

**MMC120**  
**Linear Motion Control Module**  
**Rev 1.00**  
**991221**

Delta Computer Systems, Inc.

11719 NE 95th Street, Suite D  
Vancouver, WA 98682-2444

Tel: 360/254-8688

Fax: 360/254-5435

<http://www.deltacompsys.com>

[email@deltacompsys.com](mailto:email@deltacompsys.com)

Although great effort has been taken to ensure the accuracy of the information in this document, 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 document. The information in this document 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.

<b>OVERVIEW</b> .....	<b>9</b>
FEATURES .....	9
General .....	9
Position Inputs .....	9
Drive Outputs.....	10
<b>SPECIFICATIONS</b> .....	<b>11</b>
TEMPERATURE .....	11
HUMIDITY .....	11
VIBRATION RESISTANCE .....	11
SHOCK RESISTANCE.....	11
ESD IMMUNITY .....	11
POWER REQUIREMENTS .....	11
TRANSDUCER INPUTS.....	11
OUTPUTS .....	11
ISOLATION .....	11
I/O PROTECTION .....	11
AGENCY COMPLIANCE.....	11
BUS COMPATIBILITY.....	12
FAIL SAFE TIMERS .....	12
Drive Output Disable -- 15 microseconds .....	12
Hardware Reset -- 17 milliseconds .....	12
<b>DESCRIPTION</b> .....	<b>13</b>
PRINCIPLE OF OPERATION.....	13
Position Measurement .....	13
Control Loop.....	14
Drive Output .....	15
Quantum Interface .....	15
Programming .....	15
Event Control.....	15
FLASH Memory.....	15
<b>FRONT PANEL INDICATORS</b> .....	<b>16</b>
<b>SETUP CHECKLIST</b> .....	<b>17</b>
WIRING (PAGES 18 AND 19).....	17
INSTALLATION (PAGE 18) .....	17
START-UP AND TUNING.....	17
<b>PREPARING FOR INSTALLATION</b> .....	<b>18</b>
WIRING NOTES.....	18
DRIVE OUTPUTS .....	18
MAGNETOSTRICTIVE TRANSDUCER WIRING.....	18
SERIAL PORT .....	20
CONNECTOR INFORMATION .....	20
HYDRAULIC SYSTEM NOTES .....	21
<b>STARTUP</b> .....	<b>22</b>
INITIALIZATION PARAMETERS .....	22
SCALE Parameter (Default: 30300) Range: -32767 to 32767, excluding 0.....	23
OFFSET Parameter (Default: 0) Range: -65535 to 65535 position units .....	23
MOVING THE AXIS .....	23
Powering Up .....	24
TUNING .....	25
JOGGING THE AXIS.....	26

SAVING PARAMETERS AND PROFILES.....	26
<b>MOTION CONTROL PARAMETERS .....</b>	<b>27</b>
INITIALIZATION PARAMETERS .....	27
CONFIGURATION Word (Default: 0000).....	27
CONFIGURATION Word Bit Map.....	29
SCALE (Default: 30300) Range: -32767 to 32767, excluding 0 .....	30
SCALE Calculation Examples .....	30
OFFSET (Default: 0) .....	31
EXTEND LIMIT (Defaults to current position on power-up) .....	31
RETRACT LIMIT (Defaults to current position on power-up) .....	31
PROPORTIONAL GAIN (Default: 1) Range: 0 to 65535 .....	31
INTEGRAL GAIN (Default: 1) Range: 0 to 65535.....	32
DIFFERENTIAL GAIN (Default: 0) Range: 0 to 65535.....	32
EXTEND FEED FORWARD (Default: 100) Range: 0 to 65535.....	32
RETRACT FEED FORWARD (Default: 100) Range: 0 to 65535.....	32
EXTEND ACCELERATION FEED FORWARD (Default: 0) Range: 0 to 65535 .....	33
RETRACT ACCELERATION FEED FORWARD (Default: 0) Range: 0 to 65535 .....	33
DEAD BAND ELIMINATOR (Default: 0) .....	33
IN POSITION (Default: 50).....	33
FOLLOWING ERROR (Default: 250).....	33
AUTO STOP (Default: 0FFE0 - Soft Stops enabled, Hard Stop enabled .....	33
MOTION CONTROL COMMANDS .....	35
MODE (Default: 00000)(See page 36 for the MODE Word bit map).....	35
ACCELERATION (Default: 1000) .....	36
MODE Word Bit Map.....	37
DECELERATION (Default: 1000).....	38
SPEED (Default: 1000).....	38
COMMAND VALUE .....	38
COMMAND .....	38
Move Profiles.....	39
MMC120 Initial Move Profiles .....	40
READBACK PARAMETERS.....	41
ACTUAL POSITION.....	41
AXIS STATUS Word (See map on page 42) .....	41
COMMAND POSITION.....	43
TARGET POSITION.....	43
TRANSDUCER COUNTS .....	44
ACTUAL SPEED .....	44
DRIVE.....	44
STATUS Word Bit Map.....	45
<b>FLASH MEMORY .....</b>	<b>46</b>
<b>COMMUNICATING WITH THE MMC120.....</b>	<b>47</b>
QUANTUM BUS CONFIGURATION.....	47
MEMORY REQUIREMENTS.....	47
I/O REGISTER OVERVIEW .....	48
OUTPUT REGISTERS.....	49
COMMANDS .....	49
Command Types .....	50
Command Type 1 - Go Using Profile (0000 XXXX 0001 NNNN).....	51
Command Type 2 - Set Profiles (0000 XXXX 0010 NNNN) .....	52
Command Type 3 - Set Parameters (0000 XXXX 0011 NNNN).....	54
Command Types 4, 5, 6, and 7 - ASCII Commands (0000 XXXX 01NN NNNN).....	55

