING TO THE '-Ret' PIN OF THE MMC120.



Some Temposonics I transducers from MTS have 200 Ohm termination resistors installed between their interrogation pins and common. If yours do not, it may be necessary to install them as close to the transducers as possible to reduce electrical noise in the system.

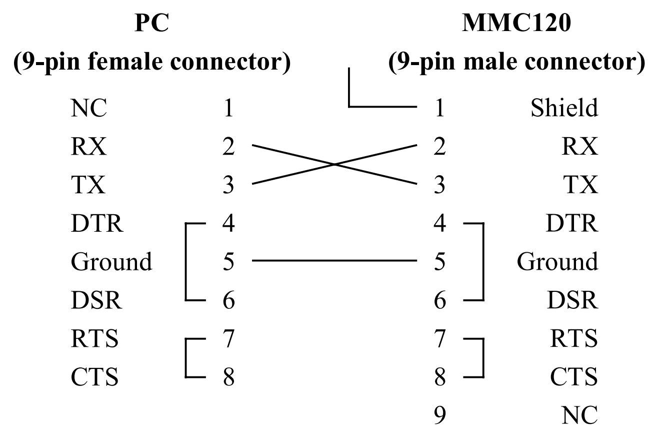
Serial Port

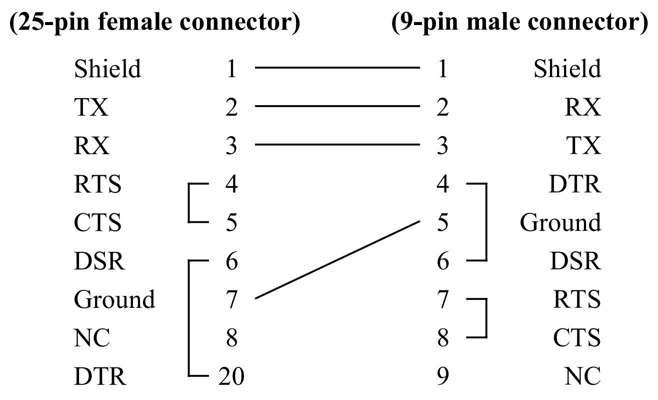
DB-9

Pin	Function
2	Receive
3	Transmit
5	Common

The communication cable attached to the serial port is a potential source of electromagnetic radiation from the MMC120. To minimize radiation, use a well-shielded cable that is as short as possible, and route it out the bottom of the module and against the back panel.

A standard Modicon Modbus cable will work in this port, or one can be built using the following diagram:





Tuning

There is no substitute for experience when tuning an axis. This section offers some guidelines, tips, and suggestions for tuning your system. While these steps will work for many systems, they may not be the best for a particular system.

In many hydraulic systems the feed forward parameters (EXTEND FEED FORWARD and RETRACT FEED FORWARD) are the most important parameters for position tracking during a move. One way to adjust these parameters is to set the DIFFERENTIAL GAIN and INTEGRAL GAIN to zero and the PROPORTIONAL GAIN to a small value (between 1 and 5), then make long slow moves in both directions. Adjust the EXTEND FEED FORWARD and RETRACT FEED FORWARD until the axis tracks within 10% in both directions. In hydraulic systems, the EXTEND and RETRACT FEED FORWARD terms will differ by the ratio of the extend and retract piston areas.

Alternately, you can find the appropriate value for the FEED FORWARD terms by making moves with the axis at a SPEED of 10,000 or 10 inches/second. The amount of output drive required to maintain this SPEED is the correct value for the FEED FORWARD parameter.

A third approach is to use the Set Feed Forward command. This command, used after a move without oscillation or overdrive on an axis, will automatically adjust the FEED FORWARD parameter for the direction of that move.

PROPORTIONAL GAIN affects the responsiveness of the system. Low gains make the system sluggish and unresponsive. Gains that are too high make the axis oscillate or vibrate. You can adjust the PROPORTIONAL GAIN by slowly increasing it and moving the axis. When you see a tendency to oscillate as the axis moves or stops, reduce the gain by 10 to 30 percent.

Many hydraulic systems do not require INTEGRAL GAIN or DIFFERENTIAL GAIN. However, it is usually desirable to have some INTEGRAL GAIN (5 to 50 counts) to help compensate for valve null drift or changes in system dynamics. Some systems may require larger INTEGRAL GAIN, in particular if they are moving a large mass or are nonlinear. Too much INTEGRAL GAIN will cause oscillations.

DIFFERENTIAL GAIN is used mainly on systems that have a tendency to oscillate. This happens when heavy loads are moved with relatively small cylinders. DIFFERENTIAL GAIN will tend to dampen out oscillations and help the axis track during acceleration and deceleration. If you use DIFFERENTIAL GAIN, you may be able to increase the PROPORTIONAL GAIN somewhat without causing the system to oscillate.

A disadvantage to DIFFERENTIAL GAIN is that it amplifies position measurement noise which can cause the system to chatter or oscillate if the gain is too high or there is too much noise.

The ACCELERATION FEED FORWARD terms are particularly useful for axes which move large masses with relatively small cylinders. This combination delays the start of movement, and the ACCELERATION FEED FORWARD terms can help compensate for this delay. ACCELERATION FEED FORWARDS are easiest to adjust with the PID gains set low and the VELOCITY FEED FORWARDS adjusted properly. After commanding a move, plot the move using the monitor program and look for a following error during the acceleration. Increase the ACCELERATION FEED FORWARD until the error disappears. For large masses the ACCELERATION FEED FORWARD can be in the tens of thousands.

If the axis hunts around the set point, you can increase the DEAD BAND ELIMINATOR value slowly until the hunting stops or the axis starts to oscillate. If it oscillates, reduce the DEAD BAND ELIMINATOR value.

If the axis gets no following errors, reduce the FOLLOWING ERROR until errors start to occur then adjust the FEED FORWARD gains.

Increase the SPEED and ACCELERATION values gradually while making long moves. Use the monitor program to plot the moves and look for following errors, overshoot, or oscillations. Eventually, when the SPEED and ACCELERATIONS are too high, the moves will cause an error on the axis.

If an overdrive error occurs, there is not enough drive capacity to drive the axis at the requested SPEED. Should this occur, reduce the SPEED. If a following error occurs, the appropriate FEED FORWARD must be increased. If the FOLLOWING ERROR occurs on an extend move, increase the EXTEND FEED FORWARD; it the error occurs on a retract move, increase the RETRACT FEED FORWARD. If this doesn't solve the problem, the ACCELERATION and DECELERATION ramps are too steep for the response of the system. Their values can be reduced, or the ACCEL FEED FORWARD terms can be increased. After the problem which caused the error has been corrected, keep moving the axis back and forth with increasing speed until you reach the desired speed. Should the system seem a little sloppy, try decreasing the FOLLOWING ERROR parameter and adjusting the PROPORTIONAL GAIN until the axis can be moved without getting an error.

Remember: the parameters are not updated in the motion controller until the Set Parameters command is issued. They are not stored into the motion controller FLASH memory until the Update FLASH command is issued.

Jogging the Axis

You can jog the axis by setting the COMMAND VALUE to the EXTEND LIMIT or RETRACT LIMIT and using the Go and Halt commands repeatedly. This causes the Motion Control Module to Go and Halt.

Saving Parameters and Profiles

After the system is set up and tuned, you need to store the parameter and profile values in the Programmable Controller. This is done using the Get Parameter and Get Profile commands. Only the parameters for the axes used need to be saved, and only the profiles used need to be saved.

Using DCS120-Win

DCS120-Win Overview

Description

DCS120-Win is a Windows 98/NT/2000/XP/Vista/7 based software package that allows you to access, display, troubleshoot, configure and control features of Delta's motion control products. DCS120-Win allows you to adjust the MMC120's parameters and make simple movements. You can display a motion trajectory using DCS120-Win's graphing capability.

Basic Topics

- Understanding the Screen
- Changing Data from the Keyboard
- Accessing Context Sensitive Help
- Read-back versus Write Mode
- Using Popup Editors
- Using the STATUS Bits Window
- Using the Command Log
- Using the Parameter Error List Window

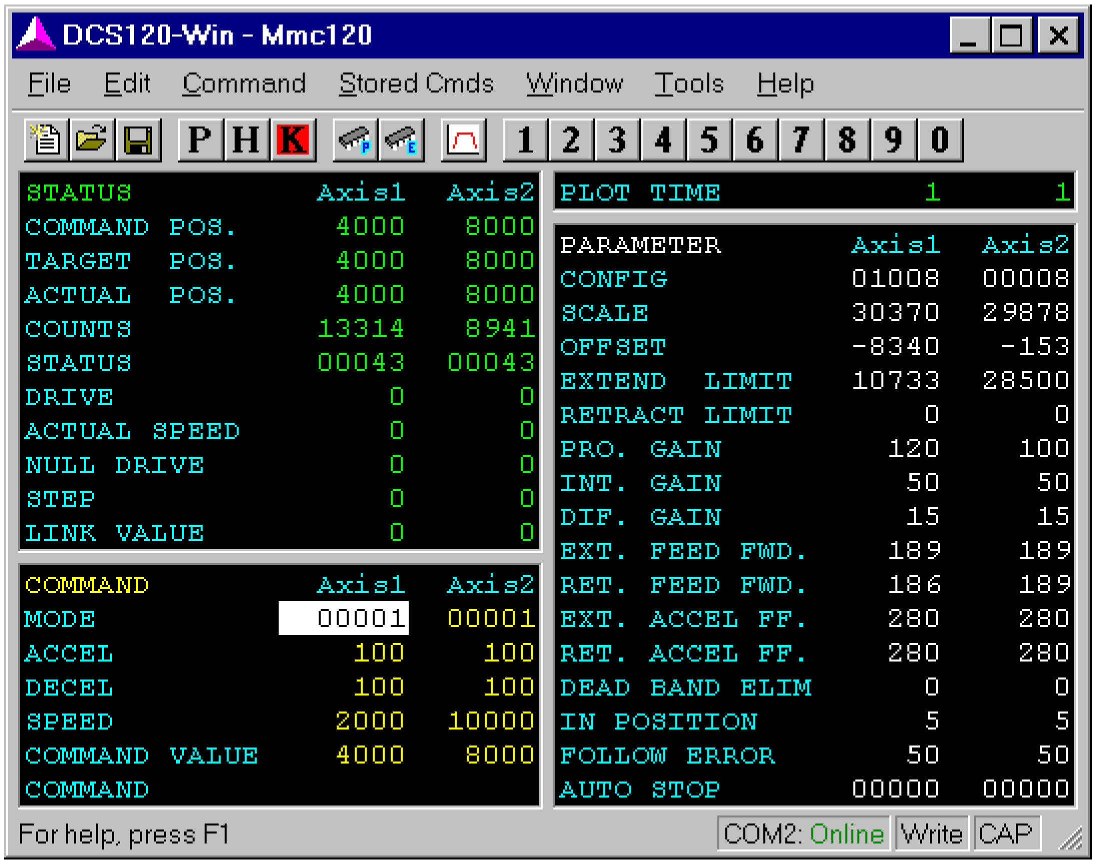
Screen Layout

Understanding the Screen

The main window in this program display several kinds of information. It can be divided into six sections:

- Toolbar
- Status Bar
- Status area (top-left pane)
- Command area (bottom-left pane)
- Plot Time area (top-right pane)
- Parameter area (bottom-right pane)

Here is a sample of the main screen:


DCS120-Win - Mmc120
File Edit Command Stored Cmds Window Tools Help
P H K 1 2 3 4 5 6 7 8 9 0
For help, press F1
COM2: Online
Write
CAP

```
STATUS            Axis1   Axis2    PLOT TIME              1       1
COMMAND POS.       4000    8000
TARGET  POS.       4000    8000    PARAMETER          Axis1   Axis2
ACTUAL  POS.       4000    8000    CONFIG             01008   00008
COUNTS            13314    8941    SCALE              30370   29878
STATUS            00043   00043    OFFSET             -8340    -153
DRIVE                 0       0    EXTEND  LIMIT      10733   28500
ACTUAL SPEED          0       0    RETRACT LIMIT          0       0
NULL DRIVE            0       0    PRO. GAIN            120     100
STEP                  0       0    INT. GAIN             50      50
LINK VALUE            0       0    DIF. GAIN             15      15
                                   EXT. FEED FWD.       189     189
COMMAND           Axis1   Axis2    RET. FEED FWD.       186     189
MODE              00001   00001    EXT. ACCEL FF.       280     280
ACCEL               100     100    RET. ACCEL FF.       280     280
DECEL               100     100    DEAD BAND ELIM         0       0
SPEED              2000   10000    IN POSITION            5       5
COMMAND VALUE      4000    8000    FOLLOW ERROR          50      50
COMMAND                            AUTO STOP          00000   00000
```

Command area

This area is located in the lower left portion of the main window. It holds the Command fields for each axis. This area is updated only in Read-back Mode. Refer to the following sections for details on the command fields:

- MODE
- ACCELERATION
- DECELERATION
- SPEED
- COMMAND VALUE
- COMMAND

To the right of each of the Command field labels are the values for each axis's command fields. You can change these values using the keyboard as described in Changing Data from the Keyboard. Changes are not sent to the motion control module until the COMMAND field itself is changed. At this time, all six command fields are sent to the controller.

For details on saving and loading commands, see Changing Between Board Files.

Parameter area

This area is located in the lower right portion of the main window. It holds the Parameter fields for each axis. This area is updated only in Read-back Mode. Refer to the following sections for details on the parameter fields:

CONFIGURATION Word
SCALE
OFFSET
EXTEND LIMIT
RETRACT LIMIT
PROPORTIONAL GAIN
INTEGRAL GAIN
DIFFERENTIAL GAIN
EXTEND FEED FORWARD
RETRACT FEED FORWARD
EXTEND ACCELERATION FEED FORWARD
RETRACT ACCELERATION FEED FORWARD
DEAD BAND ELIMINATOR
IN POSITION
FOLLOWING ERROR
AUTO STOP

To the right of each of the Parameter field labels are the values for each axis's parameter fields. You can change these values using the keyboard as described in Changing Data from the Keyboard. Changes are sent to the motion control module only when a Set Parameters Command is issued. There are several ways to send this command:

- Select a field in the axis you want initialized and press ALT+P.
- Select a field in the axis you want initialized and from the **Command** menu, click **P-Set Parameters**.
- Select the COMMAND field in the axis you want initialized, type P, and then press ENTER.

NOTE: The color of the Parameter values is significant. If an axis's parameters are displayed in WHITE, then no parameters have been changed since the last time they were sent to the motion controller, otherwise they are displayed in RED. This is done to remind the user to send a Set Parameters Command after updating a parameter. After this command is issued to an axis, the parameters for that axis will be displayed in WHITE.

NOTE: When in Read-back Mode, you will notice that RED parameters will be replaced with WHITE parameters as the current values are read from the motion control module. This is done to indicate that the values displayed match those used by the motion controller.

For details on saving and loading parameters, see Changing Between Board Files.

Plot Time area

This area is located in the top right portion of the main window. It holds the PLOT TIME field for each axis. Refer to this topic for details on its use. This area is updated on startup and each time that a new module is connected to the serial port.

To the right of the “PLOT TIME” heading is a column for each axis. You can change these values using the keyboard as described in Changing Data from the Keyboard. Changes are sent to the motion control module immediately.

Status area

This area is located in the upper left portion of the main window. It holds the Status fields (also called Read-back Parameters) for each axis. This area is updated constantly when a motion control module is connected to the program. Refer to the following sections for details on the Read-back Parameters:

- COMMAND POSITION
- TARGET POSITION
- ACTUAL POSITION
- TRANSDUCER COUNTS
- AXIS STATUS Word
- DRIVE
- ACTUAL SPEED
- NULL DRIVE
- STEP
- LINK VALUE

To the right of each of the Status field labels are the values for each axis’s command fields. These values cannot be changed.

Status Bar

The status bar is located at the bottom of the main screen. This bar is divided into four areas:

Menu Help - All of the status bar except the three panes described below is used to display help on menu items. When no menu item is selected, it displays "For help, press F1." If a menu item is selected, or the cursor is over a toolbar button, then a brief line of help is displayed here.

COMx - This pane displays the current serial port in use and the state of the serial port. This pane responds to double-clicks and right-clicks; see Selecting a Serial Port to Use for details.

The serial port can be in one of the following states:

Online - This indicates that the monitor program is currently communicating with a motion control module.

Offline - This indicates that the monitor program has not detected a motion control module. The serial port is currently being used to detect when a motion controller is connected.

Closed - This indicates that the monitor program is not using a serial port at all.

Loader - This indicates that serial connection is working, but the motion control module does not have a valid firmware program and is therefore waiting for firmware to be downloaded to it. Refer to Downloading New Firmware for details on updating the firmware.

Read/Write - This pane indicates whether the Command and Parameter areas of the main display are in read-back or write mode. Double-clicking this pane will toggle between Read-back and Write modes. Refer to Read-back versus Write Mode for a details on these modes.

CAP - This pane indicates whether the CAPS LOCK key is toggled on or off. If this pane is blank, then CAPS LOCK is not enabled. Otherwise, it will display CAP.

Toolbar

The follow buttons are available on the toolbar:

New	This creates a new board file with default parameters. Refer to Using Multiple Motion Modules for details on board files.
Open	This opens a different board file. Refer to Using Multiple Motion Modules for details on board files.
Save	This saves the current board file. Refer to Using Multiple Motion Modules for details on board files.
Set Parameters	This sends the parameters to the board for the current axis, and issues a Set Parameters command.
Halt	This issues a Halt command to the current axis.
Kill	This issues a Disable Drive Output command to all axes.
Plot	This displays the plot for the current axis.
Save Parameters and Profiles to FLASH	This issues an Update FLASH command with a 1 command value, which saves all axes' parameters and profiles to the FLASH.
Save Events to FLASH	This issues an Update FLASH command with a 2 command value, which saves the

		event step table to the FLASH.
1 … 0	**Stored Commands**	Each of these buttons issues a stored command to the current axis. This is equivalent to holding CTRL and pressing a number key. Also, if the user holds down the ALT key while pressing one of these buttons, the full profile stored command is executed. This is equivalent to holding ALT and pressing a number key. See Using Stored Commands for further details.

Current Axis

On the main screen window, there is always one field that is highlighted. The axis that this selected field is under is the current axis.

Basic Topics

Changing Data from the Keyboard

The data in the COMMANDS, PLOT TIME and PARAMETER sections may be changed from the keyboard. Additionally the data in any of the table editors can be modified from the keyboard.

To enter values from the keyboard you must first select one or more cells. Selected cells are highlighted.

To select a single cell from the keyboard:

1. Use the arrow keys to move the selected cell around.

To select multiple cells.

1. Press and hold SHIFT at the first cell to be selected.
2. Use the arrow keys to change the last cell to be selected.
3. Release SHIFT.

Once one or more cells are selected, simply type in the value you want using one of the following formats:

- To enter decimal numbers, simply type in the value, without a leading zero.
- To enter hexadecimal numbers, type a leading zero, followed by the hexadecimal digits. For example, **0FFE0**.
- To enter an ASCII command, type the letter of the command. Notice that this works

only in the COMMAND field. For example, **W**.

Press ENTER to finalize your changes, or ESC or cancel your edits.

NOTE: In the fields that are displayed values in hexadecimal, you can use Popup Editors to edit the data. This is the easiest way to ensure that these words are modified correctly.

Commands can also be issued using shortcut keys. To learn the command shortcut keys, click the **Command** menu, and look at the right column of the menu. For example, ALT+P will issue the **P – Set Parameters** command. Commands can also be issued using Stored Commands; see Using Stored Commands for more information.

Data may also be copied around the main screen. The keys used for doing so are identical to most spreadsheets.

- To cut cells to the clipboard, press CTRL+X.
  This key is available on in the table editors, and not on the main window.
- To copy cells to the clipboard, press CTRL+C.
- To paste from the clipboard to the current location, press CTRL+V. You should not have a group of cells selected when you paste, just have the cursor in the upper-leftmost cell to which you want the block to be copied.

Accessing Context Sensitive Help

Each of the fields displayed on the main screen and the table editors can have help associated them.

To display the context sensitive help for a field:

- Right-click on the field for which you want help to display the shortcut menu, and click **Help on *field***.
- Or, select the cell of the field for which you want help, and press F1.

Read-back versus Write Mode

The Command and Parameter areas of the main screen can operate in either of the following two modes:

Read-back Mode - In this mode, the Command and Parameter areas will be continually read from the motion control module. This mode is necessary to monitor the commands given from another source (such as the PLC) and also to determine the parameters stored on the motion control module. The Command area field values are displayed in RED.

NOTE: Because the Command and Parameter fields are constantly being updated it is possible to have changes you are making overwritten by a field update. To avoid this in most circumstances, the motion program will not update the current axis in the current area (Command or Parameter).

Write Mode -In this mode, the Command and Parameter areas are never automatically updated. You can freely change values without them being overwritten by automatic updates. The Command area field values are displayed in YELLOW.

To switch between Read-back and Write modes, use one of the following methods:

- On the **Tools** menu, click **Toggle to readback/write mode**.
- Press CTRL+T from the main screen.
- Double-click the status bar pane that says either **Read** or **Write**.

On startup, the monitor program uses the last mode used the last time the monitor program was ran.

Using Popup Editors

Popup editors are dialog boxes that simplify editing fields in the motion controller that would otherwise be confusing to edit by hand.

There are popup editors for each of the following parameters: Mode, Configuration Word, and Auto Stop. These values are normally displayed in hexadecimal, but by using the popup editor, editing these fields becomes intuitive.

There is also a popup editor for editing the Link Type and Value in the Event Step table editor. This simplifies Event Step table programming by displaying all possible link types and values.

There are three ways to start a popup editor:

- Right-click on a cell you want to modify, and click **Popup Editor for *field*** from the shortcut menu, where ***field*** is the name of the field.
- Double-click on the cell you want to modify.
- Select the cell you want to modify, and then press ENTER.

NOTE: The Status read-back field also has a window that can be accessed in the above three ways, but this window is a read-only window that displays the current status bits. It can be cleared by pressing ESC or closing the window.

Using the STATUS Bits Window

The Status Bits window displays the bits of the STATUS words for each axis. It is constantly updated as the bits change in the motion controller.

To display the Status Bits window, do one of the following from the main window:

•On the **Window** menu, click **Status Bits**.

- Press CTRL+B.

Additionally, the Status Bits window can be displayed by using any of the methods of displaying a Popup Editor.

Using the Command Log

Command Log Explained

For debugging problems with a system using the motion controller, it is often difficult to determine if the problem is caused by something the motion controller is doing or with the Programmable Controller (P/C). To help with this problem, the Command Log is available. The Command Log will hold the last 256 commands received from the P/C. Remember that commands sent through the monitor program are not displayed here. Therefore, you can use the Command Log to determine which commands were actually received by the motion controller.

The "To PLC from Module" Section

This section displays status information for each axis that is available to the P/C. For each axis there are two pieces of information:

Status -	This displays the STATUS word of the axis.
Data -	The value of this field depends on the Status Area Request field of the last command sent from the P/C on this axis. It can be equal to any of the Read-back Parameters.

The "From PLC to Module" Section

These fields represent the commands and command values received from the P/C.

The most recent commands are at the top of the Command Log. In addition, you will notice that the commands that changed from the previous Command Log entry are colored yellow to aid in spotting which commands changed.

Opening and Closing the Command Log Window

To open the Command Log window, do one of the following from the main window:

- On the **Window** menu, click **Command Log**.
- Press CTRL+L.

To close the Command Log window, do one of the following:

- Press ESC.
- On the **File** menu, click **Exit**.
- Click the **Close** button.

Pause/Resuming the Data Flow

In some applications the Command Log may be scrolling continually.

To freeze the flow of commands in the log, do one of the following:

- On the **File** menu, click **Pause Log**.
- Press P while in the Command Log screen.
- Click **Pause** (⏸) from the toolbar.

To resume the flow of commands in the log, do one of the following:

- On the **File** menu, click **Update Log**.
- Press ENTER while in the Command Log screen.
- Click **Resume Update** (▶) from the toolbar.

Scrolling in the Command Log

The scroll bars may be used to scroll through the Command Log. Scrolling up will show newer data, and scrolling down will show older data. In addition, the UP ARROW, DOWN ARROW, PAGE UP, PAGE DOWN, HOME, and END keys can be used to scroll.

Saving the Command Log

You can save the command log for later reference. The file is stored in text format. It can be opened later both in DCS120-Win or a text editor. The default file extension is .log.

To save a command log:

1. On the **File** menu, click **Save**.
2. In the **File name** box, enter the name of the file.
3. Click **Save**.

NOTE: As soon as the **Save** command is clicked, the Command Log is automatically paused. After saving the file, the title bar will display the filename. To return to the current Command Log, click **Resume Update**.

Opening a Command Log

You can view command logs that were previously saved.

To save a command log:

1.On the **File** menu, click **Open**.

2. In the **File name** box, enter the name of the file.
3. Click **Open**.

When a command log is opened, the command log window will stop updating and just display the opened log. To resume displaying the current log, click **Resume Update**.

Changing the Command Log Properties

The command log font size and bold properties can be changed using the **Properties** dialog box. Changes made to these properties are automatically saved from session to session.

To open the **Properties** dialog box:

1. On the **File** menu, click **Properties**.
2. In the **Command Log Properties** dialog box, select your font size and check whether or not you want the normal and/or changed cells to use the bold version of the font.
3. To try the changes without closing the dialog box, click **Apply**.
4. To use the changes and close the dialog box, click **OK**.
5. To close the dialog box without any changes, click **Cancel**.

Using the Parameter Error List Window

Each axis's Status word has a bit called Parameter Error. This bit is set when a problem related to user-issued commands is encountered; there are dozens of specific problems that can lead to this bit being set.

Therefore, it is important to be able to identify which of the specific parameter errors caused the bit to be set. This is simplified by using the Parameter Error List Window.

To start the Parameter Error List window:

- On the **Windows** menu, click **Parameter Error List**.

The window that is displayed displays all parameter errors that have been captured by DCS120-Win since DCS120-Win was started. The axis each error occurred on and a short description of the error is listed in this dialog box. To receive more in-depth help on a particular error do one of the following:

- Double-click on the error in the **Most Recent Parameter Errors** list.
- Click the error in the **Most Recent Parameter Errors** list, and then click **Help on Error**.
- Click the error in the **Most Recent Parameter Errors** list, and then press F1.

The list of errors in the Parameter Errors dialog box is built while DCS120-Win is connected to the motion controller. The program polls the motion controller frequently for status information, including current parameter error. When a new parameter occurs on an axis, it is added to the list. Because this list is maintained within DCS120-Win and not in the motion controller itself, restarting DCS120-Win will erase the list. Similarly, pressing the **Clear List** button will clear the list, but not affect the motion controller in any way. It is also important to understand that because DCS120-Win polls to find out what errors have occurred, it is possible for an error to occur that is then cleared quickly by another valid command, and therefore DCS120-Win will miss capturing the error in its list.

Using Stored Commands

When setting up and tuning the axes, it is usually necessary to repetitively move the axes between two or more positions. For this reason, the monitor program stores 10 motion profiles for each axis; these are called stored commands. For details on setting up these profiles, see Editing the Stored Command Table. These stored commands can be used in either partial profile or full motion profile modes.