Command Type 9 - Open Loop Using Profile (0000 XXXX 1001 NNNN) .....	59
Command Type A - Get Profiles (0000 XXXX 1010 NNNN).....	60
Command Type B - Get Parameters (0000 XXXX 1011 NNNN).....	62
Command Type E - Event Step Edit (0000 XXXX 1110 00NN).....	63
Command Type F - Diagnostics (0000 XXXX 1111 XXXX).....	63
Extended Command Type E - Event Step Transfer (1110 DSSS SSSS SNNN) .....	64
INPUT REGISTERS .....	65
<b>QUANTUM PROGRAMMING HINTS.....</b>	<b>66</b>
ERROR HANDLING .....	66
<b>TROUBLESHOOTING .....</b>	<b>67</b>
PROBLEMS AND SOLUTIONS .....	67
Ladder program cannot access parameters or operate module.....	67
Red LEDS 2, 3, 4, 6, 7, or 8 are on.....	67
Red LEDS 2 and 6 are on; all other red LEDS are flashing; no green LEDS on.....	67
During a move, the Actual Position is erratic .....	67
During a move, the drive comes to a halt for no apparent reason .....	67
Transducer counts field not indicating transducer location .....	67
Transducer counts field changes but output drive does not work .....	67
The System is unresponsive and hard to tune .....	68
The axis oscillates .....	68
The axis does not finish moves or moves differently than expected.....	68
<b>HYDRAULIC SYSTEM PROBLEMS.....</b>	<b>69</b>
NONLINEAR VALVES.....	69
SLOW-RESPONSE VALVES .....	69
HOSES.....	69
PUMPS AND ACCUMULATORS .....	70
IDENTIFICATION AND CORRECTION .....	70
<b>REPAIRS AND RETURNS.....</b>	<b>73</b>
WARRANTY .....	73
MODULE RETURN FOR REPAIR .....	73
RETURNS .....	73
DELTA BULLETIN BOARD SYSTEM.....	<b>ERROR! BOOKMARK NOT DEFINED.</b>
DELTA WEB PAGE AND EMAIL .....	73
<b>GLOSSARY OF TERMS.....</b>	<b>74</b>
<b>ASCII TABLE .....</b>	<b>75</b>
<b>APPENDIX A: DCS120 DIAGNOSTIC AND SETUP PROGRAM.....</b>	<b>76</b>
DESCRIPTION .....	76
GENERAL FEATURES.....	76
INITIAL STARTUP USING DISPLAY-ONLY MODE .....	77
INITIAL STARTUP USING MOTION CONTROL MODULE .....	79
THE DCS120 SCREEN .....	80
Read Only Parameters.....	80
Command Parameters .....	80
Input Field .....	80
Initialization Parameters .....	80
Command Queue .....	80
SUMMARY OF KEYBOARD COMMANDS .....	81
DIAGNOSTIC GRAPHS.....	83
KEYBOARD COMMANDS .....	84
Getting Help .....	84