When executed in partial profile mode, only the COMMAND and COMMAND VALUE fields of the stored command are copied into the command fields of the current axis. Therefore, the MODE, ACCEL, DECEL and SPEED remain the same as before the move was requested. To use a stored command as a partial profile use any of these methods:

- Hold down CTRL and press the number of the stored command you wish to execute: 0 to 9. (e.g. CTRL+2 uses the partial profile of stored command 2).
- On the **Stored Cmds** menu, click the move you want to execute.
- Click the button of the desired stored command on the Toolbar.

When executed in full motion profile mode, all six of the command fields are copied from the stored command to the command fields of the current axis. To use a stored command as a full motion profile, use any of these methods.

- Hold down ALT and press the number of the stored command you wish to execute: 0 to 9. (e.g. ALT+2 uses the full motion profile of stored command 2).
- On the **Stored Cmds** menu, point to **Full Motion Profile**, and then click the move you want to execute.
- Click the button of the desired stored command on the Toolbar, *while* holding down the ALT key.

Setup Options

Selecting a Serial Port to Use

There are three methods to change the serial port settings:

- On the **Tools** menu, click **Options**, and then click the **Serial Port** tab. For a description of the settings in this dialog box, see Options Dialog Box.
- Double-click the **COMx** pane of the status bar. This method, too, brings up the **Serial Port** tab of the **Options** dialog box.
- Right-click on the **COMx** pane of the status bar to display the shortcut menu. From this menu, you can display the **Serial Port** tab of the **Options** dialog box or change the serial port or open/closed status of the port directly.

For details on the serial cable required, refer to Wiring Notes.

If you are using multiple motion control modules, then see the Using Multiple Motion Modules topic for details on using different serial ports for different motion control modules.

Changing the Axis Name

The names of the axes are displayed throughout the monitor program. They are used only for display purposes. An axis name can have no more than five characters.

To change axis names:

1. On the **Tools** menu, click **Options**, and then click the **Axis Names** tab.
2. Edit the axis names.
3. Click **OK**.

To edit an individual axis name:

1. On the main screen, double-click the axis name.
2. Edit the axis name.
3. Click **OK**.

The axis names are attached to a single board file. If you are using multiple motion control modules, then see the Using Multiple Motion Modules topic for details on using board files to keep track of multiple motion control modules.

Using Multiple Motion Modules

This monitor program can keep track of several modules. Associated with each module are the following pieces of information:

- Module name (the filename)
- Names of each axis
- Parameters of each axis (configuration word, scale, offset, etc.)
- Command fields for each axis, except the Command itself (Mode through Command Value).
- Plot times for each axis.
- Serial port to be used

A board file (.bd1) stores all of the above with the exception of the serial port. The serial port is remembered for the board file in Windows' internal Registry. Therefore, different computers can use a different serial port for the same board file.

See the following related topics:

- Creating a New Board File
- Editing Board File Information
- Changing Between Board Files

Creating a New Board File

Board files are used to store the following pieces of information:

- Names of each axis
- Parameters of each axis (configuration word, scale, offset, etc.)
- Command fields for each axis, except the Command itself (Mode through Command Value).
- Plot times for each axis.

Refer to Using Multiple Motion Modules for general board file information.

To create a new board file:

1. On the **File** menu, click **New**.

The resulting board file will have the following characteristics:

- It will use the same serial port that is currently opened.
- It will use the currently selected communication configuration.
- The parameters and commands will be reset to the defaults.
- The plot times will be set to the minimum.
- The axis names will be set to the default axis names.

Changing Between Board Files

Board files are used to store the following pieces of information:

- Names of each axis
- Parameters of each axis (configuration word, scale, offset, etc.)
- Command fields for each axis, except the Command itself (Mode through Command Value).
- Plot times for each axis.

Refer to Using Multiple Motion Modules for general board file information.

To save a board file:

1. On the **File** menu, click **Save As**.
2. In the **File name** box, enter the name of the file.
3. Click **Save**.

To open a board file:

1. On the **File** menu, click **Open**.
2. In the **File name** box, enter the name of the file.
3. Click **Open**. The file will be loaded. If the new board file uses a different serial port, the previous serial port will be closed and the new serial port will be opened.

Editing Board File Information

The following pieces of information associated with a board file can be changed in the manner described below:

Axis Names -	Refer to the Changing the Axis Name topic for details.
Parameters -	Refer to the Parameter area topic for details.
Plot Times -	Refer to the Plot Time area topic for details.

Default Commands -	Refer to the Command area topic for details.
Serial Port -	Although the serial port is not actually stored in the board file itself, it is remembered for every board file in the Windows Registry. Refer to the Selecting a Serial Port to Use topic for details.

Scale/Offset Calibration Utilities

Using the Scale/Offset Calibration Utilities

The Scale/Offset Calibration utility provides an easy way to set the Motion Controller's Scale parameter, the Offset parameter, and Prescale Divisor bits of the Configuration parameter.

There are actually several Scale/Offset Calibration Utilities:

- Position Scale/Offset Calibration Utility
- MDT Scale/Offset Calibration Utility

Position Scale/Offset Calibration Utility

For a description of all Scale/Offset Calibration Utilities, see Using the Scale/Offset Calibration Utilities. To use the general position Scale/Offset Calibration Utility:

1. Choose two axis positions to use for this calculation. These points should be significantly far apart to minimize errors. Generally, the extend and retract limits are fine points to choose.
2. Place the cursor in a field under the position axis you wish to calibrate.
3. On the **Tools** menu, click **Scale/Offset Calibration**.
4. Move the axis to the first point. This can be done by jogging the axis manually, or by using the Open Loop command.
5. Measure the physical distance to the first point on the axis in the position units that you intend to use (for example, thousandths of inches, millimeters).
6. Under **First position**, in the **Actual position** box, type the distance measured to the first point in position units.
7. Under **First position**, click **Use Current**, which copies the COUNTS on this axis being calibrated to the **Counts** box under **First position**. You can also manually type a value in this box, but it is easiest to use the **Use Current** button.
8. Move the axis to the second point. You may want to move the **Scale/Offset Calibration** dialog box out of the way so that you can use the main monitor window.
9. Measure the physical distance to the second point on the axis in the position units that you intend to use.
10. Under **Second position**, in the **Actual position** box, type the distance measured to the second point in position units.

11. Under **Second position**, click **Use Current**.
12. Under **Extend/Retract Limits**, choose how you wish to have the limits set:
    - If you had the extend and retract limits set correctly, click **Use current limits, adjusted for new Scale and Offset** to adjust the limits for your new Scale and Offset.
    - If the points you used for the two positions are your limits, click **Set limits to above positions**.
    - Otherwise, click **Set limits to the following values**, and type the limits in the edit boxes.
13. Click **Apply**, which sets the Scale, Offset, the Prescale Divisor bits of the Configuration word, Extend Limit, and Retract Limit.
14. Click **Done**.
15. Issue a 'P' command for the axis to have the new parameters take affect.

MDT Scale/Offset Calibration Utility

For a description of all Scale/Offset Calibration Utilities, see Using the Scale/Offset Calibration Utilities. To use the MDT Scale/Offset Calibration Utility:

1. Gather the following pieces of information for an axis:
    - The gradient of the transducer and its units (e.g. 9.012µs/inch)
    - The number of recirculations for the transducer
    - The desired number of position units per inch or centimeter
    - The transducer counts at your desired zero position
    - Whether you wish to have your position units increase or decrease with increasing counts
2. Place the cursor in a field under the axis you wish to calibrate.
3. On the **Tools** menu, click **MDT Scale/Offset Calibration**.
4. Enter the above pieces of information.
5. Under **Extend/Retract Limits**, choose how you wish to have the limits set:
    - If you had the extend and retract limits set correctly, click **Use current limits, adjusted for new Scale and Offset** to adjust the limits for your new Scale and Offset.
    - Otherwise, click **Set limits to the following values**, and type the limits in the text boxes.
6. Click **Apply**, which sets the Scale, Offset, the Prescale Divisor bits of the Configuration word, Extend Limit, and Retract Limit.
7. Click **Done**.
8. Issue a 'P' command for the axis to have the new parameters take affect.

Using Plots

Using Graphs of Axis Moves

The motion controller automatically gathers plot data for each move.   A plot is triggered when any of the following commands are issued: Go, Open Loop, and Relative Move. The plot data then is collected for the duration and at a sampling rate as determined by the Plot Time field on the main screen. The Plot functionality in the monitor program can be used to read and display these graphs.

Click on one of the following topics for more details on using plots:

- Opening a Plot Window
- Reading Plot Data from the Motion Controller
- Selecting the Data to Plot
- Using the Plot Detail Window
- Viewing the Raw Plot Data
- Saving and Restoring Plots
- Printing a Plot

Opening a Plot Window

You can use one of the following methods to open a plot window:

- Click **Plot** ( ) on the Toolbar. This opens a plot window for the current axis.
- Press INSERT. This opens a plot window for the current axis.
- On the **Window** menu, click on the **Plot** item of your choice. This opens a plot window for the axis indicated by the menu item name. If a plot window for this axis is already open, this command will shift the focus to the plot.

Reading Plot Data from the Motion Controller

If a motion controller is connected to the monitor program, then when a plot window is opened, the data will be read from the motion controller. The plot data stored in the motion controller for that axis will change as new commands are issued to the axis. Therefore, it is often desirable to re-read data from the axis.

To re-read data from the motion control module after opening a plot window, using one of these methods:

- On the **Data** menu, click **Upload plot from module**.
- Press INSERT.

Selecting the Data to Plot

All plots read from the motion controller contain the following information:

- ACTUAL POSITION
- TARGET POSITION
- DRIVE
- STATUS word

In addition, ACTUAL SPEED and TARGET SPEED are calculated from the ACTUAL POSITION and TARGET POSITION values and stored in the plot.

The motion controller can provide two more pieces of data; the user chooses what data is collected using the **Plot Options** dialog box. To use this dialog box:

1. If you are going to change the extra graph information on only one axis, select a field in the axis.
2. On the **Tools** menu, click **Plot options**.
3. Click the option button of the data you want to include. The options are described below.
4. If you wish to change the extra graph information for all axes, select the **Set for all axes** check box.
5. Click **OK**.
6. Trigger a new graph to be stored by the module. This requires making a new move because the new information is not collected until the module begins a new graph.

The four extra pieces of information that can be included are:

- **Extra Position Precision** – No additional data is displayed in the graph when this data is selected. Instead more precise positions are read from the module, which results in the better speed calculations. Use this option to get better approximations of the speed.
- **Command/Command Value** – The last command and command value received on the axis are stored for every plot entry. This is useful to verify the arrival and effects of new commands given during a move. **NOTE:** commands given from the monitor program are not included in the plot data.
- **Current Event/Link Value** – The current event STEP and LINK VALUE are stored for every plot entry. This is useful for debugging event sequences.
- **Raw Transducer Counts** – The raw counts read from the transducer are stored for every plot entry. Refer to Raw Transducer Counts for details.

Using the Plot Detail Window

A plot detail window is opened by default when a plot is read into the monitor program. This window has a title of **Data at 0.000 s** with the actual number changing depending on the hairline position. The hairline is a vertical line that can be moved using the arrow keys, page up and down keys, and by using the left mouse button. The data values displayed in the detail window represent the plot at the hairline position. Therefore, to read exact values at a position in the plot, move the hairline to the desired position on the plot and read the detail window data.

To hide the Detail Window, use one of the following:

- On the **Data** menu, click **Hide Detail Window**.
- Click the **Close** button of the Detail window.

To show the Detail Window after it has been hidden:

- On the **Data** menu, click **Show Detail Window**.

To display the individual bits on the status word, do one of the following:

- Click on the body of the detail window. This toggles the detail window between displaying the status word as a hexadecimal number and as the bit names. When displayed as bit names, the names of the off bits are displayed in dark gray, while the names of the on bits are displayed black for non-errors, and red for errors.
- On the **Data** menu, click **Display plot status bits**.
- Press CTRL+B while the plot status window is displayed. This will toggle the detail status bits.

To move the Detail Window, do one of the following:

- Drag the detail window by its title bar.
- On the **Data** menu, click **Move detail window**. Each time this command is clicked, the detail window will move to the next corner of the plot window.
- Press the TAB key, which executes the **Move detail window** command.

Viewing the Raw Plot Data

To view the plot data in numerical form rather than in graph form, you can use the Raw Data chart. This chart displays all the data in the plot at every sample. The sample that was marked by the hairline on the plot is highlighted in gray. You can use the scroll bar or arrow keys to move through the data.

To view the Raw Data chart, do one of the following:

- On the **Data** menu, click **View Raw Data**.
- Press CTRL+V while in the plot window.

To return from the Raw Data chart to the plot, do one of the following:

- On the **Data** menu, click **View Plotted Data**.
- Press ESC while in the Raw Data chart.

Saving and Restoring Plots

To save a plot, follow these steps:

1. Display the plot you wish to save.
2. On the **File** menu, click **Save As**.
3. In the **File name** box, enter the name of the file.
4. Click **Save**.

To view a previously saved plot, follow these steps:

1. Open a plot window. This may start loading a new plot from the motion control module. It is not necessary to stop the download before opening a saved plot.
2. On the **File** menu, click **Open**.
3. In the **File name** box, enter the name of the file.
4. Click **Open**.

Printing a Plot

Plots can be printed out; each will take a single page. The user can set the margins, decide whether or not to print out the hairline with its detailed information, and choose to print in either landscape or portrait modes.

To set the margins:

1. On the **File** menu, click **Print Setup**.
2. Under **Margins**, enter the sizes of the margins in the four text boxes. These settings are in inches.
3. Click **OK**.

To print the hairline and the plot details at that time:

1. On the **File** menu, click **Print Setup**.
2. Select the **Print hairline and plot details** check box.
3. Click **OK**.

To select landscape or portrait modes:

1. On the **File** menu, click **Print Setup**.
2. Click **Printer Setup**.
3. Under **Orientation**, click either **Portrait** or **Landscape**.
4. Click **OK** in the **Print Setup** dialog box.
5. Click **OK** in the **Plot Print Options** dialog box.

To print a plot:

1. On the **File** menu, click **Print**.

2. Select the printer and number of copies to print.
3. Click **OK**.

Plot Time

The plot time field controls the time interval between plot samples. Because the number of samples for a full graph is fixed, the total graph length is also set by this parameter. In a full graph, there are 1024 samples. The plot time units are 1000/1024 milliseconds. Therefore, to request the graphs on an axis to be 4 seconds in length, this parameter should be set to 4. This will result in 1024 samples, each 4000/1024 milliseconds apart, for a total length of 4000 milliseconds.

The following two equations can be used for requesting a plot length:

$$\text{Full Plot Length} = \text{PLOT TIME} \times 1\ \text{second}$$

$$\text{Plot Interval} = \frac{\text{PLOT TIME} \times 1000\ \text{ms}}{1024}$$

Special Status Values Available in Plots

TARGET SPEED

The target speed is displayed in a plot, both graphically and in the detail window, but it is not displayed in the main status area. Like the ACTUAL SPEED, this parameter is calculated. However, instead of being calculated off of TRANSDUCER COUNTS, it is calculated from the TARGET POSITION. Therefore, when viewing a plot you can compare the ACTUAL and TARGET SPEEDS to determine how well the actual move is tracking the intended move.

Why is the TARGET SPEED Jagged During Constant Velocity?

With older versions of the firmware and monitor software, you may notice that during some constant velocity moves, the TARGET SPEED is not flat. The jaggedness caused by quantizing. That is, suppose that at constant velocity the axis is moving an average of 4.8 position units per millisecond. Because the motion controller only reads whole numbers of position units, the TARGET POSITION would appear to increase by 4 position units for four milliseconds, and every fifth millisecond it would move 5 position units. Therefore, if we were to calculate the speed based on a single interval, the speed would just by 25% every fifth interval.

When the TARGET SPEED is calculated, smoothing is performed so that the speed does not vary by 25%, but some jaggedness is left from rounded-off position units.

Raw Transducer Counts

The **MDT Counts (lo)** field holds bits 0-15 bits of the TRANSDUCER COUNTS at the given plot time period. The **MDT Counts (hi)** field is provided for use by technical support, but also holds bits 16 and 17 of the current count. Therefore, if you are using a long enough MDT and are using counts above 65535, then you can combine the two fields to find the actual counts at each point on the graph.

Sum of Errors Squared

This field is computed by summing the squares of the position error for each time unit on a graph. The lower this value, the closer the actual position curve tracked the target position curve. The numerical value displayed in this field has no meaning by itself, but it can be used during tuning to check whether a tuned parameter was helpful or harmful to the tracking of the position. In order to have a valid comparison the two graphs must have the same length, and the moves need to have had the same target profile. Otherwise, the changed parameter is not isolated in the test.

Table Editors

Table Editor Basics

Starting a Table Editor

To start an editor, from the **Tools** menu, click the desired editor. Notice that each editor also has a shortcut key that can be used to start the editor. These are displayed in the menu.

When the table editor is opened, the following steps will be taken:

1. For the tables other than the Stored Command table: if a motion controller is connected to the monitor, the table will be read from the module.
2. If a motion controller is not connected to the monitor, the table will be read from a file. The filename used is given by the board filename, with the appropriate extension appended.
3. If a motion controller is not connected to the monitor and the table could not be read from disk, then default values will be filled into the table.

Exiting an Editor

To close an editor, use one of these methods:

- Press ESC.
- On the **File** menu, click **Exit**.
- Click the **Close** button.

If changes have been made to the table that have not been saved to a file or downloaded to the motion controller you will be prompted to do so upon exit. You will always be given the option of exiting without saving.

Editing the Tables

Once the editor window is open, you can type values in the same manner as on the main window. See Changing Data from the Keyboard for details.

Additional Editing Features in the Table Editors

To copy a columns of values to many adjacent columns:

1. Enter the values you wish to have copied into adjacent columns in all fields in the *left-most* column in the range.
2. Select the range of cells with the keyboard or mouse. Make sure that the fields you entered are the left-most selected column.
3. On the **Edit** menu, click **Fill Right**. The left-most selected values will be copied to all other selected columns.

Resizing the Window

To resize the window, you may drag the borders of the window. After the border is released, the window will snap to the largest number of columns and rows that fit on the screen.

Resizing the Columns

Resizing the window does not affect the width of the columns. There are two column widths that are stored. The first column, which gives the row titles, can have its width set independent of the other column widths. However, changing any one of the other column widths affects all columns other than the label column.

To resize the row title column:

1. Drag the right edge of the label heading. This will only affect the title column.

To resize all other columns:

1. Drag the right edge of a data column. This will affect the width of all data columns.

Saving and Restoring Tables

To save a table:

1. On the **File** menu, click **Save As**.
2. In the **File name** box, enter the name of the file.
3. Click **Save**. The file will be saved in text format.

To restore a table:

1. On the **File** menu, click **Open**.
2. In the **File name** box, enter the name of the file.
3. Click **Open**.

Uploading and Downloading Tables

Except the Stored Command table, all tables are stored on the motion module. The motion controller module will only use the tables stored in its memory. Therefore, you must upload and download the tables to work on them in the table editors. If a motion controller is attached to the monitor program, then the table will be uploaded when the table editor is opened.

To download a table to the Module, use one of these methods:

- On the **Online** menu, click **Download to Motion Controller**.
- Click **Download to Module** (D) from the toolbar.

To upload a table from the Module, use one of these methods:

- On the **Online** menu, click **Upload from Motion Controller**.
- Click **Upload from Module** (U) from the toolbar.

Editing the Stored Command Table

Stored Commands Explained

The Stored Command table stores the 10 sets of commands with full profiles (MODE, ACCELERATION, DECELERATION, SPEED, COMMAND, and COMMAND VALUE) for each axis. See Using Stored Commands for details on using these stored commands to send commands to an axis using the monitor program.

Changing the Stored Command Table

These stored commands are stored in the monitor program and can be saved to disk for later retrieval. They are edited using the Stored Command table editor. Refer to Table Editor Basics for topics common to all table editors. The default extension for saved Stored Command tables is .fn1. Changes made to the Stored Command table are applied immediately.

Editing the Profile Table

Profiles Explained

The Profile table stores 16 motion profiles. For details on Motion Profiles, see Motion Profiles.

Changing the Profile Table

Changes are made to the Profile table using the Profile Editor. Refer to Table Editor Basics for topics common to all table editors. The default extension for saved Profile tables is .pr1.

Editing the Event Step Table

Event Steps Explained

The Event Step table contains 256 Event Steps. For details on the use of Steps, refer to Event Control Overview.

Changing the Event Step Table

Changes are made to the Event Step table using the Event Step Editor. Refer to Table Editor Basics for topics common to all table editors. The default extension for saved Event Step tables is .st1. Features specific to the Event Step table editor are described below:

Jumping to an Event Step

Because the Event Step table has many columns, you may want to go to a specific Event Step. There are two ways of doing this:

- On the **Edit** menu, click **Goto Next Step in Sequence** or press CTRL+F. This will move the cursor to the step referred to by the current step's Link Next field.
- On the **Edit** menu, click **Go to Event Step** or press CTRL+G. This is a more general method, which prompts you to type the step number you want to go to.

Deleting Columns

If you wish to delete a column or columns, follow these steps:

1. Click on the step number that is located at the top of each table column. This will highlight all fields in the step.
2. If you wish to select a range of steps, hold down the SHIFT key and click on the step number and the other end of the range. Now all steps between and including the two selected steps will be highlighted.
3. On the **Edit** menu, click **Delete Column x to Clipboard**.

Notice that the cells that were deleted are stored in the clipboard.

Inserting Columns

You can insert columns into the Event Step table in two ways:

To insert an empty column:

1. Select a field in the step that you want the step to be inserted in front of.
2. On the **Edit** menu, click **Insert Empty Column**.

To insert the columns in the clipboard from a delete or copy command:

1. Select a field in the step that you want the steps to be inserted in front of.
2. On the **Edit** menu, click **Insert x Columns from Clipboard**.

Automatically Updating Links

Whenever event steps are shifted left or right as described above under **Deleting Columns** and **Inserting Columns**, there is the potential of links to the shifted steps becoming broken. RMCWin is capable of adjusting links to steps that move using one of the above methods. This feature is enabled by default.

To toggle this feature on and off:

- On the **Settings** menu, click **Adjust Links on Column Operations**.

NOTE: The links affected by this option include those by the Link Next field, and the Command Value of Start Event (E) and Teach Step (t) commands. Links from the Input to Event table can be checked as described under **Maintaining Input to Event Table Links** below.

Reporting Orphaned Links

Steps are deleted when the **Delete Columns to Clipboard** command described above is used. Also when steps are inserted into the Event Step table, the steps at the top are lost. For example, if five steps are inserted before step 10, then steps 10 to 255 are all shifted right. However, because there can only be 255 steps in the table, steps 251 to 255 are shift off the end and lost.

Care must be taken to ensure that these lost steps weren't used. Therefore, the Event Step editor can automatically report when links to deleted steps are broken (orphaned). This feature is enabled by default.

To toggled this feature on and off, do the following:

•On the **Settings** menu, click **Report Orphaned Links**.

Maintaining Input to Event Table Links

Whenever event steps are shifted left or right as described above under **Deleting Columns** and **Inserting Columns**, there is the potential of entries in the Input to Event table to become broken. When this option is enabled, the currently open Input to Event table will be checked to see if any events pointed to by the Input to Event table have moved or been deleted. If the Input to Event table editor is not open, it will temporarily be opened and load the table from the module or from the default file if no module is connected. This newly-loaded table will be checked.

To toggle this feature on and off, do the following:

- On the **Settings** menu, click **Maintain Input to Event Links**.

Adding Comments

The Event Steps editor offers the ability to enter a comment for each event step. These comments are then saved and restored with the step file. When a step has a comment associated with it, a comment icon appears on the heading for the step (🗒). Comments are copied or moved with the step when all fields in a step are copied or moved —whether by clipboard commands such as cut, copy, and paste, or by inserting or deleting steps.

The comment editor is used to add, edit, and remove comments. Use one of the following methods to start the comment editor while in the Event Steps editor:

- On the **Edit** menu, click **Edit Comment**.
- Press CTRL+N.

- Double-click the heading of a step.

Using any of the above methods when the comment editor is already open will move the keyboard focus to the comment editor.

The comment editor will stay up until explicitly closed; you can freely switch between editing steps and their comments. The comment editor will display the comment for the step currently selected in the Event Steps editor. Changes made to comments take place immediately; therefore, you do not need to close the comment editor or move to another step to finalize changes before saving the file.

The comment editor responds to the following commands:

- Click **Restore** or press ALT+R to reset the comment for the current step back to its state when this step was first selected. This can also be viewed as an undo.
- Click **Clear** or press ALT+C to erase the entire comment for the current step.
- Press CTRL+F or CTRL+G to jump to another step in the event step table without changing the keyboard focus back to the main window. See the **Jumping to an Event Step** section above for details on the differences of these commands.
- Press CTRL+N to switch keyboard focus back to the Event Steps editor.
- Press ALT+F4 or click the **Close** button to close the comment editor.

Advanced Topics

Downloading New Firmware

If new features have been added or problems fixed in the motion controller firmware, then it is necessary to update the firmware to take advantage of these improvements. You should only use the firmware download feature at the instruction of technical support to address a specific need.

To download new firmware:

1. Obtain the new firmware from technical support.
2. Start DCS120-Win.
3. Turn off the hydraulics system. This is necessary because the system must be reset for the new firmware to take effect. During this time, the module cannot control motion.
4. On the **Tools** menu, click **Module Configuration**.
5. Click **New Firmware**.
6. After reading the warning, click **Yes** if you feel it is a safe time to have the motion controller reset. Otherwise, click **No**.
7. You will be asked to find the file you will use as the new firmware file. Enter the name of the firmware file provided by technical support, and click **Open**. This can be on a floppy or on the hard drive.
8. The version number will be displayed. Verify that this firmware date is newer than the current firmware you are using. If so, click **Yes**, otherwise click **No** to select

another firmware file.

9. The firmware will then be downloaded.

Forcing Initialization

It is possible to have the monitor program automatically send the parameters it has stored in a file to the motion control module. In most cases this should not be necessary because there are two other methods of ensuring a module is initialized. Both are more desirable:

1. The parameters can be stored in the FLASH. A module uses the parameters stored in the FLASH when it starts up.
2. The parameters can be stored in the Programmable Controller and downloaded when the module is started up. This is usually desirable when a Programmable Controller is available to ensure that the modules are always configured correctly even if a module is swapped out.

To enable Forced Initialization, you must start the program using the –F command line parameter. This can be added as a parameter by editing the shortcut which starts the monitor program or by typing the parameter after the program name when start it from the command line.

Using Look-only Mode

In Look-only mode, the user cannot make changes to the module. This prevents the user from changing any parameters or issuing any commands. It is still possible to view all data.

To toggle look-only mode:

1. On the **Tools** menu, click **Options**, and then click the **Preferences** tab.
2. Select or clear the **Look-only** check box to enable or disable look-only mode.
3. Click **OK**.

Notice that this setting is saved away so that it will be remembered the next time that the monitor program is started.

Using PC Mode

If the monitor program is going to be used for demonstration purposes without actually being attached to a module, you can use PC mode. The difference between PC mode and regular operation is that in PC mode a serial port is not used at all. This means that you cannot connect to a module.

Although a monitor program that is not in PC mode can also run when not being connected to a module, it will still be using a serial port, thus preventing another application from using it.

To run the monitor program in PC mode, you must start the program using the –P command line parameter. This can be added as a parameter by editing the shortcut which starts the monitor program or by typing the parameter after the program name when start it from the command line.

Command Line Options

The monitor program can be started with various command line options. These options can be used to ensure that the program starts with the correct options. Notice that most of these settings are already saved between restarts of the monitor program, so most users will have no need for these options. Options given on the command line override the previously saved settings.

The following options are available:

-1, -2, -3, -4	Select COM1, COM2, COM3 or COM4 as the serial port to use. See Selecting a Serial Port to Use for details.
-F	Force initialization onto module. See Forcing Initialization for details.
-D	Generate debug file. All serial port communications are logged to DEBUG.TXT in the current directory.
-L	Start in Look-only mode. See Using Look-only Mode for details.
-P	Start without using any COM port. This is useful for demonstrations or offline editing when all other serial ports are in use.
-R	Start in Read-back mode. See Read-back versus Write Mode for details.
-W	Start in Write mode. See Read-back versus Write Mode for details.
Board file	The parameter after the filename that does not start with a '-' or '/' gives the board filename that is used on startup (e.g. mmcwin carriage.bd1). See Using Multiple Motion Modules for details on board files.

Module Configuration Dialog Box

The **Module Configuration** dialog box gives information on the firmware and features available in the MMC120. It also provides the ability to update the firmware in the MMC120.

It contains the following fields:

- **Boot Version**
  This version corresponds to a piece of firmware in the motion controller that decides whether to run the loader or controller firmware.
- **Loader Version**
  This version corresponds to the firmware used to download new controller firmware.
- **Controller Version**
  This version corresponds to the actual controlling firmware.
- **Features**
  This hexadecimal value indicates the features available in the motion controller module. This value is undocumented but is provided for use in technical support.

- **Xilinx Version**
  This version corresponds to the Field Programmable Gate Array (Xilinx) firmware.

There is one command available:

- **New Firmware**
  This command starts the Controller firmware download procedure. See Downloading New Firmware for details.

Options Dialog Box

To display the monitor options dialog, from the **Tools** menu, click **Options**.

This dialog box has the following information:

Serial Port Tab

Comm. Port list:

This list allows the user to change the serial port used to communicate between the motion controller and the monitor program on the PC. Refer to Selecting a Serial Port to Use for further details. An alternative way to change this setting is to right-click on the main-screen status bar's COM pane, and select the serial port from the shortcut menu.

Open/Closed list:

This list allows the user to manually open or close the serial port. By closing the serial port manually, the serial port can be freed up to use by other applications. This setting does not affect whether DCS120-Win will open the serial port by default. This can instead be controlled through the **-P** command-line option; see Command Line Options for details. An alternative way to change this setting is to right-click on the main-screen status bar's COM pane, and click **Open** or **Close** from the shortcut menu.

Serial Port Update Rate slider:

This slider adjusts a delay that is inserted between communications over the serial port on the PC when talking to a motion controller. The only purpose for this control is to decrease the load on slower PCs; the motion controller can handle any of the communication speeds. If this software seems to slow down Windows, move this slider closer to “Slow”.

Move the slider to the left in decrease the delay between communications with the motion controller. Move the slider to the right to increase this delay.

NOTE: This setting is ignored while reading up a plot; no delay is used.

Use Small-Packet Mode check box:

By selecting this check box, the size of the packets sent over the serial port are reduced. This is necessary on some computers that do not give Windows enough time to service the serial port queue, and therefore cause serial port overruns. Most notably this has been known to happen on some computers running Steeplechase Software's Visual Logic Controller. If you experience problems maintaining the connection with the motion controller, try checking this box.

NOTE: By checking this box, the communication speed will slow by roughly 40%. Therefore, only use this option if necessary.

Axis Names Tab

This tab contains text boxes for editing the names of the axes. Refer to Changing the Axis Name for further details.

Preferences Tab

Do not show confirmation warnings check box

Select this check box to suppress confirmation-warning messages. This includes all warnings when closing a window to save the contents to either the motion controller or the disk. Select this check box *only* if you know the program very well and understand when your work is saved.

Look-only check box

In Look-only mode, the user cannot make changes to the module. This prevents the user from changing any parameters or issuing any commands. It is still possible to view all data. Notice that this setting is saved away so that it will be remembered the next time that the monitor program is started.

Controller Features

Event Control Overview

The Event Control feature allows you to execute a sequence of commands without intervention from the Programmable Controller (P/C). This lets the module respond to events within one millisecond rather than the scan rate of the P/C. It also reduces the controller programming required.

Event Control consists of a series of Steps that are linked together in sequences. The Steps consist of a command area containing the instruction to be executed and a link area that specifies the next Step number and its trigger. There are a total of 256 Steps that can be shared by all axes.

Steps

Each Step contains a command with its associated parameters, plus the information necessary to link to the next Step in a sequence. The Step format is as follows:

Command Info:	Mode	MODE word
	Accel	Acceleration
	Decel	Deceleration
	Speed	Requested Speed
	Command Value	Requested position or command value
	Command	Any valid ASCII command
Link Info:	Link Type	Condition that triggers execution of next step
	Link Value	Parameter associated with Link Type
	Link Next	Number of next Step in sequence

Storing Steps

Steps are stored in the motion controller's memory. This step table can be saved in the motion controller's FLASH using the Update FLASH command. However, in some applications this information should be stored in the Programmable Controller so that it can be downloaded to the motion controller on initialization. Refer to the Event Step Transfer for details on storing the steps.

Changing Event Steps

Refer to Editing the Event Step Table for details on using the Event Step editor.

Basic Event Step Flow

After a valid event step sequence is loaded into the motion controller, event control can be used. This section describes the basic flow:

To start a sequence of events, do the following:

- Issue a Start Event command to trigger an event from the monitor program or Programmable Controller.

A sequence of events will stop when one of the following occurs:

- A Quit Events command has been issued from the monitor program or Programmable Controller.
- A step is executed which has a Link Type of 0.
- A Halt or Disable Drive Output command will stop the event sequence as well.
- Another Start Event command will stop the current sequence and start the new one.

The following additional details should also be kept in mind:

- The Mode, Accel, Decel, Speed and/or Command Value fields need not be filled out if the command does not use them.
- The command field can be left blank in a step if only the link type is being used. You can chain steps together to make a sequence wait for multiple conditions to all be true.

- The command in an event will be executed as soon as a step is reached in the event sequence.
- The next Step in the sequence will be executed as soon as the conditions from the Link Type and Link Value are met.
- Multiple axes can be made to follow identical patterns simultaneously by using the same steps on each.
- You cannot conditionally branch steps. This is because each step only has a single Link Next field.

Example

In the following example three steps are executed. They cause the axis to make a move, wait and then make another move:

	Step 15	**Step 16**	**Step 17**
Mode	00001	00001	00001
Accel	100	100	100
Decel	100	100	100
Speed	10000	10000	10000
Command Value	15500	10000	3000
Command	G		G
Link Type	B	D	D
Link Value	00001	500	0
Link Next	16	17	0

Step 15 issues a Go command to 15.5 inches (15500). The link type is B with a link value of 00001, which causes the motion controller to look for the least significant bit in the STATUS word (the In Position bit). When the In Position bit turns on, indicating the move is complete, Step 16 (Link Next – 16) is executed.

Step 16 has no command so no command is issued. The link type is a Delay with a value of 500 milliseconds. After 500 milliseconds pass, Step 17 executes.

Step 17 issues a Go command to 3 inches. The Link Type D with a value of 0 (zero) causes the axis to go immediately to Step 0, which ends the sequence.

FLASH Memory

The MMC120 contains FLASH memory that has two user-accessible sections. Section 1 contains the following:

- All axes' parameters
- The profile table

Section 2 contains the following data:

- The entire step table

Each section is protected by a checksum; these checksums are tested on module power-up and reset.

Determining if the FLASH Data was Used

Prior to issuing the first Set Parameters command on an axis, the Initialized bit in the STATUS word will be cleared. While this bit is cleared, the Parameter error bit in the STATUS word of each axis will indicate whether the FLASH memory checksums were valid.

If the parameter error bit for axis 1 is set on startup, then the checksum for section 1 was invalid and default parameters and profiles will be used. If the parameter error bit for axis 2 is set on startup, then the checksum for section 2 was invalid and a blank step table will be used.

Storing Data in the FLASH

To update the FLASH information, you must first download correct data to the MMC120 volatile memory. Use the Set Parameter, Set Profile, and Event Step Transfer commands to download the information to the motion controller.

After the data to be stored has been downloaded to the motion controller's volatile memory, transfer it to the FLASH memory using the Update FLASH command. To transfer data to section 1 (both parameters and profiles), put a '1' in the COMMAND VALUE field when the Update FLASH command is issued. To transfer data to section 2 (event steps), put a '2' in the COMMAND VALUE field when the Update FLASH command is issued.

The Acknowledge bit of the STATUS word is used to report the result of the update. If the FLASH updated successfully, then the Acknowledge bit of axis 1 is toggled. Otherwise the Acknowledge bit of axis 2 is toggled.

The disadvantage of storing configuration data in the motion controller is that, when one module is replaced with another, the parameters, profiles and event steps must be loaded into the new module. Because of this, all parameters, profiles and event step table data must be stored either in the control program or in monitor program files so they can be later transferred to the motion controller when needed.

LED Indicators

There are 14 green and red light emitting diodes (LEDS) used on the front panel of the MMC120. These LEDS provide status information about the module and each of the two axes.

The LED labeled "F" is the FAIL indicator for the MMC120 microprocessor. If the FAIL indicator is red, the drive outputs are disabled, and the axes will not move unless the value is out of null.

The LED labeled "Active" indicates the status of communications over the back plane. When this LED is green, communication is normal. When it is off, the communication has stopped.

The numbered LEDS indicate the status of the axes.

NOTE: There are three transducer errors: No Transducer, Transducer Noise, and Transducer Overflow. The default AUTO STOP settings will make any of these errors cause a Hard Stop, set the output of the axis to the current drive null, and turn on its red LED. Once the transducer error has been corrected, a new command to the axis will turn the red LED off.

"F" LED red	Onboard processor halted; drive outputs disabled
"Active" LED off	Communication disabled
"Active" LED green	Communication OK
Green LED 1	Axis 1 Parameters Initialized
Green LED 2	Axis 1 Transducer OK
Green LED 5	Axis 2 Parameters Initialized
Green LED 6	Axis 2 Transducer OK
Red LED 1	Axis 1 Hard Stop/Open Loop
Red LED 2	Axis 1 No Transducer
Red LED 3	Axis 1 Transducer Noise
Red LED 4	Axis 1 Transducer Overflow
Red LED 5	Axis 2 Hard Stop/Open Loop
Red LED 6	Axis 2 No Transducer
Red LED 7	Axis 2 Transducer Noise
Red LED 8	Axis 2 Transducer Overflow

Motion Profiles

What is a Motion Profile?

A profile is a speed, acceleration, deceleration and mode used during a move. The profile specifies *how* the axis will move to the requested position. Commands that will change the profile are:

Set Mode - Changes the MODE bits.

Change Acceleration - Changes the acceleration rate.

Change Deceleration - Changes the deceleration rate

Set Speed (Unsigned) and Set Speed (Signed) - Changes the speed.

NOTE: The profile does *not* include the requested or starting positions.

What is the Motion Profile Table?

The Motion Profile Table stores sixteen motion profiles (Mode, Acceleration, Deceleration, and Speed) in its FLASH memory.

Why use the Motion Profile Table?

The PLC can issue only a Command and Command Value in a single command cycle. Therefore, the Go command can only give the Requested Position as the command value, so to set the entire motion profile and give the Go command would require five commands. While this is flexible, it requires a great deal of programming and run-time.

By using the Motion Profile Table, a complete move including all four motion profile fields can be set with a *single* Go Using Profile command, speeding up both execution and development time.

How is the Motion Profile Table Used?

The Go Using Profile and Open Loop Using Profile commands select one of the 16 motion profiles in the table and use that profile to issue the command.

How is the Motion Profile Table Changed?

There are two ways to the change the motion profiles table:

- Use the Motion Profile Table Editor. On the **Tools** menu, click **Profile Editor** to start this editor; see Editing the Profile Table for details.
- Use the Set Profile commands to change the table from the PLC.

How do I Save the Motion Profile Table?

There are three places where the Motion Profile Table can be saved:

- FLASH memory. Use the Update FLASH command to store the current profile table in the FLASH memory.
- Disk file. The table can be saved to disk from within the Motion Profile Table Editor.
- PLC memory. Use the Set Profile and Get Profile commands to send the table to and from the PLC.

What are the Default Motion Profile Values?

When the motion controller starts, it first does a checksum on the profile table in the FLASH memory. If the FLASH-stored profile table checksum is correct, it is used. If the checksum is incorrect, then the following default profiles are used:

Profile	**0**	**1**	**2**	**3**
MODE	0	0	0	0
ACCEL	1000	2000	5000	8000
DECEL	1000	2000	5000	8000
SPEED	1000	2000	5000	8000

Profile	**4**	**5**	**6**	**7**
MODE	0	0	0	0

ACCEL	10000	12000	15000	18000
DECEL	10000	12000	15000	18000
SPEED	10000	12000	15000	18000

Profile	**8**	**9**	**10 (0A)**	**11 (0B)**
MODE	1	1	1	1
ACCEL	100	100	100	200
DECEL	100	100	100	200
SPEED	20000	25000	30000	35000

Profile	**12 (0C)**	**13 (0D)**	**14 (0E)**	**15 (0F)**
MODE	1	1	1	1
ACCEL	200	200	200	100
DECEL	200	200	200	100
SPEED	40000	45000	50000	10000

Synchronizing Axes

Axis synchronization is achieved by setting the Sync bit in the MODE word on both axes and then issuing a Go or Relative Move command to the last axis.

The following requirements exist when synchronizing axes:

- Both axes must be initialized, as indicated by the Parameter Initialized bit in the STATUS word. Do this by issuing the Set Parameters command.
- The low eight MODE bits of both synchronized axes must match exactly.
- Synchronized commands (Go and Relative Move) must be issued to the second axis.
- Synchronized commands issued to the first axis are ignored.
- The MODE, ACCELERATION, DECELERATION, SPEED, and COMMAND VALUE fields must be set on both axes before any synchronized commands are issued.

Internal to the MMC120, the synchronization is accomplished as follows:

- The axis with the longest travel distance is designated the “master”. If both axes are moving the same distance, then first axis is designated the “master”. The other axis is designated as the “slave”.
- The maximum speed of the master axis is limited so neither of the synchronized axes will go above their requested SPEED.
- The master then moves toward its destination using its own ACCELERATION, DECELERATION, and the speed calculated in the above step.
- At the same time, the TARGET POSITION of the slave axis is ratioed to the target position of the master based on the distance each is traveling.
- If either synchronized axis is halted during the move, the other axis will halt as well.

- If a new Go or Relative Move command is given to the *slave* axis *which removes* the Sync bit used in the MODE word, then that slave will perform the new move, and the master will continue its move.
- If a new Go or Relative Move command is given to the *master* axis *which removes* the Sync bit used in the MODE word, then the master will perform the new move, but the slave axis will stop.
- If a new Go or Relative Move command is given to the second axis *which retains* the Sync bit in the MODE word, then the axes will start a new synchronized move from their current positions.

NOTE: Because a new Go or Relative Move command will cause the travel distance ratios to be recalculated, any change in the speeds or travel distances of any of the axes may result in a speed discontinuity in the new slave axis, although the position will always be continuous.

It is normally not possible to use DCS120-Win to issue commands to both axes simultaneously. However, when a command is issued to an axis that has a sync bit set, the software will automatically issue commands all axes with the Sync bit set. This can only be done while in Write mode. Similarly, if a Stored Command is a synchronized command and is executed, the Stored Command for the other axis will be executed simultaneously if it has the Sync bit set also.

VC2100 Two Axis Voltage-to-Current Converter

The VC2100 voltage-to-current converter transforms ±10V signals into current signals capable of driving hydraulic servo valves or similar loads. It also provides a convenient way to set the full scale current to match valve requirements, limit maximum current and set optimum working ranges.

Features

- Two channels of voltage-to-current conversion
- Full scale range (FSR) output current for each channel is switch-selectable from ±10mA to ±100mA in 10mA steps
- Dual-color LEDs indicate input polarity and amplitude
- Outputs protected against inductive voltage spikes
- Compact DIN rail mount
- Use with ±12 to ±15 volt power supplies

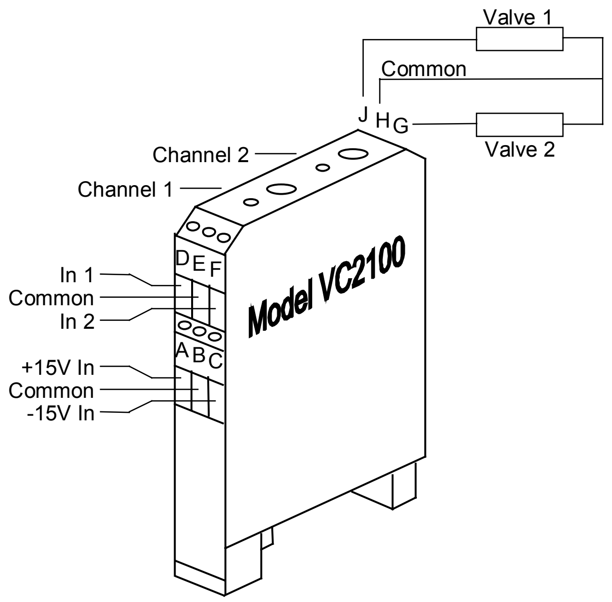
Output Characteristics

This table shows the minimum output drive voltage and maximum load resistance for various output currents and power supply voltages.

Output	±15V ±5% Supplies			±12V ±5% Supplies		
Current	Vout		Maximum	Vout		Maximum
mA	Typ	Min	Load Ω	Typ	Min	Load Ω
10	13.7	12.7	1265	10.7	9.8	980
20	13.2	12.2	608	10.2	9.3	465
30	12.7	11.7	388	9.7	8.8	293
40	12.2	11.2	279	9.2	8.3	208
50	11.7	10.7	213	8.7	7.8	156
60	11.2	10.2	169	8.2	7.3	122
70	10.7	9.7	138	7.7	6.8	97
80	10.2	9.2	114	7.2	6.3	79
90	9.7	8.7	96	6.7	5.8	65
100	9.2	8.2	82	6.2	5.3	53

The VC2100 can drive a short circuit to common—an internal 50Ω current-limiting resister limits the output current. The output amplifier will shut down under severe overload (such as driving a short to a power supply).

Wiring Information



Terminal	Function
A	+15 volt supply in
B	Power supply common
C	-15 volt supply in
D	Voltage Input 1
E	Common
F	Voltage Input 2

G	Current Output 2
H	Common
J	Current Output 1

Hardware Specifications

Inputs	Two per unit
Indicators	1 dual color LED per channel; Green = positive input, Red = negative, intensity indicates amplitude
Input Impedance	40kΩ
Input voltage range	±10V
Overvoltage protection	±100V (Outputs may reverse polarity if inputs are above power supply voltages)
Outputs	Two per unit
Output current range	±10mA to ±100mA, switch-selectable (See the Output Characteristics table for more information)
Conversion Accuracy	0.6% full scale, typical, all ranges
Offset	0.05% typical, ±100mA range 0.5% typical, ±10mA range

Power Supply Requirements

Voltage	±12 VDC to ±15 VDC
Current	±50mA plus load current (±250mA maximum)
Isolation	There is no isolation between inputs, outputs, and power supplies. If isolation is required, it must be done external to the VC2100 (Delta motion control modules provide isolated voltage outputs).
Protection	Power supply inputs are protected against over-voltage, spikes, and reverse voltage. User should fuse with appropriate fast-blow fuse.

Mechanical Specifications

Dimensions	0.94 x 2.94 x 3.94 in (2.4 x 7.5 x 10.0 cm)
Weight	3.5oz (100 g)
Mounting	Mounts directly to DIN rail
Connectors	Cage clamp terminal blocks integrated into package

Environment

Operating temp.	+32 to +140°F (0 to +60°C)
Storage temp.	-40 to +185°F (-40 to +85°C)

Communicating with the Quantum Controller

Communicating with the MMC120

Quantum Bus Configuration

Select four 3XXXXX registers for inputs and four 4XXXXX registers for outputs for each module. The MMC120 must be configured as an **MMC 120 0x** module in the I/O map. Depending on the Modicon software, **MMC 120 0x** may or may not be available as a device. It can be added using the steps below for your software:

Modsoft

Confirm that the **\MODSOFT\RUNTIME\GCNFTCOP.SYS** file has the following exact line, with the possible exception of the sequence number (156), which must be unique:

DCS MMC 120 0x,156**,0,08,08,2 AXES HYD MOTION,1,L0128,2,**

The your file does not have this line or has this line but it differs in more ways than just the sequence number, then it must be added to your file.

Concept 2.0

To add the MMC120 to Concept 2.0, copy **SYSINFDB.S0, SYSINFDB.S1, SYSINFDB.S2,** and **SYSINFDB.S3** from the **Concept20\Patch** subfolder of the folder to which you installed DCS120-Win to the **dat** subfolder of the folder to which you installed Concept 2.0. This will overwrite the existing Concept database files.

Concept 2.1

This version of Concept allows ModConnect partner devices such as the MMC120 to be easily added to the Concept device database through Module Description (.mdc) files. The MMC120's module description file is named mmc120.mdc and is installed to the DCS120-Win folder. Use the following steps to use this file from Concept 2.1:

1. Ensure that Concept 2.1 is not running.
2. Start Concept's ModConnect Tool.
3. On the **File** menu, click **Open Installation File**.
4. Navigate to the DCS120-Win directory, select mmc120.mdc, and click **OK**.
5. In the **Select Module** dialog box, click **Add All** and then **Close**.
6. On the **File** menu, click **Save Changes**.
7. Close the ModConnect Tool.

In the remainder of this chapter, 3TTTTT represents the base address for the input registers and 4TTTTT represents the base address for the output registers. These are the addresses at which the MMC120 is configured.

Memory Requirements

The MMC120 has non-volatile FLASH memory that can be used for parameter, profile and event step storage. If you also want to store the parameters in the Programmable Controller's memory you must reserve 32 words (registers) of memory—16 per axis.

Tip: Use the monitor program to set up and modify the parameters, profiles and event steps. Then move them to the programmable controller.

This memory contains the initialization parameters for the axis. The parameters are arranged in the following order:

0	CONFIGURATION Word
1	SCALE
2	OFFSET
3	EXTEND LIMIT
4	RETRACT LIMIT
5	PROPORTIONAL GAIN
6	INTEGRAL GAIN
7	DIFFERENTIAL GAIN
8	EXTEND FEED FORWARD
9	RETRACT FEED FORWARD
10	EXTEND ACCELERATION FEED FORWARD
11	RETRACT ACCELERATION FEED FORWARD
12	DEAD BAND ELIMINATOR
13	IN POSITION
14	FOLLOWING ERROR
15	AUTO STOP

In addition to the parameter storage blocks, memory may be allocated for profile and event step storage if these features are used.

Each profile requires four words of memory, and each module can use up to 16 profiles. The first eight profiles can be changed from the PLC and all 16 can be changed from the monitor program. Only enough memory must be allocated to hold the profiles used in all the modules in the system. The profiles consist of Mode, Acceleration, Deceleration and Speed information.

If you use Event Control and want to store the step sequences on the PLC, you must reserve eight registers for each step used.

The demo program on the monitor program disk shows an example of ladder logic for transferring parameters, profiles and Event Control steps to and from the MMC120.

See also:
Input Register Overview
Output Register Overview
Programmable Controller Commands

Input Register Overview

In addition to the Output registers, there are four 16-bit registers that are read from the MMC120 controller:

Register Number	Contents
3TTTTT + 0	STATUS for axis 1
3TTTTT + 1	Specified by command for axis 1
3TTTTT + 2	STATUS for axis 2
3TTTTT + 3	Specified by command for axis 2

In the table above, 3TTTTT is the Input base address. For most commands, the value returned in the second register for each axis is selected using the Status Area Request bits in the command register. Refer to the individual commands for exceptions. The possible fields that can be returned in the second register when is it controlled by the Status Area Request bits are:

Status Area Request	Read-back Register
0 (0000)	COMMAND POSITION
1 (0001)	TARGET POSITION
2 (0010)	ACTUAL POSITION
3 (0011)	TRANSDUCER COUNTS
4 (0100)	STATUS Word
5 (0101)	DRIVE OUTPUT
6 (0110)	ACTUAL SPEED
7 (0111)	NULL DRIVE
8 (1000)	CURRENT STEP
9 (1001)	LINK VALUE
10-15 (1010-1111)	Reserved

Example

Suppose you would like the STATUS and ACTUAL POSITION for axis 1, and you would like the STATUS and DRIVE for axis 2. You would send commands with the following format:

```
               SAR  CMND INDX (HEX)
4TTTTT+1 0000|0010|XXXX|XXXX          Requests ACTUAL POSITION
(02XX)
      +2 XXXX|XXXX|XXXX|XXXX          Data for command on axis 1
```

```
(XXXX)
4TTTTT+3 0000|0101|XXXX|XXXX        Requests DRIVE OUTPUT
(05XX)
      +4 XXXX|XXXX|XXXX|XXXX        Data for command on axis 2
(XXXX)
```
The following input registers would be returned:
```
               SAR  CMND INDX (HEX)
3TTTTT+1 XXXX|XXXX|XXXX|XXXX        STATUS of axis 1
(XXXX)
      +2 XXXX|XXXX|XXXX|XXXX        ACTUAL POSITION of axis 1
(XXXX)
3TTTTT+3 XXXX|XXXX|XXXX|XXXX        STATUS of axis 2
(XXXX)
      +4 XXXX|XXXX|XXXX|XXXX        DRIVE of axis 2
(XXXX)
```

See also:
Output Register Overview
Communicating with the MMC120

Programmable Controller Commands

Output Register Overview

In addition to the Input registers, there are four 16-bit registers that are sent to the MMC120 each time the I/O drop is accessed. The registers contain commands and data:

Register Number	**Contents**
4TTTTT + 0	Command – axis 1
4TTTTT + 1	Data out – axis 1
4TTTTT + 2	Command – axis 2
4TTTTT + 3	Data out – axis 2

The commands sent to the two axes are independent of each other.

Data should only be written to the module once per segment, otherwise new data overwrites previous data. Only the last information placed in the output registers is transferred to the module.

See also:
Input Register Overview
Communicating with the MMC120

Programmable Controller Commands

As described in Output Register Overview, each axis has one command register and one data out register. The data out register depends on the command being selected.