Function Keys .....	84
Exiting DCS120 Screens.....	84
Displaying Axis Status Information .....	84
Editing the .INI File.....	84
Motion Control Commands.....	84
Open Loop Command.....	85
Readback and Write Mode Toggle .....	85
SCALE & OFFSET Calculation Commands .....	85
OFFSET Parameter.....	85
VARIABLE DATALOGGING RATES.....	85
MOVING AXES SIMULTANEOUSLY .....	86
TABLE EDITORS .....	86
ADVANCED FEATURES.....	87
DOS Command Line Options .....	87
SETTING UP ADVANCED FEATURES.....	87
DESCRIPTION OF INITIALIZATION FILE AND PROJECT FILES.....	87
Initialization File .....	88
Project Files.....	90
USER NOTES .....	93
Negative Offset .....	93
View Data Command.....	93
Sum of Errors Squared ( $\sum e^2$ ).....	93
MMC120 Communication Cable.....	94
<b>APPENDIX B: EVENT CONTROL.....</b>	<b>95</b>
INTRODUCTION .....	95
OVERVIEW .....	95
STEPS .....	95
Commands.....	95
Link Word .....	95
Step Table.....	96
LINKING STEPS.....	97
Link Next .....	97
Link Type and Link Value .....	97
STEP EXECUTION .....	99
Example .....	100
STARTING A SEQUENCE .....	100
Triggering Events.....	100
Ending a sequence .....	100
USING DCS120.....	101
Accessing the DCS120 Step Editor screen.....	101
Step Editor Commands .....	101
Reading Step files from disk and module .....	102
Sending Step files to disk, module, and Flash memory .....	102
NOTES .....	103
Reading the Current Step.....	103
Terminating a Loop .....	103
Interrupting a Sequence .....	103
Multiple Axes Can Use the Same Steps.....	103
No Conditional Branching .....	103
<b>INDEX .....</b>	<b>104</b>







## 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  $\pm 10$  volt outputs.

The MMC120 occupies one slot of the Quantum rack. The MMC120 and the Programmable Controller communicate over the backplane through four output and four input registers. The onboard 80C188EB 20MHz processor relieves the Programmable Controller of the overhead needed for servo control. The Motion Control Module updates the axis position and drive output 1000 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

### General

- Modicon TSX Quantum Automation Series® compatible (ModConnect® Certified)
- Uses one slot of Quantum rack
- Hot swappable
- Linear position control
- 80C188EB processor operating at 20 MHz
- 1 millisecond control loop time
- Magnetostrictive Transducer Interface
- Maximum velocity in excess of 120 inches per second
- Maximum transducer length greater than 220 inches
- Hardware fail safe timers
- FLASH memory for parameter and profile storage
- Software configurable -- no jumpers or switches
- Diagnostic and tuning software included

### Position Inputs

- Two optically isolated magnetostrictive transducer interfaces
- Differential or single-ended signal compatibility
- Start/Stop or Pulse Width Modulated signal compatibility
- 120 MHz count rate -- counting both edges of a 60 MHz clock
- 1 circulation
- Position resolution better than 0.001 of an inch

Modicon, ModConnect and TSX Quantum Automation Series are registered trademarks of Schneider

**Drive Outputs**

- Two optically isolated  $\pm 10$  volt outputs
- 12 bit digital-to-analog converters
- Current output available with VC2100 voltage-to-current option

## SPECIFICATIONS

### Temperature

Operating: 0°C to +60 °C

Storage: -40°C to +85 °C

### Humidity

93% non-condensing

### 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

### ESD Immunity

8 kV

### Power Requirements

+5V @ 1 Amp from the backplane

### Transducer Inputs

Direct interface to magnetostrictive transducers

Differential or single-ended transducer compatibility

Start/Stop pulses or Pulse Width Modulated signals

### Outputs

±10 volts into 2000 Ohm or greater load -- current output is available with the optional VC2100 voltage-to-current converter

### Isolation

All inputs and outputs used for control purposes are optically isolated from the module, with 2500VDC withstand voltage

Each input and output section is isolated from the other sections

The RS-232 port is not isolated

### I/O Protection

All inputs and outputs are protected against transients by clamp diodes

### Agency Compliance

UL, CSA, and CE

## Bus Compatibility

ModConnect® Certified partners

Quantum Automation Series

Direct connection to Quantum backplane

4 input and 4 output registers, using 64 input and 64 output points

32 registers for parameter storage (16 per axis)

32 registers maximum for profile storage (16 per axis)

2048 registers maximum for event steps (256 steps)

## Fail Safe Timers

The MMC120 has two fail safe timers:

### Drive Output Disable -- 15 microseconds

The fastest fail safe timer on the MMC120 is 15 microseconds long and is retriggered by activity on the module's internal bus. If the timer is not retriggered within 15 microseconds, it disables the drive outputs.