The command register is divided into four nibbles (one nibble is four bits or a half byte). There are two formats used. The standard commands use a most significant nibble of binary 0000 (hex 0). The following chart shows the use of each nibble:

								Command							
0	0	0	0	SAR				Type				Index			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

For a detailed description of Status Area Request (SAR), refer to Input Register Overview. The table below refers shows each command type. Each command index is described under the command type topic.

Command Type	**Binary Representation**	**Description**
0	0000	Reserved
1	0001	Go Using Profile
2	0010	Set Profile
3	0011	Set Parameter
4	0100	ASCII Commands
5	0101	ASCII Commands
6	0110	ASCII Commands
7	0111	ASCII Commands
8	1000	Reserved
9	1001	Open Loop Using Profile
A	1010	Get Profile
B	1011	Get Parameter
C	1100	Reserved
D	1101	Reserved
E	1110	Event Step Edit Commands
F	1111	Diagnostics

In addition to these standard commands, the following format is used for downloading and uploading Event Steps:

1	1	1	0	R/W	Event Step Number								Step Field		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

See Event Step Transfer Command for details on these commands.

See also:
Input Register Overview
Output Register Overview
Communicating with the MMC120

Support and Troubleshooting

Warranty

This motion controller shall be free from defects in materials and workmanship under normal and proper use and service for a period of fifteen (15) months from the date of shipment by Delta Computer Systems, Inc. (Delta) or Delta's authorized distributor so long as the module was under warranty when shipped to the customer by the distributor.

The obligation of Delta under this warranty shall be limited to repairing or replacing this motion controller or any part thereof which, in the opinion of Delta, shall be proved defective in materials or workmanship under normal use and service during the warranty period.

Repairs required because of obvious installation failures (burned resistors, traces, etc.) are not covered and will be billed at standard repair rates.

Disclaimer of other Warranties: There are no other representations or warranties made by Delta, express or implied. **Delta expressly disclaims any and all implied warranties, including any implied warranty of merchantability and any implied warranty of fitness for a particular purpose.** Further, Delta disclaims any liability for special, consequential or incidental damages resulting from any breach of warranty by Delta under this Agreement.

Module Return for Repair

Should a module require repair, refer to Technical Support for return details.

Troubleshooting

Programming Hints

The Programmable Controller is responsible for storing the initialization parameters used by the Motion Controller and initializing the Motion Control Module with those parameters.

The Motion Controller provides a STATUS word for each axis. If an error bit in the STATUS word, the Programmable Controller is responsible for shutting down the axis drive power. It must have a watchdog timer that will shut down the drives if a time-out occurs.

Write to the Motion Controller only once per scan, otherwise the newest data will overwrite the previous data.

Error Handling

The motion controller reports errors to the Programmable Controller within one control loop of detection. Errors are reported by setting bits in the affected axis's STATUS word and turning on the appropriate LEDS. The Programmable Controller is responsible for checking errors by reading the STATUS words. It is up to the Programmable Controller to determine what should be done if an error is detected.

The system must be able to shut down the axis drive power using a normally open output that is held closed when the system is running. This contact should be in series with an operator emergency off button. If power to the rack is lost, the contact will open and the axes will stop. If an error occurs in the motion controller the contact can be deactivated, which stops the axes. Usually the Programmable Controller will not take so drastic a step until it has determined that all control is lost. An Halt command to the axis with an error can take care of most error conditions.

When two axes are making a coordinated move and one axis starts moving slower than it should, it is best to issue a Halt command to both axes to stop all movement until the problem with the faulty axis has been resolved.

You can also use AUTO STOP detection by setting the appropriate bits in the SOFT STOP the HARD STOP bytes corresponding to the error bits in the STATUS word.

MMC120 Module Problems

Problems and Solutions

Control program cannot access parameters or operate module

1. Module not configured properly—active LED off. Configure the module as a 4-input and 4-output register module with binary format.
2. Make sure the Programmable Controller is accessing the correct I/O registers.

Red LEDS 2, 3, 4, 6, 7, or 8 are on

This indicates the transducer is not responding to the module. Every millisecond the module interrogates the transducer's position. If a return response is not seen after about two milliseconds, the internal counters overflow, and red LED 4 or 8 will be latched. If a return response is not seen after six milliseconds, red LED 2 or 6 will be latched. If the transducer count between interrogations changes too much (indicating noise), red LED 3 or 7 will be latched. Check the transducer power supply and check the wires to the transducer.

Red LEDS 2 and 6 are on; all other red LEDS are flashing; no green LEDS on

This indicates that the flash memory on the module has an invalid checksum. You must download a new copy of the firmware to the module.

During a move, the Actual Position is erratic

Electrical noise or a defective transducer is usually the cause of this problem. Monitor bits 1, 2, and 3 of the axis's STATUS word to determine if the module is detecting a transducer error. To reduce electrical noise check the following:

1. Make sure the transducer wiring is separated from all other wiring.
2. Add a termination resistor (220 ohm for Temposonics I) as close to the transducer as possible.
3. Connect the shield at the module end, the transducer end, or both.

During a move, the drive comes to a halt for no apparent reason

When the module detects a 'transducer not responding' error it makes a Hard Stop. See 'Actual Position is erratic' above for more information. If any of the following conditions are enabled, the axis will also halt:

Following Error
Overdrive Error
Position Overflow
Parameter Error
Integrator Windup

Transducer counts field not indicating transducer location

See "Red LEDS 2, 3, 4, 6, 7, or 8 are on" above.

Transducer counts field changes but output drive does not work

See "During a move, the drive comes to a halt for no apparent reason" above.

The System is unresponsive and hard to tune

This problem could have several causes. The first items to check are:

1. Is there hose, rather than rigid pipe, installed between the hydraulic valve and the cylinder? The hose expands and acts like an accumulator and the fluid goes to fill the hose rather than move the cylinder.
2. Does the valve have overlap? Overlap in hydraulic valves causes a significant dead band and slows the system response. Some proportional valve amplifier cards have

dead band eliminator circuits that make tuning easier.

3. If you have a servo motor and a ball screw, is there any backlash? Backlash produces a dead band when the axis changes direction, so the controller will tend to oscillate around the dead band.

The axis oscillates

This problem could have several causes. The first items to check are:

1. Make sure that the DEAD BAND ELIMINATOR value is not too high.
2. Try reducing the PROPORTIONAL, INTEGRAL, or DIFFERENTIAL GAINS.

The axis does not finish moves or moves differently than expected

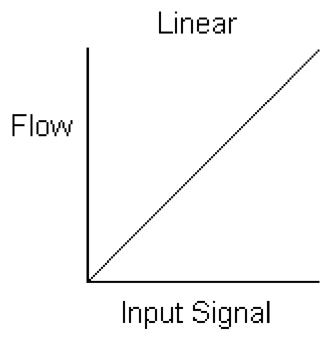
This can be caused by the P/C issuing unintended commands to the module. Use the Command Log to monitor commands sent by the P/C to the module. Confirm that only the expected commands are being sent.

Hydraulic System Problems

These hydraulic system problems can make system tuning difficult or impossible.

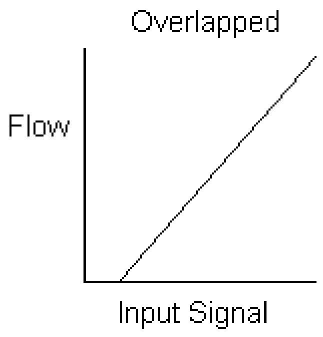
Nonlinear Valves

A valve is linear when the flow through it is directly proportional to the input signal over the entire range of the input signal:

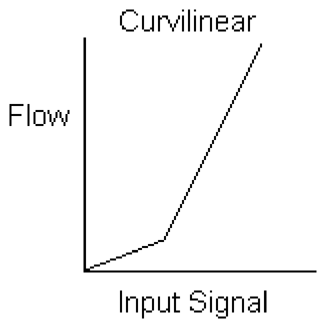


It is nonlinear when the output is not directly proportional to the input. You may find two types of valve non-linearity:

Overlapped valves - Oil does not start to flow through these valves until the spool has moved some distance. This causes a dead band in the system, where small amounts of drive do not produce motion. Overlapped valves are designed for manual and on/off type control and are not suited for servo control. These valves should be replaced with non-overlapped valves.



Curvilinear valves - The flow through these valves increases slowly as the input signal increases for the first 20% of range. Beyond 20% the flow increases rapidly as the input increases. This is equivalent to having two different gains for different signal levels. The low gain at low flow causes poor response at slow speeds, and the high gain at high speed can cause instability. These problems are more pronounced when heavy loads are moved by relatively small cylinders.



Slow-Response Valves

Valves with slow response cause the Motion Control Module (MCM) to overcompensate for disturbances in the motion of the system. Since the system does not respond immediately to the control signal, the MCM continues to increase the drive signal. By the time the system begins to respond to the error, the control signal has become too large and the system overshoots. The MCM then attempts to control in the opposite direction, but again it overshoots. These valves can cause the system to oscillate around the set point as the MCM overshoots first in one direction, then the other.

Hoses

Long hoses between the valves and cylinder act as accumulators and make the system respond as if it has a spring in it (imagine trying to control the position of one end of a Slinky™ by moving the other end!). The lines between the valves and cylinders must be as short and rigid as possible.

Pumps and Accumulators

Insufficient pump and/or accumulator capacity will cause the system response to degrade during a move because the effective pressure drops.

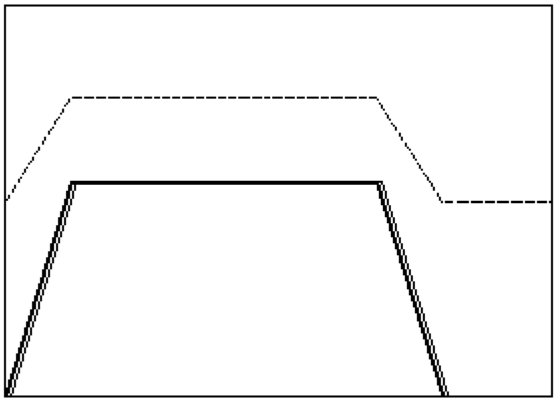
Pressure transients due to insufficient accumulator volume cause jerky motion, particularly during starts and stops.

NOTE: Even systems with 'fast' pumps usually require at least a small accumulator near the cylinder to maintain the constant pressure needed to get smooth motion.

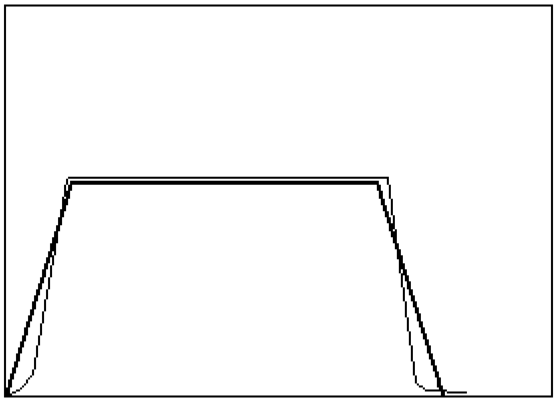
Insufficient pump capacity can result in inadequate control when moving many axes simultaneously or when making long moves. In these cases pressure can drop so much a fully open valve cannot maintain the requested speed.

Identification and Correction

To identify and correct these problems, make a move with very low (or zero) gains except the FEED FORWARD terms. Graph the move. The graph (ignoring the Position terms) should show:

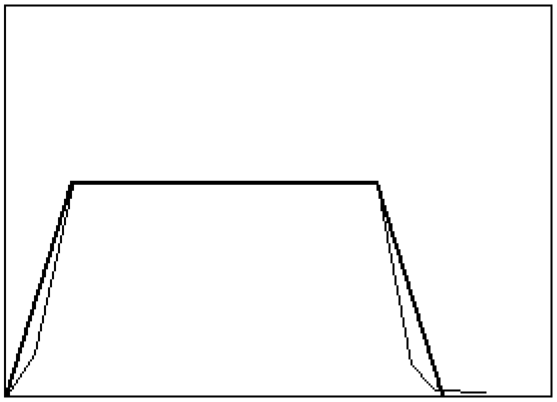


If the actual speed and target speed show:



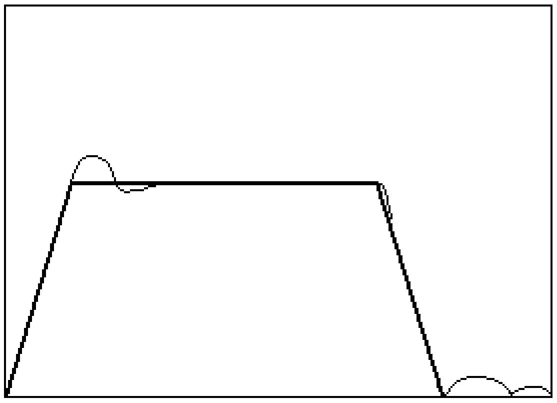
Your valve probably has overlap. Replace the valve with a linear one or try increasing the DEAD BAND ELIMINATOR value.

If the speeds show:



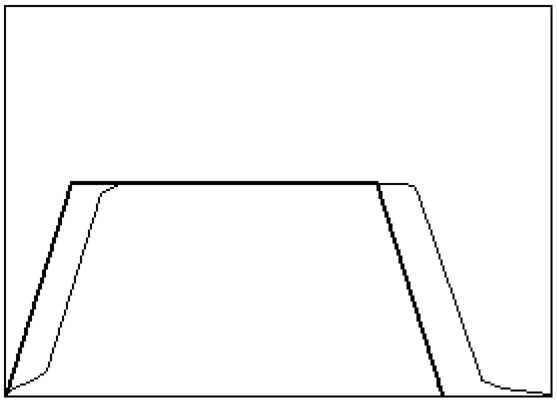
Your valve is probably curvilinear. Replace the valve with a linear one or increase the proportional gain and tune the system for high-speed stability; expect poor control at low speed and when stopped.

If the speeds show:



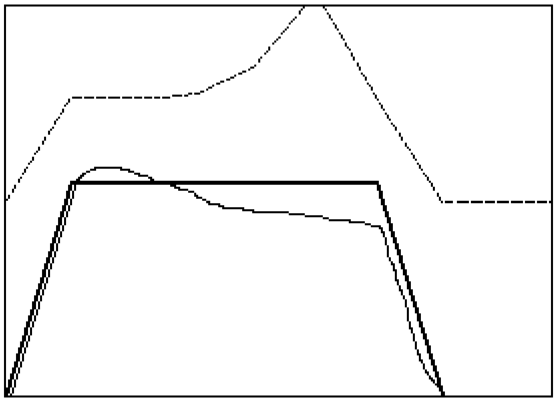
You may have too much hose between the valve and the cylinder. Reduce the amount of hose or add differential gain (usually less than 5).

If the speeds show:



Your valve may have slow response. Change to a faster valve or add ACCELERATION FEED FORWARD.

With normal gain values, if the graph shows:



Your pump and/or accumulator may be inadequate (you are running out of oil). Reduce speed, increase pump pressure, add accumulator volume, or get a bigger pump.

Technical Support

Technical Support

Delta Technical Support Numbers

Phone: **360-254-8688 (24-hour emergency support available)**

Fax: **360-254-5435**

http://www.deltacompsys.com

email@deltacompsys.com

Module Return for Repair

If you need to return the motion control module for repair, please contact Delta prior to shipment for an RMA number. Returned modules must be packaged in static protection material and have the RMA number clearly marked on the outside of the package. Please include a short note explaining the problem. Send the module to:

Delta Computer Systems, Inc.
1818 SE 17th St
Battle Ground, WA 98604

Parameter Errors

Target position moved outside limits

This parameter error indicates that the axis was about to move its Target Position outside the extend or retract limit. The target position is truncated to the limit when this error occurs. Commands that have a requested position outside the limits are caught immediately by the "Attempt to go beyond extend limit" and "Attempt to go beyond the retract limit" parameter errors.

Therefore, the only way this error can occur is by lowering the deceleration rate in the middle of a move enough so that the axis overshoots the commanded position and limit. We could catch this error when the command is issued but do not because some applications have an axis decelerate slowly for a period of time, but then issue another command to stop it within the limits. These applications would be impossible if we didn't allow this deceleration even though, if uninterrupted, it would bring the axis outside the limit.

NOTE: This error will not occur when the Actual Position goes outside the limits. That is, you will be able to control position at either limit even though the Actual Position may go outside the limits temporarily.

FLASH contained no data on startup

This parameter error will appear on either the first or second axis on startup whenever there is a problem loading the parameters or event steps from FLASH memory. If this parameter occurs on axis 1, then profiles and parameters could not be loaded. If this parameter occurs on axis 2, then event steps could not be loaded. Default will be used. To store parameters in the FLASH, load the motion controller with all desired configuration parameters, and issue a Update FLASH command. You must wait for the ACK to toggle—indicating that the FLASH write is complete—before powering off the module. If this parameter error appears, one of the following has occurred:

- The module is newly shipped. New modules have no data stored in the FLASH.
- The module was powered down while updating the FLASH. This results in an invalid checksum being found for the data, so the FLASH is not usable on next power-up.
- The FLASH itself has a problem with it. If you find that the FLASH consistently does not store data, although the module is not being powered down while the FLASH is being written to, contact the manufacturer for details on having the FLASH replaced.

Attempt to go beyond extend limit

This parameter error will occur whenever either a Go or Relative Move command requests that the axis moves to a position beyond the EXTEND LIMIT. The COMMAND POSITION will be set to the EXTEND LIMIT. The move will continue if the bits for the Parameter Error are cleared in the AUTO STOP word. To remedy this problem:

- Fix the control program to avoid issuing illegal commands.
- If the EXTEND LIMIT is set incorrect, fix its value.

Attempt to go beyond retract limit

This parameter error will occur whenever either a Go or Relative Move command requests that the axis moves to a position beyond the RETRACT LIMIT. The COMMAND POSITION will be set to the RETRACT LIMIT. The move will continue if the bits for the Parameter Error are cleared in the AUTO STOP word. To remedy this problem:

- Fix the control program to avoid issuing illegal commands.
- If the RETRACT LIMIT is set incorrect, fix its value.

Requested drive too large

This parameter error indicates that requested drive used by either an Open Loop command or a move command using Quick Mode is too large. The maximum allowed drive is ±12000 millivolts, although values this large can easily cause an Overdrive Error.

Invalid command value

This parameter error is set if either the Integral Drive or Null Drive is set to a value greater than ±10000 with the Set Integral Drive and Set Null Drive commands.

Invalid step number given in "Start Events" command

Because there are only 256 event steps—numbered 0 to 255—any COMMAND VALUE outside of that range with the Start Events command will generate this error.

Invalid scale value

This parameter error indicates that a SCALE was set to an invalid value: either 0 or 32768. If you are having a difficult time setting the SCALE parameter, you may want to try using the Scale/Offset Calibration Utilities.

Extend limit must be greater than retract limit

This parameter error indicates that the EXTEND LIMIT was set to a position in which the TRANSDUCER COUNTS would be less than or equal to those at the RETRACT LIMIT position. Therefore, if the SCALE is positive, this error indicates that the EXTEND LIMIT is less than the RETRACT LIMIT. Otherwise, this error indicates that the EXTEND LIMIT is greater than the RETRACT LIMIT.

Dead band eliminator out of range

This parameter error indicates that the DEAD BAND ELIMINATOR parameter was set to a value greater than ±2000. This maximum prevents a dead band of greater than 2 volts. If a valve still does not work to satisfaction with this large of a dead band, then the valve is most like not appropriate for the job. Refer to Hydraulic System Problems for details.

Invalid command received

This parameter error indicates that the COMMAND is invalid. Refer to that topic for a list of valid commands.

Move would cause discontinuity

This parameter error is generated when the requested move fields would have generated a discontinuity in the speed. This occurs if the axis is stopped, and the user specifies an ACCELERATION distance of 0 on a move to another location. In order to carry out this request, the motion controller would have to instantaneously increase the speed to the maximum speed. This would generate a discontinuity, and therefore instead the move is rejected and this parameter error is generated. To fix this problem:

- Verify that your control program is sending the command correctly.
- If you really want to accelerate instantaneously to the maximum speed, you can use a very high ACCELERATION in Acceleration Modes 0 and 1, or a very low ACCELERATION in Acceleration Modes 2 and 3.

The acceleration or deceleration ramp is too slow

When a new move command is sent to the motion controller, the motion controller must calculate the distance that each of the ramps (acceleration and deceleration) will take. If this calculation overflows, this parameter error is generated. The overflow occurs when the calculated ramp would take longer than 65535 position units. This overflow can occur even if the axis is moving a much shorter distance. There are two ways to fix this problem:

- Increase the ACCELERATION and/or DECELERATION values.

- Decrease the maximum SPEED value.

The command acceleration is invalid

This parameter error indicates that an invalid ACCELERATION was given for an Open Loop command. The ACCELERATION must be less than 20000 millivolts per millisecond in Acceleration Mode 1, and greater than zero in Acceleration Mode 3.

The command deceleration is invalid

This parameter error indicates that an invalid DECELERATION was given for an Open Loop command. The DECELERATION must be less than 20000 millivolts per millisecond in Deceleration Mode 1, and greater than zero in Deceleration Mode 3.

Both sync bits cannot be set in the "Mode" word

There are two Sync bits in the MODE word. Only one or neither can be set at a time. This parameter error is generated if both are set.

One or more synced axes are uninitialized

This error indicates that at least one of the axes selected to participate in the synchronized move has not been initialized. The axis that is not initialized will be marked with the Parameter Error bit and will not have the Parameters Initialized bit set in the STATUS word. To initialize the parameters on this axis, issue a Set Parameters command.

Incompatible sync mode words

This error indicates that the low eight bits of the mode words do not match between a set of synchronized axes. These bits are required to match to ensure that the same type of move is executed between all of the synchronized axes. Decide which MODE bits you would like to use, and use the same mode bits for all synchronized axes. Because only the low eight bits are required to match, the Graph Disable bit can be different between synchronized axes.

"Event Step Edit" indices are invalid

The Programmable Controller sets the beginning and ending step numbers to edit with the E0 and E1 Event Step Edit commands. When the E3 Event Step Edit command is issued—which actually performs the step table modifications—the beginning and ending step numbers are checked. Both must hold valid step numbers (between 0 and 255, inclusive), and the ending step number must be greater than or equal to the starting step number.

If the beginning and ending step numbers are out of range or reversed, this parameter error is generated.

Unknown Parameter Error

Automatic parameter identification is a feature which has been added to these motion controllers in firmware dated 19971016 (year-month-day) and newer. If you have firmware older than this date and would like to use this feature, contact Delta Computer Systems, Inc. technical support and ask for the latest firmware. You can check your controller firmware date by selecting **Module Configuration…** from the **Tools** menu.

If your firmware is newer than the above date, and a parameter error is still not known, contact Delta Computer Systems, Inc. technical support with the parameter error number. It will most likely be necessary to obtain a newer version of the monitor program that recognizes the new parameter error number.

Otherwise, you can attempt to trace down the error by looking at the command that generated the parameter error and compare your usage of the command with the help topic for the particular Command. To clear a parameter error, issue a Set Parameters command to the axis. If this does not clear the error, then one of the sixteen axis parameters has an invalid value.

Appendix A: Command Reference

General ASCII Commands

Change Acceleration Command

Character: A

Decimal: 65

Hexadecimal: 041

Command Value: New ACCELERATION value (units depend on Accel/Decel Mode in the MODE word)

This command sets the ACCELERATION parameter to the value in the COMMAND VALUE field. The change in acceleration takes place immediately if a move is in progress.

Change Deceleration Command

Character: D

Decimal: 68

Hexadecimal: 044

Command Value: New DECELERATION value (units depend on Accel/Decel Mode in the MODE word)

This command sets the DECELERATION parameter to the value in the COMMAND VALUE field. The change in deceleration takes place immediately if a move is in progress.

Start Events Command

Character: E

Decimal: 69

Hexadecimal: 045

Command Value: Event Step to Start Execution

This command starts an event step sequence at the step specified in the COMMAND VALUE field. If the axis already was in the middle of an event sequence, that sequence will end and the new event sequence will begin.

Set Feed Forward Command

Character: F

Decimal: 70

Hexadecimal: 046

Command Value: Unused

The 'F' command is used to automatically set the feed forward values. After a move is made where the axis is allowed to reach constant velocity and the overdrive bit is not set, an 'F' command will set the FEED FORWARD for the direction last moved. This command is quick and easy, and it allows the system to adjust for changing system dynamics. This also makes setup easier.

NOTE: The 'F' command should be used only after the axis is moving smoothly. If the axis oscillates or doesn’t reach steady state during the move, this command will give erroneous results.

If the NULL DRIVE is not adjusted correctly, the value for the Feed Forward will be incorrect.

Go Command

Character: G or g

Decimal: 71 or 103

Hexadecimal: 047 or 067

Command Value: Requested Position, in position units

This command requests the axis to move to the requested position specified in the COMMAND VALUE field. The user must make sure that all parameter words are valid when the Go command is issued. Normally, once the MODE, ACCELERATION, DECELERATION, and SPEED are set, only the COMMAND VALUE needs to be changed. Once set, a 'G' put in the COMMAND word is used to get the axis to move.

NOTE: The 'G' command can be given while the axis is in motion. If you do this, the motion controller will ramp to the new speed at the rate specified by the ACCELERATION and DECELERATION parameters.

Halt Command

Character: H or h

Decimal: 72 or 104

Hexadecimal: 048 or 068

Command Value: Unused

The HALT command is used for a Soft Stop, jogging the axis, or when the drive power is off. Putting an 'H' in the Command word while the axis is moving will cause the axis to ramp down until it stops. Jogging an axis is accomplished by alternating Go and HALT commands. The 'H' command disables the integral gain term and the null update.

If an event sequence is in progress when this command is issued, the event sequence will be stopped. Compare this command with the Disable Drive Output and Quit Events commands.

NOTE: Issue a HALT command when hydraulic power is turned off. This prevents the integrator from winding up.

Set Integral Drive Command

Character: I (Upper case i)

Decimal: 73

Hexadecimal: 049

Command Value: New Integral Drive, in millivolts

This command sets the Integral Drive to the value in the COMMAND VALUE field. This command can be used to unwind the integrator. There are several alternative ways to unwind the integrator. See also the Set Integral Drive to Null Drive, Save Integral Drive and Restore Integral Drive commands.

Set Integral Drive to Null Drive Command

Character: i
Decimal: 105
Hexadecimal: 069
Command Value: Unused

This command sets the Integral Drive to the Null Drive value. This can be used to unwind the integrator. In most cases this is more desirable than using the Set Integral Drive command because this accounts for the valve being non-nulled. You can also use the Save Integral Drive and Restore Integral Drive commands to unwind the integrator.

Relative Move Command

Character: J or j
Decimal: 74 or 106
Hexadecimal: 04A or 06A
Command Value: Change in Position, in position units.

This command changes the COMMAND POSITION by the amount specified in the COMMAND VALUE field. For example, suppose that one position unit is equal to 0.001" and the current position is 5000 (5.0"). If a J command is given with a COMMAND VALUE of –1000, then the axis will move to 4000 (4.0").

Disable Drive Output Command

Character: K
Decimal: 75
Hexadecimal: 04B
Command Value: Unused

This command immediately sets the drive output on the module to the current null value. This is equivalent to a Hard Stop. The output will remain at null in open loop control until a new command is issued.