When internal bus activity resumes, the drive outputs are re-enabled.

### Hardware Reset -- 17 milliseconds

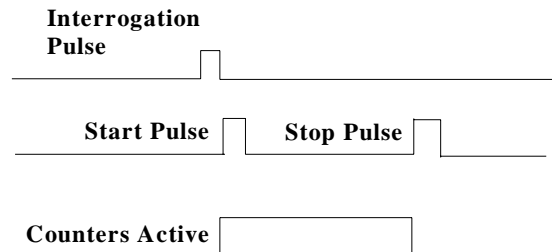
If the microprocessor fails to retrigger the onboard watchdog timer, the 17 millisecond hardware reset timer expires and resets the hardware. This has the same effect on the MMC120 as powering up. While the module is being reset, the Fail LED on the front panel is illuminated.

## DESCRIPTION

### Principle Of Operation

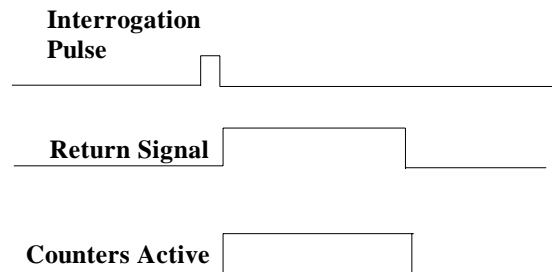
#### Position Measurement

The MMC120 has interface circuitry for two magnetostrictive transducers. Each axis can be configured for a Start/Stop transducer or a Pulse Width Modulated transducer by changing the axis' Configuration Word. To make a measurement with a Start/Stop transducer, the MMC120 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 MMC120 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 MMC120 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 MMC120 are active while the return signal is high. The width of the return signal is proportional to the transducer position.



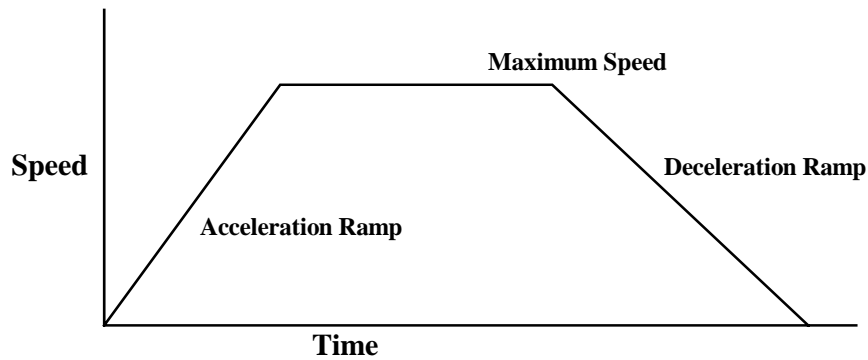
**Pulse Width Modulated Transducer**

While the counters on the MMC120 are active, they count the number of half-cycles of a 60 MHz clock (120 MHz effective rate). At this frequency, each half-cycle represents less than 0.001 inch displacement of the axis.