When this command is issued from the monitor program, it is issued to all axes simultaneously. When issued from the controller, only the axis receiving this command is affected.

If an event sequence is in progress when this command is issued, the event sequence will be stopped. Compare this command with the Halt and Quit Events commands.

Set Mode Command

Character: M
Decimal: 77
Hexadecimal: 04D
Command Value: New MODE value

This command sets the MODE to the value in the COMMAND VALUE field.

Set Null Drive Command

Character: N
Decimal: 78
Hexadecimal: 04E
Command Value: Millivolts of Drive

The 'N' command sets the Null Drive to the value in the COMMAND VALUE field.

Set Null Drive to Integral Drive Command

Character: n
Decimal: 110
Hexadecimal: 06E
Command Value: Unused

This command sets the Null Drive to the current Integral Drive value.

Why Bother?

It is important that the Null Drive be set because it is used in open loop mode and hard stops. The ‘n’ command should only be issued when the axis is in position and its speed is zero.

Open Loop Command

Character: O
Decimal: 79
Hexadecimal: 04F
Command Value: Millivolts of Drive

CAUTION: Use this command with care! Open Loop operation disables all safety features on the motion controller!

NOTE: The open loop command will not affect drive output while the Simulate bit in the Config word. Drive output is always 0 volts in simulate mode.

The Open Loop command allows the Controller to directly specify values for the analog output. The output range is -10000 to 10000 where -10000 is -10 volts and 10000 is +10 volts.

The O command uses the following parameters from the last open loop profile specified:

The COMMAND VALUE field specifies the desired drive in millivolts to output. The current null drive is added to this value.

The SPEED field is not used.

The ACCELERATION and DECELERATION fields control the rate at which the drive output ramps to the requested value. Acceleration is used when the drive output is moving away from 0 and deceleration is used when drive output is moving toward 0. The actual meaning of the values depends on the Acceleration/Deceleration mode bits in the MODE word. Notice that only modes 1 and 3 are used:

Mode 1 -Acceleration and deceleration are given in millivolts per millisecond. For example, if the current drive output is 0 millivolts, and the Command value is 1000 millivolts, and the Acceleration is 50, then it will take 20 milliseconds for the drive output to reach 1000 millivolts.

Mode 3 -Acceleration and deceleration are given in milliseconds. For example, if the current drive output is 0 millivolts, the Command value is 1000 millivolts, and the Acceleration is 50, then it will take 50 milliseconds for the drive output to reach 1000 millivolts, as the drive will increase 20 millivolts each millisecond.

NOTE: If a move cross 0 drive, then Deceleration will be used until the drive reaches 0, and then Acceleration would be used until the final drive is reached.

Tip: If you are going to be changing from closed loop to open loop, then you need to be aware of the following trickiness. While in closed loop and holding a position, the drive may be switching back and forth between small positive and negative drives. If an open loop command is given to go to a non-zero drive, then it may happen that the starting drive has an opposite sign than the requested drive, and therefore the drive will decelerate down to zero millivolts and then accelerate up to the new drive. For example, if the current drive is –10 millivolts and an open loop command requests a drive of 2000 millivolts (2V), the drive will ramp from –10 to 0 and then accelerate from 0 to 2000mV. Therefore, you should ensure that the deceleration is set so that this deceleration happens quickly (by setting it to 0 in Mode 3, and to a high number (1000 or greater) in Mode 1.

Set Parameters Command

Character: P or p
Decimal: 80 or 112
Hexadecimal: 050 or 080
Command Value: Unused

When a 'P' command is given all initialization parameters are updated. The minimum requirement of this command is to set the EXTEND and RETRACT LIMITS to their proper values (see Start-Up and Tuning). When a 'P' command is given, the motion controller will copy the ACTUAL POSITION of the axis into the TARGET and COMMAND POSITIONS.

Quit Events Command

Character: Q
Decimal: 81
Hexadecimal: 051
Command Value: Unused

This command stops the event control sequence. Any motion commands in progress are not stopped. Therefore, if an axis is currently moving to 4.000", it will continue moving to that position; if an axis is in open loop with 1.000V drive, the drive will continue holding that drive, etc. If you desire to stop the axis at the same time, look at the Halt and Disable Drive Output commands.

Restore Null Drive Command

Character: R
Decimal: 82
Hexadecimal: 052
Command Value: Unused

This command restores the last saved value of null. This value will be 0 if no previous Save Null Drive was issued.

Restore Integral Drive Command

Character: r
Decimal: 114
Hexadecimal: 072

Command Value: Unused

This command restores a previously saved Integral Drive value. The Integral Drive value that is restored is taken from a buffer internal to the motion controller that will set using the Save Integral Drive command. The Integral Drive will be set to 0 if no previous Save Integral Drive command was issued. The Save/Restore Integral Drive pair can be used to unwind the integrator while in the middle of a move.

For example, suppose the integral drive is saved while the move is taking place. If during the move, the axis gets stuck and the integrator winds up, then the integral drive can be restored after the cause of the stall is fixed to avoid an overshoot.

For alternative ways to unwind the integrator, see the Set Integral Drive and Set Integral Drive to Null Drive commands.

Save Null Drive Command

Character: S
Decimal: 83
Hexadecimal: 053
Command Value: Unused

The Save Null Drive command saves the current value of the null so it can be recalled later with a Restore Null Drive command ('R').

Save Integral Drive Command

Character: s (lower case S)
Decimal: 115
Hexadecimal: 073
Command Value: Unused

This command saves the current value of the Integral Drive so it can be recalled later with a Restore Integral Drive command ('r'). For details on the uses of these two commands, see Restore Integral Drive.

Update FLASH Command

Character: U
Decimal: 85
Hexadecimal: 055
Command Value: Data to Save (1=Parameters and Profiles; 2=Event Steps)

This command instructs the motion controller to write data to FLASH for storage in case of power loss or reset. Is the COMMAND VALUE is 1 when this command is executed, Parameters and Profiles are saved to FLASH. If the COMMAND VALUE is 2 when this command is executed, the Step Table is saved to FLASH. Do not use a COMMAND VALUE greater than 2 with this command.

The data stored by each of these two COMMAND VALUE are stored in separate locations, so one does not overwrite the other. If this command is successful, the Acknowledge bit of axis 1 is toggled. If this command fails, then the Acknowledge bit of axis 2 is toggled.

Set Speed (Unsigned) Command

Character: V

Decimal: 86

Hexadecimal: 056

Command Value: New SPEED Value

This command sets Speed to the value in the Command Value field. This will cause the current move to ramp up or down to the new speed. This command does not change the Command Position of the axis.

If you wish to simply jog in one direction or another under closed loop control, see the Set Speed (Signed) command.

Set Speed (Signed) Command

Character: v

Decimal: 118

Hexadecimal: 076

Command Value: New SPEED Value (Signed)

This command sets the SPEED of the axis to the COMMAND VALUE, which is a value between –32,768 and 32,767. In addition, the axis given a COMMAND POSITION of either the extend or retract limit depending on the sign of the COMMAND VALUE.

If the COMMAND VALUE is positive, the axis will move in the direction of *increasing position units*. If the COMMAND VALUE is negative, the axis will move in the direction of *decreasing position units*.

If you wish to change only the speed of a move in progress, you may wish to use the Set Speed (Unsigned) command instead.

Start a Graph Command

Character: y

Decimal: 121
Hexadecimal: 079
Command Value: Unused

This command begins a new graph immediately. A move can, but doesn't have to, be in progress.

Programmable Controller Commands

Go Using Profile Commands

Format: 0000 RRRR 0001 NNNN

- **R** Used for Status Area Request
- **N** Used for Command Index described below

These commands allow the Programmable Controller to tell the motion controller to move the axis using stored profiles. These are generally the most commonly used commands. The Go Using Profile command operates identical to a standard Go command except that the move uses the MODE, ACCEL, DECEL and SPEED fields from the selected profile.
The format of the Command Register for the Go Using Profile commands is given below:

```
          |    | 111|1111
BIT # 1234|5678|9012|3456
      -------------------
HEX       |SAR |CMND|INDX
VALUE -------------------
0X10  0000|XXXX|0001|0000 GO USING PROFILE 0
0X11  0000|XXXX|0001|0001 GO USING PROFILE 1
0X12  0000|XXXX|0001|0010 GO USING PROFILE 2
0X13  0000|XXXX|0001|0011 GO USING PROFILE 3
0X14  0000|XXXX|0001|0100 GO USING PROFILE 4
0X15  0000|XXXX|0001|0101 GO USING PROFILE 5
0X16  0000|XXXX|0001|0110 GO USING PROFILE 6
0X17  0000|XXXX|0001|0111 GO USING PROFILE 7
0X18  0000|XXXX|0001|1000 GO USING PROFILE 8
0X19  0000|XXXX|0001|1001 GO USING PROFILE 9
0X1A  0000|XXXX|0001|1010 GO USING PROFILE 10
0X1B  0000|XXXX|0001|1011 GO USING PROFILE 11
0X1C  0000|XXXX|0001|1100 GO USING PROFILE 12
0X1D  0000|XXXX|0001|1101 GO USING PROFILE 13
0X1E  0000|XXXX|0001|1110 GO USING PROFILE 14
0X1F  0000|XXXX|0001|1111 GO USING PROFILE 15
```

The data for the command Output register represents the requested position in position units.

Example

Suppose you would like to move axis 1 to 5000 position units using profile 2 and axis 2 to 10000 position units using profile 5. You would send commands with the following format:

```
                 SAR  CMND INDX (HEX)
4TTTTT+1 0000|XXXX|0001|0010 (0X12)   Axis 1 Go using profile 2
      +2                5000 (1388)   Axis 1 Requested Position
4TTTTT+3 0000|XXXX|0001|0101 (0X15)   Axis 2 Go using profile 5
      +4               10000 (2710)   Axis 2 Requested Position
```

Set Profile Commands

Format: 0000 RRRR 0010 NNNN

R Used for Status Area Request

N Used for Command Index described below

These commands allow the programmer to change motion profiles stored in the motion controller. Only one value in one profile can be changed per axis each time the synchronization register is changed, but the profiles can be changed while the axis is moving. The new profile will be used by the next Go Using Profile command specifying that profile. Recall that new Go and Go Using Profile commands can be given while the axis is moving.

The format of the Command Register for the Set Profile commands is given below:

```
          |    | 111|1111
BIT # 1234|5678|9012|3456
      -------------------
HEX       |SAR |CMND|INDX
VALUE -------------------
0X20  0000|XXXX|0010|0000 SET PROFILE 0 or 4 MODE
0X21  0000|XXXX|0010|0001 SET PROFILE 0 or 4 ACCEL
0X22  0000|XXXX|0010|0010 SET PROFILE 0 or 4 DECEL
0X23  0000|XXXX|0010|0011 SET PROFILE 0 or 4 SPEED
0X24  0000|XXXX|0010|0100 SET PROFILE 1 or 5 MODE
0X25  0000|XXXX|0010|0101 SET PROFILE 1 or 5 ACCEL
0X26  0000|XXXX|0010|0110 SET PROFILE 1 or 5 DECEL
0X27  0000|XXXX|0010|0111 SET PROFILE 1 or 5 SPEED
0X28  0000|XXXX|0010|1000 SET PROFILE 2 or 6 MODE
0X29  0000|XXXX|0010|1001 SET PROFILE 2 or 6 ACCEL
```

```
0X2A  0000|XXXX|0010|1010 SET PROFILE 2 or 6 DECEL
0X2B  0000|XXXX|0010|1011 SET PROFILE 2 or 6 SPEED
0X2C  0000|XXXX|0010|1100 SET PROFILE 3 or 7 MODE
0X2D  0000|XXXX|0010|1101 SET PROFILE 3 or 7 ACCEL
0X2E  0000|XXXX|0010|1110 SET PROFILE 3 or 7 DECEL
0X2F  0000|XXXX|0010|1111 SET PROFILE 3 or 7 SPEED
```

Notice that for each command, two different profile numbers are listed. The actual profile affected is determined by the axis issuing the command. The following chart show which commands affect which profiles when issued on a particular axis (Notice that only the first eight (8) profiles can be modified from the Programmable Controller):

COMMAND	AXIS 1	AXIS 2
0x20-0x23	0	4
0x24-0x27	1	5
0x28-0x2B	2	6
0x2C-0x2F	3	7

The data for the command Output register represents the value of the field to be set in the profile.

Example:

We wish to set profiles 2 and 7 with the following values:

	PROFILE 2	PROFILE 7
MODE	00001	00001
ACCEL	100	150
DECEL	70	70
SPEED	12000	20000

Looking at the chart above, we can see that we issue Set Profile commands to axis 0 to set profile 2 and we use axis 1 Set Profile commands to set profile 7. Therefore, we can do both at the same time. This would take four scans because we can send one word of each profile on each axis per scan:

First scan:`4TTTTT+0 0X28h (Set profile 2 MODE)`

```
4TTTTT+1 0001h (Value of profile 2 MODE)
4TTTTT+2 0X2Ch (Set profile 7 MODE)
4TTTTT+3 0001h (Value of profile 7 MODE)
```

Second scan:`4TTTTT+0 0X29h (Set profile 2 ACCEL)`

```
4TTTTT+1   100 (Value of profile 2 ACCEL)
4TTTTT+2 0X2Dh (Set profile 7 ACCEL)
```

```
4TTTTT+3   150 (Value of profile 7 ACCEL)
```

Third scan:
```
           4TTTTT+0 0X2Ah (Set profile 2 DECEL)
           4TTTTT+1    70 (Value of profile 2 DECEL)
           4TTTTT+2 0X2Eh (Set profile 7 DECEL)
           4TTTTT+3    70 (Value of profile 7 DECEL)
```

Fourth scan:
```
           4TTTTT+0 0X2Bh (Set profile 2 SPEED)
           4TTTTT+1 12000 (Value of profile 2 SPEED)
           4TTTTT+2 0X2Fh (Set profile 7 SPEED)
           4TTTTT+3 20000 (Value of profile 7 SPEED)
```

Set Parameter Commands

Format: 0000 RRRR 0011 NNNN

R	Used for Status Area Request
N	Used for Command Index described below

These commands allow the Programmable Controller to download new initialization parameters to the motion controller. New parameters can be downloaded at any time, but they do not take effect until a 'P' ASCII command is issued. An axis must be stopped when issuing the 'P' ASCII command.

The format of the Command Register for the Set Parameter commands is given below:

```
          |    | 111|1111
BIT # 1234|5678|9012|3456
      -------------------
HEX       |SAR |CMND|INDX
VALUE -------------------
0X30  0000|XXXX|0011|0000 Set CONFIGURATION
0X31  0000|XXXX|0011|0001 Set SCALE
0X32  0000|XXXX|0011|0010 Set OFFSET
0X33  0000|XXXX|0011|0011 Set EXTEND LIMIT
0X34  0000|XXXX|0011|0100 Set RETRACT LIMIT
0X35  0000|XXXX|0011|0101 Set PROPORTIONAL GAIN
0X36  0000|XXXX|0011|0110 Set INTEGRAL GAIN
0X37  0000|XXXX|0011|0111 Set DIFFERENTIAL GAIN
0X38  0000|XXXX|0011|1000 Set EXTEND FEED FORWARD
0X39  0000|XXXX|0011|1001 Set RETRACT FEED FORWARD
0X3A  0000|XXXX|0011|1010 Set EXTEND ACCEL FEED FORWARD
0X3B  0000|XXXX|0011|1011 Set RETRACT ACCEL FEED FORWARD
0X3C  0000|XXXX|0011|1100 Set DEAD BAND ELIMINATOR
0X3D  0000|XXXX|0011|1101 Set IN POSITION
```

```
0X3E  0000|XXXX|0011|1110 Set FOLLOWING ERROR
0X3F  0000|XXXX|0011|1111 Set AUTO STOP
```

The data for the command Output register represents the new parameter value.

Example

Suppose you would like to set the EXTEND LIMIT for both axes. You would send commands with the following format:

```
                 SAR  CMND INDX (HEX)
4TTTTT+0 0000|XXXX|0011|0011 (0X33)   Sets axis 1 EXTEND LIMIT
      +1               24000 (5DC0)   New Extend Limit value
4TTTTT+2 0000|XXXX|0011|0011 (0X33)   Sets axis 2 EXTEND LIMIT
      +3               12000 (2EE0)   New Extend Limit value
```

ASCII Commands

Format: 0000 RRRR 01NN NNNN

R	Used for Status Area Request
N	Used for Command Index described below

These commands can be used to send any command which can be sent from the monitor program. These commands are listed in the Command Field topic. To send an ASCII command, just use the hex value for the commands listed in the table in the Command Field topic and use the desired Status Area Request value. For example, to send a Change Deceleration command (which has an ASCII hexadecimal value of 044) using a Status Area Request value of 2, you would send a command register of 0244.

The data for the command Output register represents the Command Value of the ASCII command being sent, if one is required.

Open Loop Using Profile Commands

Format: 0000 RRRR 1001 NNNN

R	Used for Status Area Request
N	Used for Command Index described below

These commands allow the Quantum to tell the MMC120 to change the output drive to a specified value with respect to null. The drive output will change at a rate specified by the select pre-stored profile.

NOTE: This command shares the profile table used by the Go Using Profile commands. Profiles used for one type of command should NOT be used for the other.

These commands behave identical to the Open Loop ASCII command, except that the specified stored profile information is copied into the active MODE, ACCEL, DECEL and SPEED first. Recall that the SPEED field of the profile is not used by the Open Loop. Refer to that command for further details.

The format of the Command Register for the Open Loop Using Profile commands is given below:

```
          |    | 111|1111
BIT # 1234|5678|9012|3456
      -------------------
HEX       |SAR |CMND|INDX
VALUE -------------------
0X90  0000|XXXX|1001|0000 OPEN LOOP USING PROFILE 0
0X91  0000|XXXX|1001|0001 OPEN LOOP USING PROFILE 1
0X92  0000|XXXX|1001|0010 OPEN LOOP USING PROFILE 2
0X93  0000|XXXX|1001|0011 OPEN LOOP USING PROFILE 3
0X94  0000|XXXX|1001|0100 OPEN LOOP USING PROFILE 4
0X95  0000|XXXX|1001|0101 OPEN LOOP USING PROFILE 5
0X96  0000|XXXX|1001|0110 OPEN LOOP USING PROFILE 6
0X97  0000|XXXX|1001|0111 OPEN LOOP USING PROFILE 7
0X98  0000|XXXX|1001|1000 OPEN LOOP USING PROFILE 8
0X99  0000|XXXX|1001|1001 OPEN LOOP USING PROFILE 9
0X9A  0000|XXXX|1001|1010 OPEN LOOP USING PROFILE 10
0X9B  0000|XXXX|1001|1011 OPEN LOOP USING PROFILE 11
0X9C  0000|XXXX|1001|1100 OPEN LOOP USING PROFILE 12
0X9D  0000|XXXX|1001|1101 OPEN LOOP USING PROFILE 13
0X9E  0000|XXXX|1001|1110 OPEN LOOP USING PROFILE 14
0X9F  0000|XXXX|1001|1111 OPEN LOOP USING PROFILE 15
```

The data for the command Output register represents the requested drive in millivolts.

Example

Suppose you would like to axis 0 to go into open loop at 2000mV of drive using profile 10 and axis 1 to go into open loop at 4000mV of drive using profile 11. You would send the commands in the following format:

	SAR CMND INDX	(HEX)	
4TTTTT+0	0000|XXXX|1001|1010	(0X9A)	Issue an Open Loop using Profile 10 command
+1	2000	(07D0)	Axis 1 Requested Drive
4TTTTT+2	0000|XXXX|1001|1011	(0X9B)	Issue an Open Loop using Profile 11 command
+3	4000	(0FA0)	Axis 2 Requested Drive

Get Profile Commands

Format: 0000 0000 1010 NNNN

N Used for Command Index described below

These commands allow the programmer to retrieve the motion controller's stored motion profiles. Only one value in one profile can be read per axis each time the synchronization register is changed

The format of the Command Register for the Get Profile commands is given below:

NOTE: The Status Area Request area (bits 8-11) of the command and the entire Command Value Output register are ignored.

```
          |    | 111|1111
BIT # 1234|5678|9012|3456
      -------------------
HEX       |    |CMND|INDX
VALUE -------------------
0XA0  0000|XXXX|1010|0000 GET PROFILE 0 or 4 MODE
0XA1  0000|XXXX|1010|0001 GET PROFILE 0 or 4 ACCEL
0XA2  0000|XXXX|1010|0010 GET PROFILE 0 or 4 DECEL
0XA3  0000|XXXX|1010|0011 GET PROFILE 0 or 4 SPEED
0XA4  0000|XXXX|1010|0100 GET PROFILE 1 or 5 MODE
0XA5  0000|XXXX|1010|0101 GET PROFILE 1 or 5 ACCEL
0XA6  0000|XXXX|1010|0110 GET PROFILE 1 or 5 DECEL
0XA7  0000|XXXX|1010|0111 GET PROFILE 1 or 5 SPEED
0XA8  0000|XXXX|1010|1000 GET PROFILE 2 or 6 MODE
0XA9  0000|XXXX|1010|1001 GET PROFILE 2 or 6 ACCEL
0XAA  0000|XXXX|1010|1010 GET PROFILE 2 or 6 DECEL
0XAB  0000|XXXX|1010|1011 GET PROFILE 2 or 6 SPEED
0XAC  0000|XXXX|1010|1100 GET PROFILE 3 or 7 MODE
0XAD  0000|XXXX|1010|1101 GET PROFILE 3 or 7 ACCEL
0XAE  0000|XXXX|1010|1110 GET PROFILE 3 or 7 DECEL
0XAF  0000|XXXX|1010|1111 GET PROFILE 3 or 7 SPEED
```

Notice that for each command, two different profile numbers are listed. The actual profile read is determined by the axis issuing the command. The following chart show which commands affect which profiles when issued on a particular axes (Notice that one the first eight (8) profiles can be modified from the Programmable Controller):

COMMAND	AXIS 1	AXIS 2
0xA0-0xA3	0	4
0xA4-0xA7	1	5
0xA8-0xAB	2	6

0xAC-0xAF	3	7

The second Input register for the axis will hold the requested data on the scan following these commands being issued.

Example:

We wish to get profiles 2 and 7 into the motion controller. Looking at the chart above, we can see that we issue Get Profile commands to axis 0 to get profile 2 and we use axis 1 Get Profile commands to get profile 7. Therefore, we can do both at the same time. Suppose that the motion controller had the following values for profiles 2 and 7:

	PROFILE 2	PROFILE 7
MODE	00001	00001
ACCEL	100	150
DECEL	70	70
SPEED	12000	20000

This would take at least five scans because we can get one word of each profile on each axis per scan:

First scan:
```
4TTTTT+0 00A8h (Get profile 2 MODE)
4TTTTT+1 XXXXh (Unused)
4TTTTT+2 00ACh (Get profile 7 MODE)
4TTTTT+3 XXXXh (Value of profile 7 MODE)
```

Second scan:

After the Synchronization Output register is incremented, the motion controller will process the commands and update the Synchronization Input register to match. At this time, the input registers would hold the following values:
```
3TTTTT+0 XXXXh (Axis 1 STATUS)
3TTTTT+1 0001h (Profile 2 MODE)
3TTTTT+2 XXXXh (Axis 2 STATUS)
3TTTTT+3 0001h (Profile 7 MODE)
```

We can now send the second set of requests:
```
4TTTTT+0 00A9h (Get profile 2 ACCEL)
4TTTTT+1 XXXXh (Unused)
4TTTTT+2 00ADh (Get profile 7 ACCEL)
4TTTTT+3 XXXXh (Unused)
```

Third scan:

After the Synchronization Output register is incremented, the motion controller will process the commands and update the Synchronization Input register to match.

At this time, the input registers would hold the following values:

```
3TTTTT+0 XXXXh (Axis 1 STATUS)
3TTTTT+1   100 (Profile 2 ACCEL)
3TTTTT+2 XXXXh (Axis 2 STATUS)
3TTTTT+3   150 (Profile 7 ACCEL)
```

We can now send the third set of requests:

```
4TTTTT+0 00AAh (Get profile 2 DECEL)
4TTTTT+1 XXXXh (Unused)
4TTTTT+2 00AEh (Get profile 7 DECEL)
4TTTTT+3 XXXXh (Unused)
```

Fourth scan:

After the Synchronization Output register is incremented, the motion controller will process the commands and update the Synchronization Input register to match. At this time, the input registers would hold the following values:

```
3TTTTT+0 XXXXh (Axis 1 STATUS)
3TTTTT+1    70 (Profile 2 DECEL)
3TTTTT+2 XXXXh (Axis 2 STATUS)
3TTTTT+3    70 (Profile 7 DECEL)
```

We can now send the fourth set of requests:

```
4TTTTT+0 00ABh (Get profile 2 SPEED)
4TTTTT+1 XXXXh (Unused)
4TTTTT+2 00AFh (Get profile 7 SPEED)
4TTTTT+3 XXXXh (Unused)
```

Fifth scan:

After the Synchronization Output register is incremented, the motion controller will process the commands and update the Synchronization Input register to match. At this time, the input registers would hold the following values:

```
3TTTTT+0 XXXXh (Axis 1 STATUS)
3TTTTT+1 12000 (Profile 2 SPEED)
3TTTTT+2 XXXXh (Axis 2 STATUS)
3TTTTT+3 20000 (Profile 7 SPEED)
```

Get Parameter Commands

Format: 0000 0000 1011 NNNN

N Used for Command Index described below

These commands allow the Programmable Controller to read the current state of the motion controller initialization parameters. The requested parameter will be returned in the corresponding axis status area.

The format of the Command Register for the Get Parameter commands is given below:

NOTE: The Status Area Request area (bits 5-8) of the command and the entire Command Value Output register are ignored.

```
          |    | 111|1111
BIT # 1234|5678|9012|3456
      -------------------
HEX       |    |CMND|INDX
VALUE -------------------
0XB0  0000|XXXX|1011|0000 Get CONFIGURATION
0XB1  0000|XXXX|1011|0001 Get SCALE
0XB2  0000|XXXX|1011|0010 Get OFFSET
0XB3  0000|XXXX|1011|0011 Get EXTEND LIMIT
0XB4  0000|XXXX|1011|0100 Get RETRACT LIMIT
0XB5  0000|XXXX|1011|0101 Get PROPORTIONAL GAIN
0XB6  0000|XXXX|1011|0110 Get INTEGRAL GAIN
0XB7  0000|XXXX|1011|0111 Get DIFFERENTIAL GAIN
0XB8  0000|XXXX|1011|1000 Get EXTEND FEED FORWARD
0XB9  0000|XXXX|1011|1001 Get RETRACT FEED FORWARD
0XBA  0000|XXXX|1011|1010 Get EXTEND ACCEL FEED FORWARD
0XBB  0000|XXXX|1011|1011 Get RETRACT ACCEL FEED FORWARD
0XBC  0000|XXXX|1011|1100 Get DEAD BAND ELIMINATOR
0XBD  0000|XXXX|1011|1101 Get IN POSITION
0XBE  0000|XXXX|1011|1110 Get FOLLOWING ERROR
0XBF  0000|XXXX|1011|1111 Get AUTO STOP
```

The second Input register for the axis will hold the requested parameter after the motion controller matches the Synchronization Input register with the Synchronization Output register.