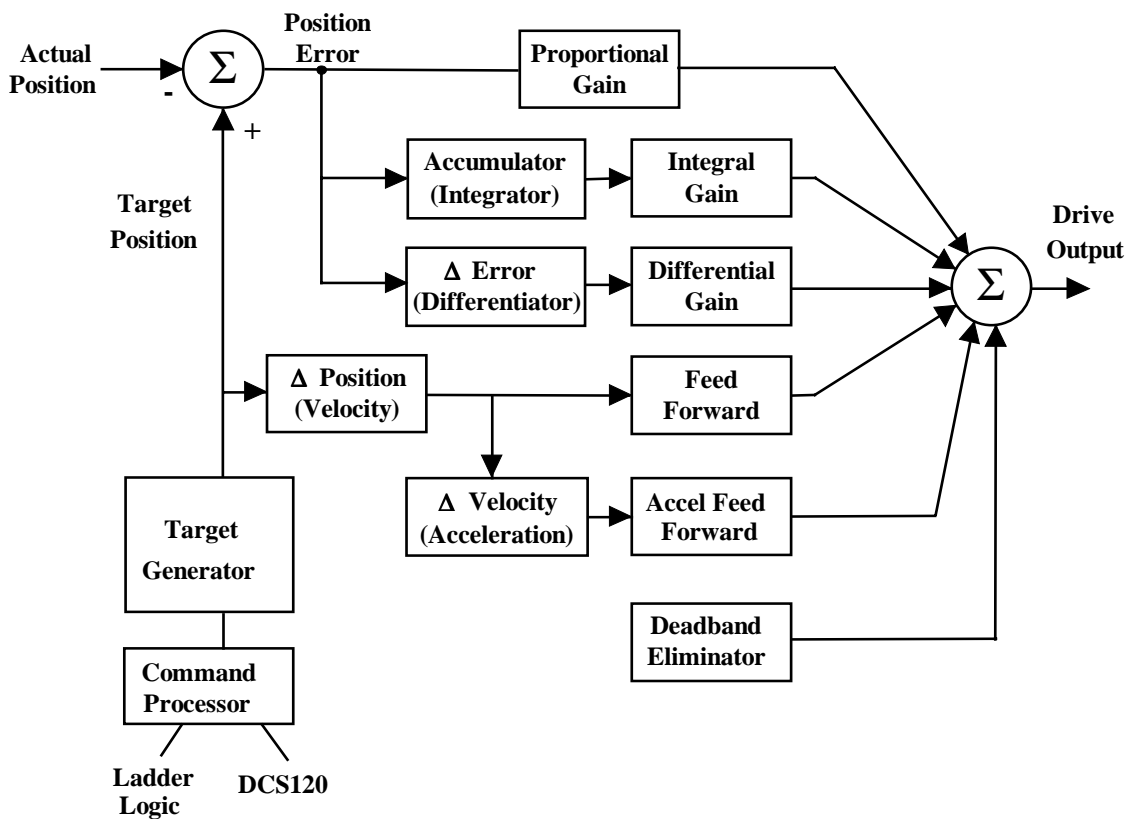
The MMC120 converts the Transducer Counts read from the counters to an ACTUAL POSITION in user-defined Position Units (usually 0.001 inch) for use by the Quantum Programmable Controller.

## Control Loop

The MMC120 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.



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, the MMC120 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 nonlinearity in the system and changes in system load.

### Drive Output

The drive generated by the MMC120 is sent through optical isolation to a 12 bit digital-to-analog converter (DAC). The output from the DAC is amplified to provide a  $\pm 10$  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 backplane. 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 backplane in groups of four 16-bit words.

### Programming

Commands to the MMC120 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 the section "COMMUNICATING WITH THE MMC120."

### Event Control

Sequences of commands can be stored and executed by the MMC120 with little intervention by the Programmable Controller. This allows a 1 millisecond response time by the MMC120 to internal events such as move done or elapsed time.

### FLASH Memory

You can store parameters, profiles, and Event Control steps in the MMC120'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 ladder logic program or in a DCS120 file so they can be transferred to a module when needed.

---

## FRONT PANEL INDICATORS

There are 14 green and red light emitting diodes (LEDS) 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 valve is out of null.

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

---

Fail indicator constant red	Onboard processor halted; drive outputs disabled
Active indicator off	Communication disabled
Active indicator 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 (Emergency Stop)	Axis 1 Hard Stop/Open Loop
Red LED 2 (No MDT)	Axis 1 No Transducer
Red LED 3 (Noise)	Axis 1 Transducer Noise
Red LED 4 (Overflow)	Axis 1 Transducer Overflow
Red LED 5 (Emergency Stop)	Axis 2 Hard Stop/Open Loop
Red LED 6 (No MDT)	Axis 2 No Transducer
Red LED 7 (Noise)	Axis 2 Transducer Noise
Red LED 8 (Overflow)	Axis 2 Transducer Overflow



## SETUP CHECKLIST

### Wiring (Pages 18 and 19)

Wire magnetostrictive transducers to 9-pin terminal block on MMC120. See your transducer documentation for connector pinout or cable color code information.