Example

Suppose you would like to get the SCALE from both axes. You would send commands with the following format:

```
                      CMND INDX (HEX)
4TTTTT+1 0000|0000|1011|0001 (00B1)   Requests SCALE for axis 1
      +2 XXXX|XXXX|XXXX|XXXX (XXXX)   Ignored
4TTTTT+3 0000|0000|1011|0001 (00B1)   Requests SCALE for axis 2
      +4 XXXX|XXXX|XXXX|XXXX (XXXX)   Ignored
```

After the Synchronization Output register is incremented, the RMC will process the commands and update the Synchronization Input register to match. At this time, the input registers would hold the following values:

```
3TTTTT+1 XXXX|XXXX|XXXX|XXXX (XXXX)  STATUS of axis 1
      +2 XXXX|XXXX|XXXX|XXXX (XXXX)  SCALE of axis 1
3TTTTT+3 XXXX|XXXX|XXXX|XXXX (XXXX)  STATUS of axis 2
      +4 XXXX|XXXX|XXXX|XXXX (XXXX)  SCALE of axis 2
```

Event Step Edit Commands

Format: 0000 RRRR 1110 NNNN

R	Used for Status Area Request
N	Used for Command Index described below

These commands let you change an event parameter's value across a range of steps in a single scan. The format of the Command Register for the Event Step Edit commands is given below:

```
          |    | 111|1111
BIT # 1234|5678|9012|3456
      -------------------
HEX       |SAR |CMND|INDX
VALUE -------------------
0XE0  0000|XXXX|1110|0000 Start Step Number (0 to 255)
0XE1  0000|XXXX|1110|0001 End Step Number (0 to 255; > E0)
0XE2  0000|XXXX|1110|0010 Parameter to Modify (0 to 7)
0XE3  0000|XXXX|1110|0011 Value to be Used
```

For command types `0XE0` and `0XE1`, the data for the command Output register represents a step number. For command type `0XE2`, the data Output register represents the field that will be modified by `0XE3`. Use the following table to select a field:

0XE2 DATA VALUE	**STEP FIELD**
0	MODE
1	ACCEL
2	DECEL
3	SPEED
4	COMMAND VALUE
5	COMMAND
6	LINK TYPE/NEXT
7	LINK VALUE

To edit the Event Step table, each of these commands must be issued once to initialize the parameters; this will take four scans. Once the beginning step number, the ending step number, and the parameter to be changed have been defined, all the steps can be changed in a single scan by sending `0XE3` command with the new parameter value. All four of these commands must be sent on the same axis.

Example:

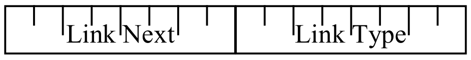
If you have a step table that is 100 steps long and you want to change the time delay link value in each step, it would take 100 scans by using the Event Step Transfer commands. If the scan time is 20ms, this would take 2 seconds. By using the Event Step Edit commands, it would take 80ms for the first change (four scans times 20ms/scan) and only 20ms for another change.

Assuming that the steps we want to modify are step 0 to 99 and the new link value will be 500, the following commands would be sent:

```
First scan:  4TTTTT+1 0XE0h (Set Start Step Number)
             4TTTTT+2     0 (Start Step Number of 0)

Second scan: 4TTTTT+1 0XE1h (Set End Step Number)
             4TTTTT+2    99 (End Step Number of 99)

Third scan:  4TTTTT+1 0XE2h (Set Parameter to Modify)
             4TTTTT+2     7 (Select LINK VALUE)

Fourth scan: 4TTTTT+1 0XE3h (Set Actual Value)
             4TTTTT+2   500 (LINK VALUE itself)
```

LINK TYPE/NEXT

In order to fit a single Event Step into eight words, the Link Type and Link Next fields share a single word. In the most significant byte, the Link Next field is stored; in the least significant byte, the Link Type is stored:



Diagnostics Command

Format: 0000 XXXX 1111 XXXX

X Don't care.

These commands are for testing the motion controller under safe conditions. This command echoes the Output registers for the axis into the Input registers. This tests the motion controller to Programmable Controller communications.

Example:

If you sent out the following data in the Output registers:

```
                   CMND      (HEX)
4TTTTT+1 0000|0011|1111|0101 (03F5)   Diagnostic command
      +2 0001|0010|0100|1000 (1248)   Diagnostic data
4TTTTT+3 0000|1100|1111|1010 (0CFA)   Diagnostic command
      +4 1000|0100|0010|0001 (8421)   Diagnostic data
```

After the Synchronization Output register is incremented, the motion controller will process the commands and update the Synchronization Input register to match. At this time, the input registers would hold the following values:

```
3TTTTT+1 0000|0011|1111|0101 (03F5)   Same as Output
      +2 0001|0010|0100|1000 (1248)   Same as Output
3TTTTT+3 0000|1100|1111|1010 (0CFA)   Same as Output
      +4 1000|0100|0010|0001 (8421)   Same as Output
```

Event Step Transfer Command

Format: 1110 DSSS SSSS SNNN

D	Direction. Write (0) or read (1) event step.
S	Step number. 0 to 255.
N	Field in the event step. 0 to 7.

These commands are used to move event step information to and from the motion controller. The **D** bit shown in the binary expansion of the Event Step Transfer Command register indicates whether the command will write an Event Step value (indicated by a 0) or read an Event Step value (indicated by a 1). Therefore, commands E000 to E7FF set (write) data to the motion controller, while commands E800 to EFFF get (read) data from the motion controller. The eight **S** bits specify the step number, and the three **N** bits represent the field in the Event Step to access. Use the following chart to select the desired field:

N BITS VALUE	**STEP FIELD**
000	MODE
001	ACCEL
010	DECEL
011	SPEED
100	COMMAND VALUE
101	COMMAND
110	LINK TYPE/NEXT
111	LINK VALUE

To write event step information to the motion controller, use the following command and data Output registers pairs:

Command	**Data**
Register	**Register**
E000	Step 0 MODE
E001	Step 0 ACCELERATION
E002	Step 0 DECELERATION
E003	Step 0 SPEED
E004	Step 0 COMMAND VALUE
E005	Step 0 COMMAND
E006	Step 0 LINK TYPE/NEXT
E007	Step 0 LINK VALUE
E008	Step 1 MODE
E009	Step 1 ACCELERATION
.	.
.	.
E7FD	Step 255 COMMAND
E7FE	Step 255 LINK TYPE/NEXT
E7FF	Step 255 LINK VALUE

To update multiple words every Programmable Controller scan, you can also use the Command and Data Output registers for multiple axes. For example, by using one axis to transfer the first half of the table and the second axis to transfer the second half of the table, transfer time can be cut in half.

To read event step information to the motion controller, use the following command Output registers, increment the Synchronization Output registers, and wait for the Synchronization Input register to match it. The second Input register on the axis will return the following values:

Command	**Data In**
Register	**Register**
E800	Step 0 MODE
E801	Step 0 ACCELERATION
E802	Step 0 DECELERATION
E803	Step 0 SPEED
E804	Step 0 COMMAND VALUE
E805	Step 0 COMMAND

```
E806    Step 0 LINK TYPE/NEXT
E807    Step 0 LINK VALUE

E808    Step 1 MODE
E809    Step 1 ACCELERATION
 .            .
 .            .
EFFD    Step 255 COMMAND
EFFE    Step 255 LINK TYPE/NEXT
EFFF    Step 255 LINK VALUE
```

Appendix B: Command Field Reference

MODE

Default: 00000

Nine bits in the MODE word determine the way the motion controller responds to control commands and parameters. Bit 1 is the MSB and bit 16 is the LSB.

Click here for the Mode Bit Map

Bit 1 - Graph Disable

When this bit is set for an axis, the controller will not log graph data during the move. This is useful for troubleshooting long sequences of moves.

Bit 9 - S-Curve

When this bit is set the module controller calculates an s-shaped target, resulting in smoother motion with more gradual starts and stops and higher peak speeds. In an s-curve move, the maximum acceleration is 1.53 times the straight-line acceleration (used by a trapezoidal motion profile). If another move command is given before the s-curve is finished, there may be a discontinuity (jerk) at the transition.

Bit 11 - Quick Mode

When this bit is set, a move will ramp the drive up in Open Loop mode to the value specified (in millivolts) in the SPEED field and maintain it there until it reaches the deceleration point. It will then ramp down to the requested position (specified by the COMMAND VALUE) in Closed Loop mode. The ACCELERATION and DECELERATION fields specify the ramp based on MODE bits 15 and 16.

For this mode to work properly the EXTEND FEED FORWARD and RETRACT FEED FORWARD parameters should be adjusted correctly. They are use to calculate the drive given to the axis at the moment ramp down begins.

If the speed achieved by the axis during the open loop portion of the move is greater than 65,535, when the ramp down starts, the target speed will be limited to 65,535.

Bit 12 - Sync

When this bit is set, it causes the axis to move synchronized to the other axes that have this bit set. The axes are synchronized as follows:

1. The axis with the longest move is the master, which uses its full profile.
2. For each other synchronized axis, the length of its move determines its profile. All synched axes will accelerate together, move at constant speed together (their constant speeds will differ, depending on their move lengths), and decelerate together, arriving simultaneously at their individual COMMAND POSITIONS.
3. If any axis is halted by an error condition, the other axes will also be halted.

For more details on synchronizing axes, see Synchronizing Axes.

Bits 13-14 - Integrator Mode Select

These two bits define four integrator modes:

Bit #	13	14
Mode 0 - Integrator always active	0	0
Mode 1 - Integrator active only during deceleration and in position	0	1
Mode 2 - Integrator active only during in position	1	0
Mode 3 - Integrator never active	1	1

Bits 15-16 - Acceleration and Deceleration Mode Select

These two bits define four acceleration/deceleration modes:

Bit #	15	16
Mode 0 - Ramp Rate (Default)	0	0
Mode 1 - Ramp Rate * 1000	0	1
Mode 2 - Distance to specified speed	1	0
Mode 3 - Time to specified speed	1	1

The actual meanings of the Acceleration/Deceleration modes depend on whether the move is in open or closed loop. For open loop mode, see the Open Loop command. The closed loop Acceleration/Deceleration modes are described below:

In Mode 0, the ACCELERATION and DECELERATION parameters define the ramp rate in position units per second per second.

In Mode 1, the ACCELERATION and DECELERATION parameters define the ramp rate in 1000 position units per second per second. For example, with position units of 0.001 inches, an ACCELERATION of 50 will cause the axis to accelerate at 50 inches per second per second.

In Mode 2, the parameters define the distance in position units to accelerate or decelerate to the specified SPEED.

In Mode 3, the parameters define the time in milliseconds to accelerate or decelerate to the specified SPEED.

Refer to Open Loop Command for details on the use of these modes when operating in open loop.

MODE Word Bit Map

The axis MODE word contains 16 bits of information. This hexadecimal table provides an easy way to convert hexadecimal numbers to bit patterns.

```
                      F  1 1 1 1   1 1 1 1   1 1 1 1   1 1 1 1
                      E  1 1 1 0   1 1 1 0   1 1 1 0   1 1 1 0
 Hexadecimal To       D  1 1 0 1   1 1 0 1   1 1 0 1   1 1 0 1
 Binary Conversion    C  1 1 0 0   1 1 0 0   1 1 0 0   1 1 0 0
 Table                B  1 0 1 1   1 0 1 1   1 0 1 1   1 0 1 1
                      A  1 0 1 0   1 0 1 0   1 0 1 0   1 0 1 0
                      9  1 0 0 1   1 0 0 1   1 0 0 1   1 0 0 1
                      8  1 0 0 0   1 0 0 0   1 0 0 0   1 0 0 0
                      7  0 1 1 1   0 1 1 1   0 1 1 1   0 1 1 1
                      6  0 1 1 0   0 1 1 0   0 1 1 0   0 1 1 0
                      5  0 1 0 1   0 1 0 1   0 1 0 1   0 1 0 1
                      4  0 1 0 0   0 1 0 0   0 1 0 0   0 1 0 0
                      3  0 0 1 1   0 0 1 1   0 0 1 1   0 0 1 1
                      2  0 0 1 0   0 0 1 0   0 0 1 0   0 0 1 0
                      1  0 0 0 1   0 0 0 1   0 0 0 1   0 0 0 1
                      0  0 0 0 0   0 0 0 0   0 0 0 0   0 0 0 0
                         _______   _______   _______   _______
                        | | | | | | | | | | | | | | | | | | | |
                        | | | | | | | | | | | |1|1|1| |1|1|1|1|
                        |1|2|3|4| |5|6|7|8| |9|0|1|2| |3|4|5|6|
 Bit Definition         |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|
----------------         / / / /   / / / /   / / / /   / / / /
Graph Disable ----- 01 -/ / / /   / / / /   / / / /   / / / /
Reserved ---------- 02 --/ / /   / / / /   / / / /   / / / /
Reserved ---------- 03 ---/ /   / / / /   / / / /   / / / /
Reserved ---------- 04 ----/   / / / /   / / / /   / / / /
                              / / / /   / / / /   / / / /
Reserved ---------- 05 ------/ / / /   / / / /   / / / /
Reserved ---------- 06 -------/ / /   / / / /   / / / /
Reserved ---------- 07 --------/ /   / / / /   / / / /
Reserved ---------- 08 ---------/   / / / /   / / / /
                                   / / / /   / / / /
S-Curve ----------- 09 -----------/ / / /   / / / /
Reserved ---------- 10 ------------/ / /   / / / /
Quick Mode -------- 11 -------------/ /   / / / /
Sync -------------- 12 --------------/   / / / /
                                        / / / /
Integrator Mode --- 13 ----------------/ / / /
Integrator Mode --- 14 -----------------/ / /
```

```
Accel/Decel Mode -- 15 ------------------/ /
Accel/Decel Mode -- 16 -------------------/
```

ACCELERATION

Default: 1000

Range: 0 to 65535

This parameter defines the acceleration ramp rate used by the axis for a move. It has four meanings depending on the **Acceleration and Deceleration Mode** bits in the MODE word.

In Mode 0, this parameter is interpreted as acceleration rate and is expressed in Position Units/sec/sec. If SCALE is set so one Position Unit equaled 0.001", then a value of 200 would represent an acceleration rate of 0.200 inches/sec/sec. In Mode 0, the relationship between the ramp distance, ACCELERATION and SPEED is:

$$\text{Ramp Distance} = \frac{\text{SPEED x SPEED}}{\text{ACCELERATION x 2}}$$

$$\text{ACCELERATION} = \frac{\text{SPEED x SPEED}}{\text{Ramp Distance x 2}}$$

In Mode 1, this field is an acceleration rate and is expressed in 1000 position units/sec/sec. If the SCALE is set so one position unit equal 0.001", then a value of 200 would represent an acceleration rate of 200 inches/second/second. In Mode 1, the relationship between the ramp distance, acceleration and speed is:

$$\text{Ramp Distance} = \frac{\text{SPEED x SPEED}}{\text{ACCELERATION x 2000}}$$

$$\text{ACCELERATION} = \frac{\text{SPEED x SPEED}}{\text{Ramp Distance x 2000}}$$

In Mode 1, if one Position Unit equals 0.001", an ACCELERATION of 200 would represent an acceleration rate of 200 inches/sec/sec.

In Mode 2, this parameter defines the **distance** in position units to the specified SPEED.

In Mode 3, it defines the **time** in milliseconds to the specified SPEED.

DECELERATION

Default: 1000

Range: 0 to 65535

This parameter is the same as ACCELERATION except it specifies the deceleration ramp length or deceleration rate.

SPEED

Default: 1000

Range: 0 to 65535

The SPEED parameter sets the constant speed to be achieved after acceleration. SPEED is expressed in position units/second. If the SCALE is set so one position unit equals 0.001", a speed of 25 inches per second is expressed as 25000.

NOTE: When using Acceleration/Deceleration Mode 2, changing the SPEED without changing the ACCELERATION and DECELERATION distances will change the acceleration and deceleration rates.

NOTE: If the SPEED is set to zero, the axis will not move.

COMMAND VALUE

Range: Depends on Command

The COMMAND VALUE field is multipurpose. Usually it is used with the 'G' command to specify the position where the axis is to move. When used this way, the value is bounds-checked by the Motion Control Module using the EXTEND LIMIT and RETRACT LIMIT and is then used as the COMMAND POSITION value.

Here is the complete list of commands that use this field:

For this command:	COMMAND VALUE is:	COMMAND VALUE Range:
'A' (Change ACCEL)	Acceleration value	0 to 65535
'D' (Change DECEL)	Deceleration value	0 to 65535
'E' (Start Events)	Event number	0 to 255
'G' (Go to Position)	Requested Position	Valid 16-bit Position
'I' (Set Integral Drive)	Integral Drive value	-10000 to 10000
'J' (Go Relative)	Relative Position	-32768 to 32767
'M' (Set Mode)	Mode	0 to 0FFFFh
'N' (Set Null Drive)	New Null value	-2000 to 2000
'O' (Open Loop Output)	Requested Millivolts of Drive	-12000 to 12000
'V' (Set Speed)	Speed	0 to 65535

'v' (Set Speed-Signed)	Speed	-32768 to 32767

Make sure that the data in the COMMAND VALUE field is correct before you issue any of these commands.

Command Field

This field represents the command to be sent to the motion controller. You may enter a command by either typing the letter or symbol of the command in a command field, or by entering the ASCII number in either decimal or hexadecimal in the field. The chart below lists all available commands with their ASCII values:

Description	Command	Decimal	Hex
Change Acceleration	A	65	041
Change Deceleration	D	68	044
Start Events	E	69	045
Set Feed Forward	F	70	046
Go	G or g	71 or 103	047 or 067
Halt	H or h	72 or 104	048 or 068
Set Integral Drive	I	73	049
Set Integral Drive to Null Drive	i	105	069
Relative Move	J or j	74 or 106	04A or 06A
Disable Drive Output	K	75	04B
Set Mode	M	77	04D
Set Null Drive	N	78	04E
Set Null Drive to Integral Drive	n	110	06E
Open Loop	O	79	04F
Set Parameters	P or p	80 or 112	050 or 070
Quit Events	Q	81	051
Restore Null Drive	R	82	052
Restore Integral Drive	r	114	072
Save Null Drive	S	83	053
Save Integral Drive	s	115	073
Update Flash	U	85	055
Set Speed (Unsigned)	V	86	056
Set Speed (Signed)	v	118	076
Start a Graph	y	121	079

Appendix C: Parameter Field Reference

CONFIGURATION Word

Default: 0000

Eleven bits of this 16-bit word control the configuration of the module. Bit 1 is the MSB; bit 16 is the LSB.

Click here for the CONFIG Word Bit Map

Bits 1 - 4 - Transducer Type bits

Set these bits all to 0 if the transducer is start-stop.

If the transducer is gated, set the bits so the value of these four bits equals the number of recirculations the gated transducer is set to (bit 1 = 8, bit 2 = 4, bit 3 = 2, and bit 4 = 1). For example, if the transducer is set to one recirculation, bit 4 should be set. If the transducer is set to 10 recirculations bits 1 and 3 should be set, and if it is set for 15 recirculations, all four bits should be set to 1.

When a transducer is set for multiple recirculations, there is a small time delay between those recirculations. During this time the counter on the motion controller continues to count, resulting in "extra" counts which are added to the TRANSDUCER COUNTS. These extra counts use some of the 65535 maximum counts that are available, reducing the maximum stroke the module can control to 65535 minus the "extra" counts. To correct for this effect the module calculates the "extra" counts based on the number of transducer recirculations and subtracts them from the transducer count. If the value of these bits and the transducer recirculations agree, the transducer count will be slightly positive when the transducer is at its minimum position.

If you are using a gated transducer and want resolution better than 0.001 inches, we suggest using 10 recirculations and setting bits 1 and 3. This will give 0.0001-inch resolution with a stroke of about 6.5 inches.

Bit 9 – Falling edge Transducer bit

When this bit is set the Transducer interface hardware is configured to respond to the falling edge of the pulses on a start/stop type transducer. If any of the transducer type bits (1 through 4) are set, this bit is ignored.

Bit 10 - Reverse Drive Mode bit

When this bit is set the polarity of the axis's output voltage is reversed. This bit is useful when you are connected to two drives which do not have differential or isolated inputs, and you need one to extend with positive drive and the other to extend with negative drive. (If the drives have differential or isolated inputs, you can just reverse the connections to the drive that must extend with negative drive.) When this bit is set, transducer counts will DECREASE with positive drive.

Warning: This bit must be set properly when the Set Parameters command is issued.

Bit 11 - Absolute Mode bit

This mode is intended for use with a two-valve system, one controlling the flow rate and the other the direction. This axis controls the flow-rate valve. The directional valve must be controlled by other means. In this mode, the axis generates a positive drive output regardless of the direction of the move. The drive will not go negative if the motion controller overshoots the target. This is useful for some injection and blow-molding applications.

When the axis is stopped, you must go into open loop mode.

Warning: This bit must be set properly when the Set Parameters command is issued.

Bit 12 - Continue Mode bit

This bit affects what happens when the module loses contact with the PLC. When this bit is set, the module will finish any move it has started, otherwise it will halt immediately upon detecting loss of contact with the PLC. This is useful for finishing a move that must complete to prevent machine downtime (for example, a partial shot in an injection molding machine).

Bit 13 - Simulate bit

When this bit is set, the drive output is set to zero and the magnetostrictive transducer inputs are ignored. Internally the TARGET POSITION is used as the ACTUAL POSITION. This mode is used for debugging. (The transducer error bits and LEDS will be cleared.)

NOTE: Drive output is always 0 volts while in simulate mode; no commands, including open loop, will change this.

Bit 14 - 15 - Prescale Divisor Bits

Prescaling may be desired on systems with long strokes. The prescale bits are used along with the SCALE parameter to convert TRANSDUCER COUNTS to ACTUAL POSITION. The effect is to divide the SCALE by 1, 2, 4, or 8. The divide-by factor is specified as follows:

Counts ÷	Bit 14	Bit 15
1	0	0
2	0	1
4	1	0
8	1	1

The advantage of these bits is that they let you maintain a higher resolution in the SCALE parameter. For example, a SCALE of 31027 and a divide-by of 4 gives an effective scale of 7756.75. Without using the divide-by bits, the SCALE would have to be rounded up to 7757, which has less resolution—with a 64" rod, the rounded value would make a 0.002" error at 64". Note: with the stroke is always limited to 220 inches due to the 18-bit transducer counter resolution.

Bit 16 - Integrator Limit

When this bit is cleared, the integrator limit is 20% of full drive. When the bit is set, the limit is 80%. If the integrator tries to go above the limit, the Integrator Windup bit in the word is set and the integrator value is held at the limit. The integrator limit is designed to prevent the drive output from saturating.

NOTE: The actual drive output may be reduced based on the values of the Extend Feed Forward and Retract Feed Forward.

CONFIGURATION Word Bit Map

The axis CONFIGURATION word contains 16 bits of information. This hexadecimal table provides an easy way to convert hexadecimal numbers to bit patterns.

```
                      F  1 1 1 1   1 1 1 1   1 1 1 1   1 1 1 1
                      E  1 1 1 0   1 1 1 0   1 1 1 0   1 1 1 0
 Hexadecimal To       D  1 1 0 1   1 1 0 1   1 1 0 1   1 1 0 1
 Binary Conversion    C  1 1 0 0   1 1 0 0   1 1 0 0   1 1 0 0
 Table                B  1 0 1 1   1 0 1 1   1 0 1 1   1 0 1 1
                      A  1 0 1 0   1 0 1 0   1 0 1 0   1 0 1 0
                      9  1 0 0 1   1 0 0 1   1 0 0 1   1 0 0 1
                      8  1 0 0 0   1 0 0 0   1 0 0 0   1 0 0 0
                      7  0 1 1 1   0 1 1 1   0 1 1 1   0 1 1 1
                      6  0 1 1 0   0 1 1 0   0 1 1 0   0 1 1 0
                      5  0 1 0 1   0 1 0 1   0 1 0 1   0 1 0 1
                      4  0 1 0 0   0 1 0 0   0 1 0 0   0 1 0 0
                      3  0 0 1 1   0 0 1 1   0 0 1 1   0 0 1 1
                      2  0 0 1 0   0 0 1 0   0 0 1 0   0 0 1 0
                      1  0 0 0 1   0 0 0 1   0 0 0 1   0 0 0 1
                      0  0 0 0 0   0 0 0 0   0 0 0 0   0 0 0 0
                         _______   _______   _______   _______
                        | | | | | | | | | | | | | | | | | | | |
                        | | | | | | | | | | | |1|1|1| |1|1|1|1|
                        |1|2|3|4| |5|6|7|8| |9|0|1|2| |3|4|5|6|
 Bit Definition         |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|
----------------         / / / /   / / / /   / / / /   / / / /
MDT Type/8 Recirc – 01 -/ / / /   / / / /   / / / /   / / / /
MDT Type/4 Recirc – 02 --/ / /   / / / /   / / / /   / / / /
MDT Type/2 Recirc – 03 ---/ /   / / / /   / / / /   / / / /
MDT Type/1 Recirc – 04 ----/   / / / /   / / / /   / / / /
                              / / / /   / / / /   / / / /
Reserved ---------- 05 ------/ / / /   / / / /   / / / /
Reserved ---------- 06 -------/ / /   / / / /   / / / /
Reserved ---------- 07 --------/ /   / / / /   / / / /
Reserved ---------- 08 ---------/   / / / /   / / / /
                                   / / / /   / / / /
Reserved ---------- 09 -----------/ / / /   / / / /
Reverse Drive ----- 10 ------------/ / /   / / / /
Absolute Mode ----- 11 -------------/ /   / / / /
Continue Mode ----- 12 --------------/   / / / /
                                        / / / /
Simulate Mode ----- 13 ----------------/ / / /
Prescale Bits ----- 14 -----------------/ / /
Prescale Bits ----- 15 ------------------/ /
Integrator Limit -- 16 -------------------/
```

SCALE

Default: 30300

Range: -32767 to 32767, excluding 0

TIP: From the **Tools** menu, select **Scale/Offset Calibration** to help calculate SCALE and OFFSET.

The SCALE parameter is used in calculating the ACTUAL POSITION in Position Units from the TRANSDUCER COUNTS, using the following formula (Position Units can be 0.001", 0.1 mm, etc.):

$$\text{ACTUAL POSITION} = \frac{\text{TRANSDUCER COUNTS x SCALE}}{\text{32768 x Prescale Divisor}} + \text{OFFSET}$$

In linear position applications, SCALE accounts for the following:

1. Transducer Compensation: The SCALE parameter compensates for differences in magnetostrictive transducers. Each transducer will indicate a different physical distance for a given interrogation cycle. Usually, only a small change from the default value is necessary to compensate the transducer.
2. Distance Translation: A second and more useful feature of SCALE is its ability to translate a fixed physical distance to usable position units (position units from thousandths to hundredths or vice versa). The translation is done using a combination of SCALE and recirculation selection.

SCALE is defined as 32768 times the number of Position Units per TRANSDUCER COUNT times the Prescale Divisor, as shown in the following equation (P0 and P1 are two positions, C0 and C1 are the corresponding TRANSDUCER COUNTS, and the Prescale Divisor is specified in the CONFIGURATION WORD):

$$\text{SCALE} = \frac{(P0 - P1)}{(C0 - C1)} \text{ x 32768 x Prescale Divisor}$$

The SCALE can be calculated on a magnetostrictive displacement transducer using its calibration number (nominally 9.012 microseconds per inch) with the following formula:

$$\text{SCALE} = \frac{\text{Position Units per Inch x 32768 x Prescale Divisor x Sign}}{\text{Cal. Number } (\mu\text{s/in}) \text{ x 120MHz x \# of Recirculations}}$$

In the above formula, the 120MHz comes from the motion module's internal counter, and Sign equals +1 when an extend move yields increasing ACTUAL POSITION counts, and -1 when an extend move yields decreasing ACTUAL POSITION counts.

The calibration number must be specified in the same units as the Position Units (inches, mm, etc.). Position Units are generally 0.001 inches or 0.1 mm. Usually there are 1000 Position Units per inch when the calibration number is in microseconds per inch.

Click here for MDT SCALE Calculation examples

NOTE: When the SCALE parameter is changed, the EXTEND LIMIT, RETRACT LIMIT, and OFFSET must also be changed.

NOTE: You should never set the SCALE below 16383. If it does go below that value, then use the Prescale Divisor bits in the CONFIGURATION WORD. As you can see from the ACTUAL POSITION calculation above, this will affect the scaling, but will reduce round-off errors.

SCALE Calculation Examples

Example 1

For a system using a magnetostrictive transducer with a calibration number of 9.1392 µs per inch, a position unit of 0.001 inch, and 1 recirculation, the SCALE will be:

$$\text{scale} = \frac{1000\text{(Position Units Per Inch)} \times 32768}{9.1392(\mu\text{s/in}) \times 120\text{MHz} \times 1\text{ (Recirculations)}} = 29878.62$$

SCALE = 29879

Example 2

For a system using a magnetostrictive transducer with a calibration number of 9.0110 µs per inch, a position unit of 0.1 millimeter, and 1 recirculation, the SCALE will be:

$$\text{scale} = \frac{254\text{(Position Units Per Inch)} \times 32768}{9.0110(\mu\text{s/in}) \times 120\text{MHz} \times 1\text{ (Recirculations)}} = 7697.14$$

Since the scale is less than 16383.5, use the Prescale Divisor bits in the CONFIGURATION WORD to divide the transducer counts by 4. Then multiply the scale by 4 to obtain:

SCALE = 7697.14 x 4 = 30789

Example 3

For a system using a magnetostrictive transducer with a calibration number of 8.9772 µs per inch, a position unit of 0.001 inch, and 8 recirculations, the SCALE will be:

$$\text{scale} = \frac{1000\text{(Position Units Per Inch)} \times 32768}{8.9772(\mu\text{s/in}) \times 120\text{MHz} \times 8\text{ (Recirculations)}} = 3802.22$$

Since the scale is less than 16383.5, use the Prescale Divisor bits in the CONFIGURATION WORD to divide the transducer counts by 8. Then multiply the scale by 8 to obtain:

SCALE = 3802.22 x 8 = 30418

OFFSET

Default: 0

Range: -65536 to 65535

TIP: From the **Tools** menu, select **Scale/Offset Calibration** to help calculate SCALE and OFFSET.

OFFSET is used to shift the ACTUAL POSITION with respect to the transducer zero. The OFFSET is specified in Position Units and is equal to the desired ACTUAL POSITION at zero TRANSDUCER COUNTS. It is defined as:

$$\text{OFFSET} = P0 - \frac{(P0 - P1) \times C0}{(C0 - C1)}$$

CAUTION: When using OFFSET, you must be familiar with the limitations of 16-bit math. These are described in **Valid 16-bit Positions**.

Why Bother?

The OFFSET can be used to set your 0 point anywhere along the axis.

What if it is displayed under the monitor program incorrectly?

The OFFSET may be displayed incorrectly on the monitor main screen under some circumstances. This is because the offset specifies the zero location of the axis. The numbers are not necessarily incorrect (the module still functions correctly), but they may not look right. To correct the problem, enter the correct value for the OFFSET and save the parameters to a file (using the **Save As…** item from the **File** menu). Whenever you start the monitor program, read the parameters from the monitor and the problem should be corrected. Each time the monitor program is started, your last-opened board file will be opened, automatically correcting the problem.

EXTEND LIMIT

Default: Current position on power-up

Range: Valid 16-bit Position

The EXTEND LIMIT specifies the maximum requested position value that the motion controller will allow as a COMMAND VALUE. (When the SCALE is negative, this is the minimum value.) A COMMAND VALUE that exceeds this value will be set to the EXTEND LIMIT, and the PARAMETER ERROR bit in the STATUS word will be set. The EXTEND LIMIT is given in Position Units.

NOTE: The EXTEND LIMIT must be changed when the SCALE or OFFSET parameters are changed. Extending is the direction that gives increasing TRANSDUCER COUNTS.

NOTE: On startup, the EXTEND LIMIT defaults to the current position of the axis, so new EXTEND and RETRACT LIMITS must be issued followed by a 'P' command before the axis will move, unless different limits have been saved in the FLASH.

RETRACT LIMIT

Default: Current position on power-up

Range: Valid 16-bit Position

The RETRACT LIMIT specifies the minimum value the motion controller will allow as a position COMMAND VALUE. (When the SCALE is negative, this is the maximum value.) A COMMAND VALUE below this value will be set to the RETRACT LIMIT. The RETRACT LIMIT is given in Position Units.

NOTE: The RETRACT LIMIT must be changed when the SCALE or OFFSET parameters are changed.

NOTE: On startup, the RETRACT LIMIT defaults to the current position of the axis, so new EXTEND and RETRACT LIMITS must be issued followed by a 'P' command before the axis will move, unless different limits have been saved in the FLASH.

PROPORTIONAL GAIN

Default: 1

Range: 0 to 65535

The PROPORTIONAL GAIN specifies the amount of drive added to the output for each unit of position error. It is defined as:

PROPORTIONAL GAIN = 0.1 millivolts per unit of Position Error

Proportional drive is defined as:

Proportional Drive = PROPORTIONAL GAIN x Position Error / 10

NOTE: The actual drive output may be reduced based on the values of the Extend Feed Forward and Retract Feed Forward .

CAUTION: Increase the PROPORTIONAL GAIN gradually. Excessive gain can cause oscillation that could cause damage or injury.

Think about this:

Internally, the motion controller must compare the error between the TARGET and ACTUAL POSITIONS with error limits to keep values from overflowing. The error limit is the error at which full drive (10 volts) will occur. This internal error limit is calculated as follows:

ERROR LIMIT = 100,000 / PROPORTIONAL GAIN

Therefore, if the PROPORTIONAL GAIN is set to 100, the ERROR LIMIT will be 1,000, which means any error greater than 1,000 will be treated as an error of 1,000 and the OVERDRIVE error bit will be set in the STATUS Word.

INTEGRAL GAIN

Default: 1

Range: 0 to 65535

The INTEGRAL GAIN is used to control the amount of drive provided by the integrator. The integrator adds the position error to an accumulator every millisecond. The INTEGRAL GAIN should be adjusted after the feed forwards have been set to optimal values. Using the integrator before the feed forwards have been set properly will cause the system to overshoot the target. We recommend that you set the INTEGRAL GAIN to a value of at least 50.

INTEGRAL GAIN is defined as:

INTEGRAL GAIN = 0.1 mV per 1024 counts of accumulated Position Error

Integral Drive is defined as:

Integral Drive = INTEGRAL GAIN x Accumulated Counts / 10240

NOTE: The actual drive output may be reduced based on the values of the Extend Feed Forward and Retract Feed Forward .

Why Bother?

INTEGRAL GAIN should be used to compensate for the fact that loads may vary, valves are non-linear and the axis may have trouble getting to the COMMAND POSITION without INTEGRAL GAIN.

DIFFERENTIAL GAIN

Default: 0
Range: 0 to 65535

The DIFFERENTIAL GAIN field is used to minimize the error between the TARGET POSITION and the ACTUAL POSITION. The change in error is multiplied by the DIFFERENTIAL GAIN value to get the differentiator drive term. DIFFERENTIAL GAIN is defined as:

DIFFERENTIAL GAIN = 0.1 millivolts per change in Position Error

Differential Drive is defined as:

Differential Drive = (E0 - E1) x DIFFERENTIAL GAIN / 10

where E0 is the position error (TARGET POSITION – ACTUAL POSITION) in the first time period and E1 is the position error in the second time period. (E0 - E1) is the change in position error between the two time periods.

NOTE: The actual drive output may be reduced based on the values of the Extend Feed Forward and Retract Feed Forward .

CAUTION: To avoid oscillation during initial tuning start with values below 10.

Why Bother?

The differentiator field should usually be set to a value of 0. This is because noise in the MDT signals causes the speed calculations to be noisy.

EXTEND FEED FORWARD

Default: 100
Range: 0 to 65535

TIP: After the axis has made a complete move without oscillations or overdrive errors, use the 'F' command to automatically set the FEED FORWARD value.

FEED FORWARD is an open loop compensation that is proportional to the TARGET SPEED of the axis. This value is expressed in terms of millivolts per 1,000 Position Units per second. EXTEND FEED FORWARD drive is added to the output only when the axis is extending. The drive output provided by the EXTEND FEED FORWARD is determined as follows:

$$\text{Feed Forward Drive} = \frac{\text{EXTEND FEED FORWARD x TARGET SPEED}}{1{,}000}$$

You can find the appropriate value for EXTEND FEED FORWARD by making a move with the axis using a SPEED of 1,000. The amount of output drive required to maintain this speed should be used as the EXTEND FEED FORWARD parameter. If the axis lags after the parameter has been set, the feed forward is too small or the system response is too slow. If the axis leads, the feed forward is too large or the system response is too slow. The 'F' command used after an extend move will automatically adjust the EXTEND FEED FORWARD parameter. The EXTEND and RETRACT FEED FORWARDS are also known as velocity feed forwards.

NOTE: Feed Forward Drive is limited to a maximum value of 10,000. Inserting that value in the equation above and solving for the Feed Forward term gives the follow relationship:

Feed Forward <= (10,000 * 1,000) / TARGET SPEED

That is, the larger the SPEED the smaller the maximum Feed Forward value.

For example, at 30 in/sec SPEED, the maximum Feed Forward is:

Max Feed Forward = (10,000 * 1,000)/30,000 = 10,000/30 = 333

If you set SPEED to 30,000 and enter a Feed Forward value larger than 333, the value will be reduced to 333.

Think about this:

If the feed forward value is less than 300, an axis going at a speed of 65535 will not get a full valve drive from the feed forward term alone. If the feed forward value is less than 200, the axis drive system is too big and the system may have trouble controlling position. If this happens, then reduce the gain on the amplifier or use a smaller valve or reduce pressure until the feed forward terms are over 300.

On hydraulic systems, the extend and retract feed forward terms will be different by the ratio of the extend and retract surface areas of the position.

The Proportional Gain, Integral Gain and Differential Gain as well as the Integral Drive Limit are adjusted internally based on the ratio of the Extend and Retract Feed Forwards. The direction with the lower Feed Forward value will have effectively lower Gains and Integral Limit. This is done to compensate for the different system dynamics in the two directions.

For example, for a system with an Extend Feed forward of 100 and a Retract Feed Forward of 200 the effective gains in the extend direction will be one half (100 / 200) as big as the specified gains.

RETRACT FEED FORWARD

Default: 100

Range: 0 to 65535

Same as EXTEND FEED FORWARD, except it is used when retracting.

NOTE: Retracting is the direction that returns decreasing TRANSDUCER COUNTS.

EXTEND ACCELERATION FEED FORWARD

Default: 0
Range: 0 to 65535

The Extend Acceleration Feed Forward causes the controller to give extra drive while accelerating and a breaking drive while decelerating (negative acceleration) in the extend direction. The drive due to Acceleration Feed Forward is calculated as follows:

Accel FF Drive = EXT ACCEL FF x 1 mV / 100,000 position units/sec/sec

The Acceleration Feed Forward provides a second order approximation (prediction) of how much drive is required to move.

If the position unit is 0.001 inches, this equals millivolts of drive per 100 inches per second per second.

Why Bother?

The acceleration feed forward help the axis track better while accelerating or decelerating only. The acceleration feed forwards should not be adjusted until the velocity feed forwards are adjusted correctly.

RETRACT ACCELERATION FEED FORWARD

Default: 0
Range: 0 to 65535

Same as EXTEND ACCELERATION FEED FORWARD, except it is used when retracting.

DEAD BAND ELIMINATOR

Default: 0
Range: 0 to 2000

Some valves and drives do not react to small changes in output around the null drive value; this effect is termed "dead band". The DEAD BAND ELIMINATOR value is the number of millivolts added to or subtracted from the drive output (depending on the direction of travel) so the drive output is outside the dead band.

If a value outside the valid range is entered, the parameter error bit will be set, and the value will be set to zero.

CAUTION: Do not make this value too large or the drive will oscillate.

Why Bother?

On an axis with overlapped spools or a large amount of static friction, the axis may oscillate slowly around the COMMAND POSITION. This happens when the axis does not respond to small changes in drive output. The integrator winds up until the axis moves and then has too much drive, overshooting the COMMAND POSITION. The axis then winds the integrator down until the axis overshoots the other way. If this happens, look at the NULL DRIVE. The NULL DRIVE will oscillate slowly as the axis oscillates. Notice the peak-to-peak values of the NULL DRIVE oscillation and subtract the minimum value from the maximum value and then divide the result by two. Enter this value in the DEAD BAND ELIMINATOR field as a starting point.

A spool that has 10% overlap will require a value of about 1,000 counts.

IN POSITION

Default: 50

Range: 0 to 65535

IN POSITION specifies the size of a window around the COMMAND POSITION. When the ACTUAL POSITION gets within this window, the In Position bit is set (but not latched) in the word.

Example:

If an axis COMMAND POSITION is 10,000 and the IN POSITION parameter is 30, the In Position bit will be set when the ACTUAL POSITION is between 9,971 and 10,029. The bit will be cleared whenever the ACTUAL POSITION is outside the range.

FOLLOWING ERROR

Default: 250

Range: 0 to 65535

The FOLLOWING ERROR determines how large the difference between the TARGET POSITION and ACTUAL POSITION can get before the Following Error bit is set in the word.

AUTO STOP

Default: 0x1FE0 (Soft Stops enabled, Hard Stops enabled for Transducer Errors)

Click here for the AUTO STOP Bit Map

The Auto Stop parameter controls the action taken on the rising edge of any of the eight (8) most-significant bits in the Status word, which are all fault bits:

Fault	Status Bit
7	15 - No Transducer
6	14 - Transducer Noise
5	13 - Transducer Overflow
4	12 - Overdrive
3	11 - Parameter Error
2	10 - Pos./Press. Overflow
1	9 - Integrator Windup
0	8 - Following Error

The available actions for each fault are listed below:

- **Status Only**
  The fault will be reflected in its corresponding status bit in the Status word, but no further action will be taken. This option is not available for the following faults: No Transducer, Transducer Overflow, and Transducer Noise.
- **Soft Stop**
  The fault will trigger ramping the axis to a stop. If the axis is in closed loop, the speed will ramp down to zero using the current Deceleration value. If the axis is in open loop, then the drive will be ramped down to zero at a rate of 100 mV per millisecond. If there is an event sequence on the axis, it will stop executing. The Halt status bit will also be set in the Status word. Some faults will use the open loop ramp down even if the axis was in closed loop: Position Overflow, No Transducer, Transducer Overflow, and Transducer Noise. This is done because we cannot count on the positions being correct, and therefore cannot maintain closed loop control.
- **Hard Stop**
  The fault will trigger the drive output going immediately to 0 mV. The axis will be placed in open loop mode, and will stay there until a new command is issued. If there is an event sequence on the axis, it will stop executing. The Halt and Open Loop status bit will also be set in the Status word.

To view or change the Auto Stop parameter value, open the Auto Stop popup editor using one of the methods described in Using Popup Editors. Select the desired action for each fault type and click OK. Changes to this parameter do not take effect until you issue a Set Parameters (P) command.

Manual Parameter Entry

This parameter can also be edited manually, but this is discouraged since it is much easier to use the popup editor. You can enter hexadecimal numbers by typing a leading **0x**, as in **0x1FE0**. Each of the eight faults listed above has two bits assigned to it, label S and H in the table below:

Bit	Description	Bit	Description
15	Fault 7 - Bit S	7	Fault 7 - Bit H
14	Fault 6 - Bit S	6	Fault 6 - Bit H
13	Fault 5 - Bit S	5	Fault 5 - Bit H
12	Fault 4 - Bit S	4	Fault 4 - Bit H
11	Fault 3 - Bit S	3	Fault 3 - Bit H
10	Fault 2 - Bit S	2	Fault 2 - Bit H
9	Fault 1 - Bit S	1	Fault 1 - Bit H
8	Fault 0 - Bit S	0	Fault 0 - Bit H

For each fault, the two bits in the Auto Stop parameter define the action as follows:

Bit H	Bit S	Description
0	0	Status Only
0	1	Soft Stop
1	0	Hard Stop
1	1	Hard Stop

If you select Status Only for a fault that cannot use that action, then the axis will use a Soft Stop action for that fault.

AUTO STOP Bit Map

This table provides an easy method to convert bit patterns to hexadecimal numbers.

```
                     F  1 1 1 1   1 1 1 1   1 1 1 1   1 1 1 1
                     E  1 1 1 0   1 1 1 0   1 1 1 0   1 1 1 0
  Hexadecimal To     D  1 1 0 1   1 1 0 1   1 1 0 1   1 1 0 1
  Binary Conversion  C  1 1 0 0   1 1 0 0   1 1 0 0   1 1 0 0
  Table              B  1 0 1 1   1 0 1 1   1 0 1 1   1 0 1 1
                     A  1 0 1 0   1 0 1 0   1 0 1 0   1 0 1 0
                     9  1 0 0 1   1 0 0 1   1 0 0 1   1 0 0 1
                     8  1 0 0 0   1 0 0 0   1 0 0 0   1 0 0 0
                     7  0 1 1 1   0 1 1 1   0 1 1 1   0 1 1 1
                     6  0 1 1 0   0 1 1 0   0 1 1 0   0 1 1 0
                     5  0 1 0 1   0 1 0 1   0 1 0 1   0 1 0 1
                     4  0 1 0 0   0 1 0 0   0 1 0 0   0 1 0 0
                     3  0 0 1 1   0 0 1 1   0 0 1 1   0 0 1 1
                     2  0 0 1 0   0 0 1 0   0 0 1 0   0 0 1 0
                     1  0 0 0 1   0 0 0 1   0 0 0 1   0 0 0 1
                     0  0 0 0 0   0 0 0 0   0 0 0 0   0 0 0 0
                        _______   _______   _______   _______
                       | | | | | | | | | | | | | | | | | | | |
                       | | | | | | | | | | | |1|1|1| |1|1|1|1|
                       |1|2|3|4| |5|6|7|8| |9|0|1|2| |3|4|5|6|
  Bit Definition       |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|
 ----------------       / / / /   / / / /   / / / /   / / / /
```

```
S  No MDT ------------ 01 -/ / / /   / / / /   / / / /   / / / /
O  MDT Noise --------- 02 --/ / /   / / / /   / / / /   / / / /
F  MDT Overflow ------ 03 ---/ /   / / / /   / / / /   / / / /
T  Overdrive  -------- 04 ----/   / / / /   / / / /   / / / /
                                 / / / /   / / / /   / / / /
S  Parameter Error --- 05 ------/ / / /   / / / /   / / / /
T  Position Overflow – 06 -------/ / /   / / / /   / / / /
O  Integrator Windup – 07 --------/ /   / / / /   / / / /
P  Following Error --- 08 ---------/   / / / /   / / / /
========================              / / / /   / / / /
H  No MDT ------------ 09 -----------/ / / /   / / / /
A  MDT Noise --------- 10 ------------/ / /   / / / /
R  MDT Overflow ------ 11 -------------/ /   / / / /
D  Overdrive --------- 12 --------------/   / / / /
                                           / / / /
S  Parameter Error --- 13 ----------------/ / / /
T  Position Overflow – 14 -----------------/ / /
O  Integrator Windup – 15 ------------------/ /
P  Following Error --- 16 -------------------/
```

If both Soft Stop and Hard Stop bits are set for a particular error condition, a Hard Stop will be executed.

Appendix D: Status Field Reference

Valid 16-Bit Positions

The positions used by the motion controller are stored in a 16-bit number. This limits the range of positions to 65536 positions. However, by using the SCALE and OFFSET parameters, the user can shift where this 65536-position range is located.

If SCALE is positive, then OFFSET holds the lowest position value. If SCALE is negative, then OFFSET holds the highest position value. However, in no case can a position be less than -65536 or greater than 65535 position units.

Example 1:

SCALE: 30300

OFFSET: 0

Position Range: 0 to 65535

Discussion: Because SCALE is positive, the positions range from OFFSET to OFFSET + 65535.

Example 2:

SCALE: 25604

OFFSET: -1000

Position Range: -1000 to 64535

Discussion: Because SCALE is positive, the positions range from OFFSET to

OFFSET + 65535.

Example 3:

SCALE: 32050

OFFSET: 1000

Position Range: 1000 to 65535

Discussion: Because SCALE is positive, the positions range from OFFSET to OFFSET + 65535. However, because the upper limit would be greater than 65535 (66535), it is truncated at 65535.

Example 4:

SCALE: -30300

OFFSET: 40000

Position Range: -25535 to 40000

Discussion: Because SCALE is negative, the positions range from OFFSET - 65535 to OFFSET.

Example 5:

SCALE: -31541

OFFSET: -10000

Position Range: -65536 to -10000

Discussion: Because SCALE is negative, the positions range from OFFSET - 65535 to OFFSET. However, because the lower limit would be less than -65536 (-75536), it is truncated at -65536.

COMMAND POSITION

The COMMAND POSITION is the requested position (specified by the COMMAND VALUE) with bounds checking applied. If the requested position is outside the RETRACT or EXTEND LIMIT, the COMMAND POSITION will be set to the value of the limit, and the axis will go only to the limit. The COMMAND POSITION is updated when any move command, or 'P' command is issued using the COMMAND parameter.

Why Bother?

If the COMMAND POSITION is not the same as the requested position then one of two things has happened:

1. A program error has asked the axis to go to an invalid position. In this case the program error should be corrected. The PARAMETER ERROR bit in will be ON when this occurs.
2. The COMMAND VALUE field has just been changed and the Motion Controller has not had a chance to acknowledge the new request. Note that the IN POSITION bit is no longer valid for the rest of the scan. The IN POSITION bit is valid only when the COMMAND VALUE is equal to the COMMAND POSITION.

TARGET POSITION

When the axis is in Closed Loop Mode, the TARGET POSITION is the calculated instantaneous ideal position for the axis. The TARGET POSITION is calculated every loop by the target generator state machine of the on-board motion control program. The PID routine uses the difference between the TARGET POSITION and the ACTUAL POSITION as its input to compute any required corrective drive.

During a move the transition of the TARGET POSITION toward the COMMAND POSITION will be the perfect profile for the ACTUAL POSITION to follow.

NOTE: When an axis is stopped, the TARGET POSITION should be the same as the COMMAND POSITION unless an error or HALT has occurred (see STATUS).

NOTE: When an axis is not in position Closed Loop Mode, TARGET POSITION is set to the ACTUAL POSITION.

Why Bother?

Knowing the relationship between the TARGET POSITION and ACTUAL POSITION is key to tuning the axis. The main goal in tuning the axis is to minimize the error between the TARGET and ACTUAL POSITIONS. The plot function is a very nice visual aid in tuning the axis.

ACTUAL POSITION

The ACTUAL POSITION is the measured position of the axis at any moment. This position is updated every loop. The ACTUAL POSITION is calculated from the TRANSDUCER COUNTS as follows:

$$\text{ACTUAL POSITION} = \frac{\text{TRANSDUCER COUNTS x SCALE}}{32768} + \text{OFFSET}$$

Why Bother?

Before applying drive or hydraulic power, make sure the target position is set to the actual position using a ' P' command. Otherwise axis will jump to the TARGET POSITION.

See TARGET POSITION.

TRANSDUCER COUNTS

TRANSDUCER COUNTS is the axis position read directly from the transducer counters with no scaling. Internally, TRANSDUCER COUNTS are stored in an 18-bit counter, but only the lower 16 bits are available through the read-back register. The display on the PC shows the value of the 18 bit counter.

Why Bother?

Knowing the COUNTS is necessary for calculating the SCALE and OFFSET parameters. It is also necessary to make sure that a positive drive causes the counts to increase.

Magnetostrictive Displacement Transducers (MDT)

MDTs are transducers that provide absolute position, in a manner similar to a radar and doesn't require a homing routine. The counts will be in the range of 2,500 to 65,000 when the MDT is working. The COUNTS field gets set to 0 when the MDT is not working. The motion controller ignores a new reading if the new count is 250 counts or more from the last count.

AXIS STATUS Word

The axis STATUS word contains 16 bits of information about the condition of the axis. You can use any of the first eight error bits to trigger a STOP on the axis using the AUTO STOP parameter.

Click here for the Axis Status Bit Map.

Error bits 2 through 8 are cleared whenever a 'G' command is given. Error bit 1 is cleared after a 'P' command if a transducer is detected at that time.

NOTE: Bit 1 is the most significant bit (MSB; left-most bit), Bit 16 is the least significant bit (LSB, right-most bit).

Tip: To display the individual fields of the STATUS word, select Status Bits from the Window menu (or press CTRL+B) to display the Bit Status window.

Bit 1 - No Transducer

This bit is set and the corresponding front panel axis LED (light emitting diode) glows red if the transducer does not respond at least once every 6 milliseconds. It causes a Hard Stop or a Soft Stop, depending on the setting of AUTO STOP bit 9. This bit will stay on until a new command is given to the axis.

Bit 2 - Transducer Noise

If a new transducer count differs from the previous reading by more than 100 counts for six consecutive milliseconds, the new value is assumed to be an error and this error bit is set. When the Transducer Noise bit is set, the system response depends on the setting of the AUTO STOP word. It causes a Hard Stop or a Soft Stop, depending on the setting of AUTO STOP bit 10. This bit will stay on until a new command is given to the axis.

Bit 3 - Transducer Overflow

This bit is set when there has been no stop pulse from the transducer by the time the counter has overflowed 18 bits (about 2 ms). When this bit is set, the system response depends on the setting of the AUTO STOP word. It causes a Hard Stop or a Soft Stop, depending on the setting of AUTO STOP bit 11. This bit will stay on until a new command is given to the axis.

Bit 4 - Overdrive

This bit is set when the calculated drive output exceeds the 12-bit range of the D/A converter. Usually this error indicates the system does not have enough power to drive the axis at the requested SPEED. The module will truncate the drive to 12 bits (+10V or –10V). It causes no action, a Soft Stop, or a Hard Stop, depending on the setting of AUTO STOP bits 4 and 12. This bit will stay on until a new command is given to the axis.

Bit 5 - Parameter Error

This bit is set when an initialization parameter or control parameter is out of bounds. In some cases one parameter's limit will depend on the value of another parameter, so definite limits may not always be available. However, the motion controller does try to replace the erroneous value with another that is within range, so the offending parameter can be determined by comparing the parameter values before and after the error bit is set. This error causes no action, a Soft Stop, or a Hard Stop, depending on the setting of AUTO STOP bits 5 and 13. This bit will stay on until a new command is given to the axis.

Because this error bit is used for many types of errors, the monitor program has the ability to display the last several parameter errors in more detail. This list of errors is compiled only while the monitor program is running, so it is recommended that the monitor program be left running while configuring the system. The last parameter error type for each axis is stored on the module and read-up when the module is connected to the monitor program.

To display this list, select **Parameter Error List** from the **Window** menu. For control firmware versions dated 19971016 or later, this list should include brief descriptions of the last twenty parameter errors to occur on the motion controller. To receive help on a particular error, in the error list, click on the error, and then click **Help on Error**.

Bit 6 - Position Overflow

This bit is set when the actual position has exceeded the range that can be displayed with a 16-bit number. See Valid 16-bit Positions for details on this limitation. This bit is also set if—when recirculations are used—TRANSDUCER COUNTS is calculated to less than zero. ACTUAL POSITION will be displayed as 65535 in the overflow case, and as zero (0) in the underflow case. It causes either an Open Loop ramp down (Soft Stop), or a Hard Stop, depending on the setting of AUTO STOP bits 2 and 10. This bit will stay on until a new command is given to the axis.

To recover from this error condition you must move the axis, either manually or with an Open Loop Command, until the ACTUAL POSITION becomes valid again.

This error occurs for one of two reasons:

1. The overflow case (Actual Position displayed as 65535) happens because the transducer is longer than can be accommodated by the 16 bit Actual Position Field given the scaling of the transducer counts. To fix this, you can either reduce the SCALE parameter; increase the Prescale Divisor in the CONFIGURATION Word; or, if you are using a PWM type transducer, increase the number of recirculations specified in the CONFIGURATION Word (without increasing the number of recirculations in the transducer).
2. The Underflow case (Actual Position displayed as 0) happens because the number of recirculations specified in the CONFIGURATION Word is too large. To fix this, decrease the number of recirculations specified in the Configuration Word (without increasing the number of recirculations in the transducer).

Bit 7 - Integrator Windup

This bit is set when the integrator value is larger than 20% or 80%, depending on the setting of the Integrator bit in the Configuration word. It causes no action, a Soft Stop, or a Hard Stop, depending on the setting of AUTO STOP bits 7 and 15. This bit will stay on until a new command is given to the axis.

Bit 8 - Following Error

This bit is set when the difference between the TARGET POSITION and the ACTUAL POSITION is greater than the MAXIMUM ERROR parameter. It causes no action, a Soft Stop, or a Hard Stop, depending on the setting of AUTO STOP bits 8 and 16. This bit will stay on until a new command is given to the axis.

Bit 9 - Acknowledge

This bit will toggle after the motion controller receives a valid command or Status Area Request. This can be used to verify that the motion controller has received the command.

Bit 10 - Initialized

This bit is set after a Set Parameter ('P') command is successfully executed. Until this bit is set, the axis will not respond to any GO ('G') commands. This bit is cleared whenever the module is reset.

Bit 11-12 - State Bits

Bits 11 and 12 show the state of the target generator:

STATE	Bit 11 (State Bit B)	Bit 12 (State Bit A)
Stopped	0	0
Accelerating	0	1
Constant Speed	1	0
Decelerating	1	1

These bits will also be active when the axis is in open loop (zero output drive, ramping drive up, constant drive, and ramping drive down).

For example, to detect when an axis's TARGET POSITION has come to a halt after an 'H' command, monitor for the following status bit combination:

Halt bit	ON (1)	(Bit 14)
State bit A	OFF (0)	(Bit 12)
State bit B	OFF (0)	(Bit 11)

You may also want to monitor the Stopped bit (bit 15).

As another example, to detect when an internal error has caused a Hard Stop or a transducer error has caused a ramped stop, monitor for the following status bits:

Halt bit	ON (1)	(Bit 14)
Open Loop bit	ON (1)	(Bit 13)

Bit 13 - Open Loop

This bit is set when the axis is in Open Loop. The axis can be in Open Loop because of an 'O' command or because an error caused a Hard Stop.

NOTE: While an axis is in Open Loop mode it may drift due to a valve 'out-of-null' condition.

Bit 14 - Halt

There are three conditions that will set this bit:

- A Halt ('H') command is issued
- An internal error causes a Soft Stop
- An internal error causes a Hard Stop

This bit is cleared when a new command is issued.

NOTE: When an axis is halted (Halt bit ON, State A bit OFF and State B bit OFF) then the integral drive is automatically set to Null Drive and the Integrator and Null Drive stop updating. This is done so a Halt command can be given to the axes when the hydraulic system is turned off or the valves are disconnected so the module no longer can control the position. If the Integrator and Null Drive were allowed to update they could become quite large and cause the axis position to jump when the hydraulic system was turned back on.

Bit 15 - Stopped

This bit is set when the average speed of the axis is less than 500 TRANSDUCER COUNTS per second and is cleared when the speed is greater than 1000 TRANSDUCER COUNTS per second. It can be used as an axis obstruction indication. The Stopped bit is not latched.

Bit 16 - In Position

This bit will be set when the difference between the ACTUAL POSITION and COMMAND POSITION is less than the value in the IN POSITION field; it is not latched.

NOTE: The control program can monitor this bit to determine when an axis has arrived at the commanded position.

STATUS Word Bit Map

The axis Status word contains 16 bits of information about the status of the axis. This hexadecimal table provides an easy way to convert hexadecimal numbers to bit patterns.

```
                        F  1 1 1 1   1 1 1 1   1 1 1 1   1 1 1 1
                        E  1 1 1 0   1 1 1 0   1 1 1 0   1 1 1 0
   Hexadecimal To       D  1 1 0 1   1 1 0 1   1 1 0 1   1 1 0 1
   Binary Conversion    C  1 1 0 0   1 1 0 0   1 1 0 0   1 1 0 0
   Table                B  1 0 1 1   1 0 1 1   1 0 1 1   1 0 1 1
                        A  1 0 1 0   1 0 1 0   1 0 1 0   1 0 1 0
                        9  1 0 0 1   1 0 0 1   1 0 0 1   1 0 0 1
                        8  1 0 0 0   1 0 0 0   1 0 0 0   1 0 0 0
                        7  0 1 1 1   0 1 1 1   0 1 1 1   0 1 1 1
                        6  0 1 1 0   0 1 1 0   0 1 1 0   0 1 1 0
                        5  0 1 0 1   0 1 0 1   0 1 0 1   0 1 0 1
                        4  0 1 0 0   0 1 0 0   0 1 0 0   0 1 0 0
                        3  0 0 1 1   0 0 1 1   0 0 1 1   0 0 1 1
                        2  0 0 1 0   0 0 1 0   0 0 1 0   0 0 1 0
                        1  0 0 0 1   0 0 0 1   0 0 0 1   0 0 0 1
                        0  0 0 0 0   0 0 0 0   0 0 0 0   0 0 0 0
                           _______   _______   _______   _______
                          | | | | | | | | | | | | | | | | | | | |
                          | | | | | | | | | | | |1|1|1| |1|1|1|1|
                          |1|2|3|4| |5|6|7|8| |9|0|1|2| |3|4|5|6|
   Bit Definition         |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|
  ----------------         / / / /   / / / /   / / / /   / / / /
**No MDT ------------ 01 -/ / / /   / / / /   / / / /   / / / /
**MDT Noise --------- 02 --/ / /   / / / /   / / / /   / / / /
**MDT Overflow ------ 03 ---/ /   / / / /   / / / /   / / / /
 *Overdrive --------- 04 ----/   / / / /   / / / /   / / / /
                                / / / /   / / / /   / / / /
 *Parameter Error --- 05 ------/ / / /   / / / /   / / / /
 *Position Overflow – 06 -------/ / /   / / / /   / / / /
 *Integrator Windup – 07 --------/ /   / / / /   / / / /
 *Following Error --- 08 ---------/   / / / /   / / / /
                                     / / / /   / / / /
  Acknowledge ------- 09 -----------/ / / /   / / / /
  Initialized ------- 10 ------------/ / /   / / / /
  State B ----------- 11 -------------/ /   / / / /
  State A ----------- 12 --------------/   / / / /
                                          / / / /
  Open Loop --------- 13 ----------------/ / / /
  Halt -------------- 14 -----------------/ / /
  Stopped ----------- 15 ------------------/ /
  In Position ------- 16 -------------------/
```

* Can cause a Soft or Hard Stop if the corresponding bits are set in the AUTO STOP field.

** Will cause either a Soft or Hard Stop depending how AUTO STOP bits 9, 10, and 11 are set.

DRIVE

DRIVE is the output to the actuator in millivolts. The 12-bit (4000 step) digital value output, -10000 (full negative drive) to 10000 (full positive drive) will generate a ±10 volt output in steps of 0.005 volts. The internal drive calculations are done to 14-bit resolution. This additional resolution is used to "dither" the least significant bit of the output, giving additional resolution.

NOTE: Software adjusts the null. There are no hardware nulling adjustments on the motion control module. Null drive will be near 0 volts.

Why Bother?

If the DRIVE tries to go below –10,000 or above 10,000, an overdrive condition has occurred; reduce SPEED to correct this.

ACTUAL SPEED

The ACTUAL SPEED is the calculated speed at which the axis is moving at any point in time. It is based on changes in TRANSDUCER COUNTS, displayed in position units per second, and updated every loop. The Actual Speed is only calculated for last four controller scans so the value it calculates is greatly affected by the resolution of the transducer.

NULL DRIVE

This field displays the amount of drive in millivolts that it takes to hold the ACTUAL POSITION at the TARGET POSITION while at rest. Ideally this number should be zero if the valve is perfectly nulled. This field is automatically updated toward the Integral Drive when the ACTUAL POSITION is exactly equal to the COMMAND POSITION and the speed is zero.

This field is only used in open loop mode. NULL DRIVE gets added to the drive output.

Why Bother?

The NULL DRIVE field can be used to null valves and drives. Usually there is a null or offset adjustment on the valve or amplifier that can be adjusted so the NULL DRIVE is 0.

STEP

This field shows the current step the axis is executing of the 256 steps in the Event Step Table. This is valid only when Event Control is being used. Use this field in conjunction with the Link Value status word to monitor the progress of an axis through the step table.

LINK VALUE

This field is used only when the axis is using the Event Step Table. It represents the current event step link value of the current step for the axis. The current step is displayed in the STEP status word.

For delay link types (types 'D' and 'd'), this field displays the number of delay units left (in either counter ticks or milliseconds).

Appendix E: Event Step Link Reference

Link Types and Link Values

Link Type and **Link Value** specify the condition that causes the motion controller to execute the next step in a sequence. These fields may be edited manually; however in most cases it is easier to use the Popup Editor. Start the popup editor by pressing **Enter** in, or double-clicking on, either the **Link Type** or **Link Value** field.

The **Link Type and Link Value** dialog box divides the link types into three categories:

•System-wide (Basic, non-axis dependent)

These link types allow the axis to wait on a global condition such as a time delay. These link types are listed below; select a link type for further details:

Link Type	Description
End of Sequence	This link type stops the event sequence.
Time Delay	Wait for a number of milliseconds.

- **Current Axis (Basic)**

  These link types allow the axis to wait for a condition based on its own state. These link types are listed below; select a link type for further details:

Link Type	Description
Absolute Limit Switch	Wait for the position to cross a fixed position.
Relative Limit Switch	Wait for the position to cross a position specified as a distance from either the start or end of the current move.
Speed	Wait for the speed to be above or below a value.
Status Bits	Wait for one or more status bits to be on or off.

- **Selected Axis (Enhanced)**

  These link types allow the axis to wait for conditions on another axis to occur before proceeding. These link types are listed below; select a link type for further details:

Link Type	Description
Position	Wait for the position on an axis to be above or below a fixed position.
Speed	Wait for the speed on an axis to be above or below a value.
Status Bits	Wait for one or more status bits on an axis to be on or off.

Link Next

The Link Next field contains the number of the next step that will execute when the condition of the Link Type and Value are met.

The Link Next field can point to any step (from 0 to 255). When a chain of steps is finished the last step should have a Link Next of 0 to indicate so. If a never-ending loop is required, the last step will have a Link Next value of a step earlier in the chain.

Tip: When editing the steps, you can press CTRL+G to cause the cursor to jump to the step indicated by the Link Next field. This allows the user to view the steps in a step sequence.

System-wide Link Types

Link Type - End of sequence

Link Type: 0 (decimal 0)
Link Value: Reserved
Range: must be 0

When this Link Type is encountered the step sequence stops: no additional steps are executed unless they are started as a new sequence. We recommend using Step 0 as the last Step of all chains so they all end in a known spot. If Step 0 is used this way, its link type must be zero.

Link Type - Time Delay

Link Type: D
Link Value: Number of milliseconds to delay
Range: 0 to 65,535 milliseconds

A timer is started for this link type. When the timer reaches the value specified by the Link Value, the next event step in the sequence is executed. Simply enter the number of milliseconds you wish to delay in the Link Value.

Current Axis Link Types

Link Type - Current Axis Absolute Limit Switch

Link Type: L (Target Position) or l (Target Position)
Link Value: Limit Position
Range: Any valid position in position units

These link types are used to detect when the current axis crosses a position. This link type can be used to change speeds on-the-fly or trigger events on other axes. The Link Value holds the Limit position in position units. Either the Target Position or Actual Position can be used to compare with the Limit position.

NOTE: The link is made once the position crosses the limit, rather than being made whenever the position is on one side or the other of the limit. If you wish to wait for the axis to be on one side or the other, use the Any Axis Position link type.

Example:

Suppose a move is being made from 4000 to 12000 position units. If the link type and value are 'L' and 8000 respectively, the next step would be executed when the Target Position reached 8000.

Using with the *Link Type and Link Value* Popup Dialog

1. Under **Link Type Category**, select **Current Axis (Basic)**.
2. Under **Link Type**, select **Absolute Limit Switch**.
3. Under **Link Condition**, select whether you wish to use Target or Actual Position for the comparison.
4. Under **Link Condition**, enter the Limit position in the **Threshold** box.

Using without the *Link Type and Link Value* Popup Dialog

1. Select the Link Type:
   Use 'L' to use the Target Position in the comparison, 'l' to use the Actual Position.
2. Enter the Limit position in the Link Value.

Link Type - Current Axis Relative Limit Switch

Link Type: R (Target Position; Start of Move)
r (Actual Position; Start of Move)
N (Target Position; End of Move)
n (Actual Position; End of Move)
Link Value: Limit Position Offset from Start or End of Move
Range: -32,768 to 32,767 position units

These link types dynamically calculate a software limit switch. The actual link type used determines whether the limit switch is relative from the start of the move (the position the axis was at when the last move command was issued) or the end of the move (the Command Position status field). When the current axis crosses this limit switch, the next event step in the sequence is executed. The Link Value field holds the Limit position offset; that is, the number of position units that are added to the reference (either Start or End of the move) to give the Limit position.

Example:

To move to the next event step when the current move has moved 1000 position units from the start of the move, a Link position offset of 1000 would be used. Therefore, if the move was from 4000 to 12000, the limit would be triggered at 5000, and if the move was from 2000 to 4000, the limit would be triggered at 3000.

Using with the *Link Type and Link Value* Popup Dialog

1. Under **Link Type Category**, select **Current Axis (Basic)**.

2. Under **Link Type**, select **Relative Limit Switch**.
3. Under **Link Condition**, select whether you wish to use Target or Actual Position for the comparison.
4. Under **Link Condition**, enter the Limit position offset in the **Threshold** box.

Using without the *Link Type and Link Value* Popup Dialog

1. Select the Link Type.
   Use the following table to choose a Link Type:

Link Type	Position Used	Relative To
R	Target Position	Start of Move
r	Actual Position	Start of Move
N	Target Position	End of Move
n	Actual Position	End of Move

2. Enter the Limit position offset in the Link Value.

Link Type - Current Axis Speed

Link Type: S (Target Speed) or s (Actual Speed)

Link Value: Limit Speed

Range: 0 to 65,535 position units per second

These link types are used to detect when the speed of the current axis has reached a user-specified value. This link type can be used to change speeds on-the-fly or trigger events on another axis. The Link Value holds the Limit speed in position units per second. Either the Target Speed or Actual Speed may be used to compare with the Limit speed.

Using with the *Link Type and Link Value* Popup Dialog

1. Under **Link Type Category**, select **Current Axis (Basic)**.
2. Under **Link Type**, select **Speed**.
3. Under **Link Condition**, select whether you wish to use Target or Actual Speed for the comparison.
4. Under **Link Condition**, enter the Limit speed in the **Threshold** box.

Using without the *Link Type and Link Value* Popup Dialog

1. Select the Link Type:
   Use ‘S’ to compare with the Target Speed, ‘s’ to compare with the Actual Speed.
2. Enter the Limit speed in the Link Value.

Link Type - Current Axis Status Bits

Link Type: B (on) or b (off)

Link Value: Bit pattern that must be set or cleared

Range: Any bits

These link types wait for at least one of a user-selectable group of Status Bits to be set or cleared on the current axis.

Using with the *Link Type and Link Value* Popup Dialog

1. Under **Link Type Category**, select **Current Axis (Basic)**.
2. Under **Link Type**, select **Status Bits**.
3. Under **Link Condition**, select the appropriate option for whether you wish to wait for one or more bits to be ON or OFF.
4. Under **Link Condition**, select the check boxes next to the Status bits that you wish to monitor with this link type.

Using without the *Link Type and Link Value* Popup Dialog

1. Select the Link Type:
   Use 'B' if you wish to wait for Status Bit(s) to be on, 'b' if you wish to wait for Status Bit(s) to be off.
2. Enter the Link Value. Use the STATUS Word Bit Map to calculate a 16-bit hexadecimal word that matches the bits you wish to wait for.

Selected Axis Link Types

Link Type - Any Axis Position

Link Type: See below

Link Value: Limit Position

Range: Any valid position in Position Units

These link types are used to detect when the position of the axis selected by the link type is above or below a limit. This link type can be used to change speeds on-the-fly or trigger events on another axis. The Link Value holds the Limit position in position units. Either the Target Position or Actual Position may be used to compare with the Limit position.

Using with the *Link Type and Link Value* Popup Dialog

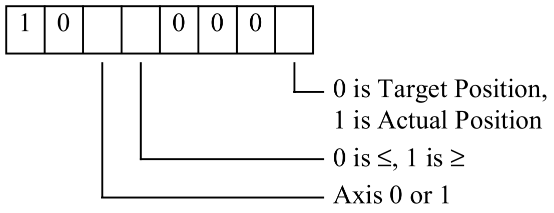
1. Under **Link Type Category**, select **Selected Axis (Enhanced).**
2. Under **Axis**, select the axis you desire to monitor.
3. Under **Link Type**, select **Position**.
4. Under **Link Condition**, select whether you wish to use Target or Actual Position for the comparison.

5. Under **Link Condition**, select whether you wish to wait until the position is *above or equal*, or *below or equal* the Limit position.
6. Under **Link Condition**, enter the Limit position in the **Threshold** box.

Using without the *Link Type and Link Value* Popup Dialog

1. Calculate the Link Type:

   This involves converting the following bit fields into a hexadecimal byte. It is highly recommended that the **Link Type and Link Value** popup dialog box be used instead of doing this manually. Use the following diagram to calculate the byte in binary and then convert to hexadecimal or decimal and enter in the **Link Type** field.



2. Enter the Limit (threshold) position in the **Link Value** field.

Link Type - Any Axis Speed

Link Type: See below

Link Value: Limit Speed

Range: 0 to 65,535 position units per second

These link types are used to detect when the speed of the axis selected by the link type is above or below a user-specified value. This link type can be used to change speeds on-the-fly or trigger events on another axis. The Link Value holds the Limit speed in position units per second. Either the Target Speed or Actual Speed may be used to compare with the Limit speed.

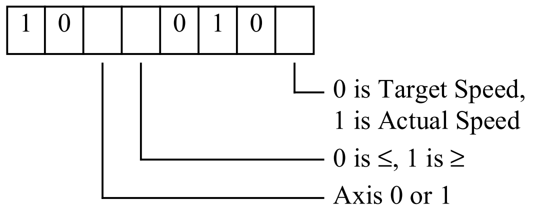
Using with the *Link Type and Link Value* Popup Dialog

1. Under **Link Type Category**, select **Selected Axis (Enhanced).**
2. Under **Axis**, select the axis you desire to monitor.
3. Under **Link Type**, select **Speed**.
4. Under **Link Condition**, select whether you wish to use Target or Actual Speed for the comparison.
5. Under **Link Condition**, select whether you wish to wait until the speed is *above or equal*, or *below or equal* the Limit speed.
6. Under **Link Condition**, enter the Limit speed in the **Threshold** box.

Using without the *Link Type and Link Value* Popup Dialog

1. Calculate the Link Type:

   This involves converting the following bit fields into a hexadecimal byte. It is highly recommended that the **Link Type and Link Value** popup dialog be used instead of doing this manually. Use the following diagram to calculate the byte in binary and then convert to hexadecimal or decimal and enter in the **Link Type** field.



2. Enter the Limit (threshold) speed in the **Link Value** field.

Link Type - Any Axis Status Bits

Link Type: See below

Link Value: Bit pattern that must be set or cleared

Range: Any bits

These link types wait for at least one of a user-selectable group of Status Bits to be set or cleared on the axis selected by the link type.

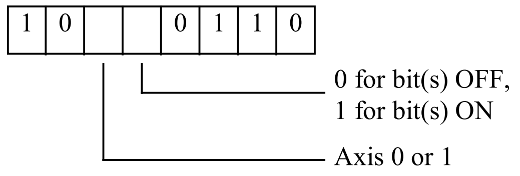
Using with the *Link Type and Link Value* Popup Dialog

1. Under **Link Type Category**, select **Selected Axis (Enhanced).**
2. Under **Axis**, select the axis you desire to monitor.
3. Under **Link Type**, select **Status Bits**.
4. Under **Link Condition**, select the appropriate option for whether you wish to wait for one or more bits to be ON or OFF.
5. Under **Link Condition**, check the boxes next to the Status bits that you wish to monitor with this link type.

Using without the *Link Type and Link Value* Popup Dialog

1. Calculate the Link Type:

   This involves converting the following bit fields into a hexadecimal byte. It is highly recommended that the **Link Type and Link Value** popup dialog be used instead of doing this manually. Use the following diagram to calculate the byte in binary and then convert to hexadecimal or decimal and enter in the **Link Type** field.



2. Enter the Link Value. Use the STATUS Word Bit Map to calculate a 16-bit hexadecimal word that matches the bits you wish to wait for.

Appendix F: MMC120 Specifications

MMC120 Specifications

Power Requirements

Back plane	+5V @ 1 Amp

Environment

Operating temperature	+32 to +140°F (0 to +60°C)
Non-operating temperature	-40 to +185°F (-40 to +85°C)
Storage temperature	-40 to +185°F (-40 to +85°C)
Humidity	93%, non-condensing
Agency compliance	UL, CSA, CE
ESD Immunity	8kV
Vibration resistance	1 g at 60 Hz to 500 Hz for 23 minutes 0.075 mm displacement from 10 Hz to 60 Hz
Shock resistance	30 g for 11 milliseconds

Transducer Inputs

MDT interface	Direct connection
Signal compatibility	Differential (RS-422) or single-ended (TTL level)
Transducer compatibility	Start/Stop or Pulse Width Modulated

Drive Inputs

Output isolation	Optically isolated, 2500VAC withstand voltage
Range	±10V @ 5mA (2kΩ or greater load) (use optional VC2100 module for current drive; ±10mA to ±100mA in 10mA steps)
Resolution	12 bits

Overload protection	One-second short-circuit duration
Overvoltage protection	Outputs are protected by clamp diodes

Bus Compatibility

ModConnect®	Certified partners
Programmable Controller	Quantum Automation Series
Quantum back plane	Direct connection
Inputs/outputs	4 input and 4 output registers, using 64 input and 64 output points
Storage register requirement	32 registers for parameters (16 per axis) 32 registers max. for motion profiles 2048 registers max. for event steps

Failsafe Timers

No internal bus activity	After 15µs, the drive outputs are disabled
Software watchdog	After 17ms, the module resets if the microprocessor does not retrigger the onboard watchdog timer

FLASH Memory Storage

Write/erase Cycles	10,000 minimum

Appendix G: Glossary

Glossary

Closed Loop Mode: Sometimes called Servo mode. In this mode the difference between the Target and Actual position/pressure is the error that the PID routine uses to compute a corrective drive that minimizes the error.

Extending: Going in the direction where the transducer's counts are increasing.

Hard Stop: An emergency stop condition where the drive is immediately set to the null drive value.

LDT: Linear displacement transducer. Technically, this is any sensor that can sense a linear position. Many use this term interchangeably with MDT.

MDT: Magnetostrictive displacement transducer. A device that senses position by sending an electron pulse down a wave guide. A twist is imparted on the wave guide as the pulse reaches the magnetic field of a magnet. The twist takes time to be sensed at the transducer head. It is this time that is measure by the controller board and it is proportional to the distance between the transducer head and the magnet.

NOTE: Balluff and Temposonics rods are both MDTs whereas any linear measurement device can be a LDT.

Open Loop Mode: The drive output is set to a value with out any corrective action taken by the PID routine.

PID:Proportional Integral Derivative. A simple algorithm used by the motion controller to reduce the error between the target and actual position, pressure, etc. The three words of this acronym represent the three types of gains used for controlling a system. See PID Loop below.

PID Loop:The process of repeatedly executing the PID routine

Error: = Target - Actual
E0: Current Error, Error for current Target and Actual
E1: Previous Error, Error for previous Target and Actual
U0: Current output
U1: Previous output
Kp: Proportional Gain
Ki: Integral Gain
Kd: Differential Gain

Proportional Term:	U0p = Kp * E0
Integral Term:	U0i = U1i + Ki * E0
Differential Term:	U0d = (E0 - E1) * Kd
The Sum is:	U0 = U0p + U0i + U0d

The PID LOOP is:

```
DO FOREVER
    WAIT FOR NEXT TIME PERIOD
    READ ACTUAL FROM POSITION OR PRESSURE SENSOR
    E0 = TARGET - ACTUAL

    U0p = Kp * E0
    U0i = U1i + Ki * E0
    U0d = (  E0 - E1 ) * Kd
    U0 = U0p + U0i + U0d
    E1 = E0
    CALCULATE NEXT TARGET POSITION
END
```

Retracting: Going in the direction where the transducer’s counts are decreasing.

Soft Stop: An emergency stop condition where the drive ramps down to the null drive value.

Target Generator: The part of the motion control software that determines where the axis should be at any moment using the requested position, speed, acceleration and deceleration provided by the user. A different target generator is used for position control, speed control, gearing, synchronized moves and pressure control.

Index