Wire MMC120 drive outputs (3-pin terminal block) to actuators (+out = axis extends)

Install and connect power supplies needed for your transducers

### Installation (Page 18)

Define input and output registers

I/O Map the MMC120 (DCS MMC 120 0x, 4 input registers, 4 output registers)

Reserve memory for the module (parameters, profiles, event steps)

### Start-up and Tuning

Run DCS120 setup program

Configure MMC120 to match transducers (See Configuration Word, bits 1 - 4 on page 25)

Set Extend and Retract Limits

Tune the axes

Save parameters in DCS120 to a .BD1 file (*S filename* command - page 78)

Save parameters to Flash memory (page 43)

Read parameters and profiles into Programmable Controller

## PREPARING FOR INSTALLATION

### 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.

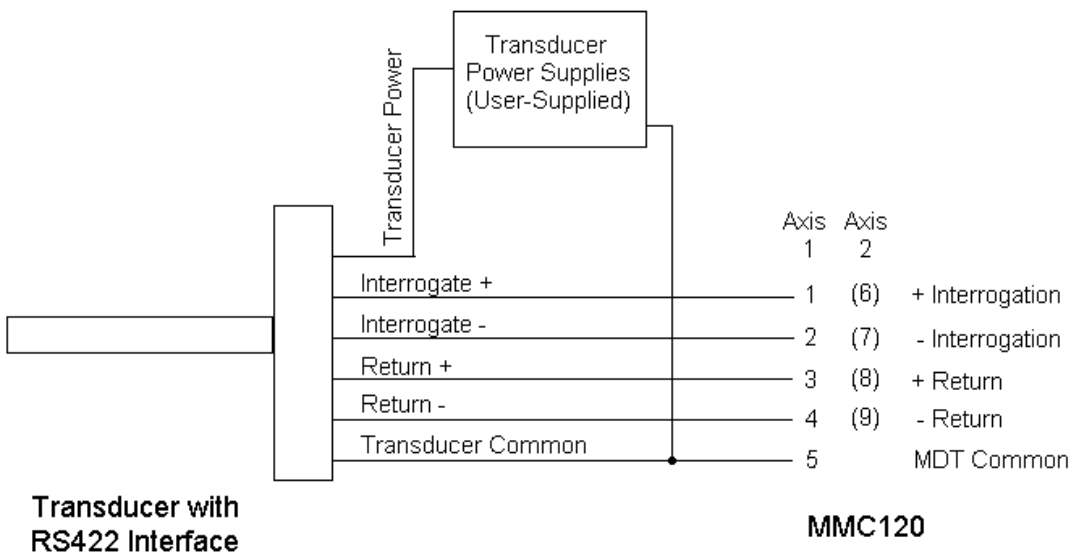
### Drive Outputs

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.

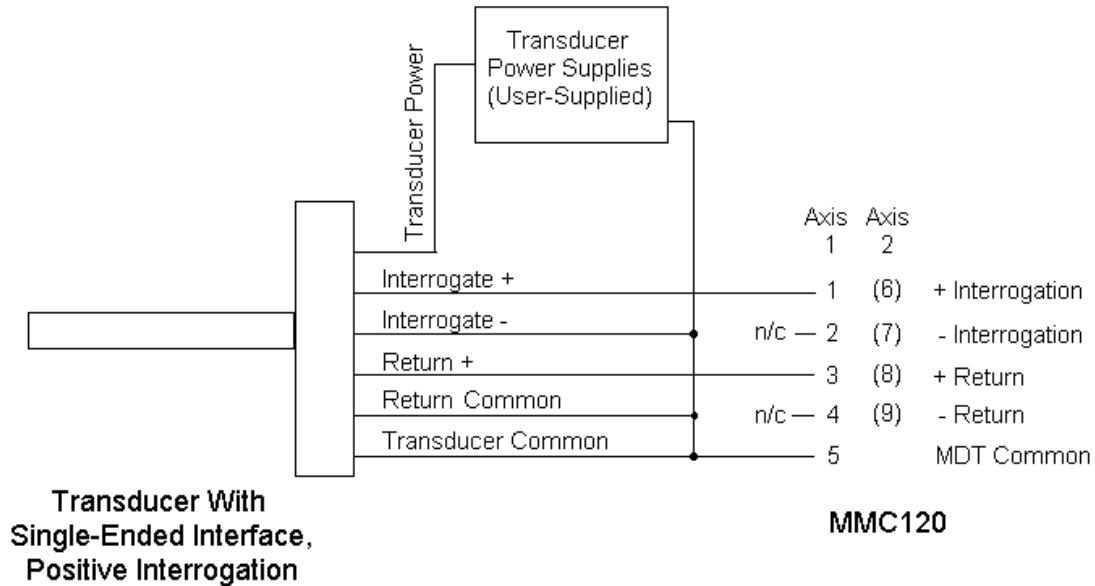
### Magnetostrictive Transducer Wiring

**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 pinouts or cable color codes, consult your transducer documentation.

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

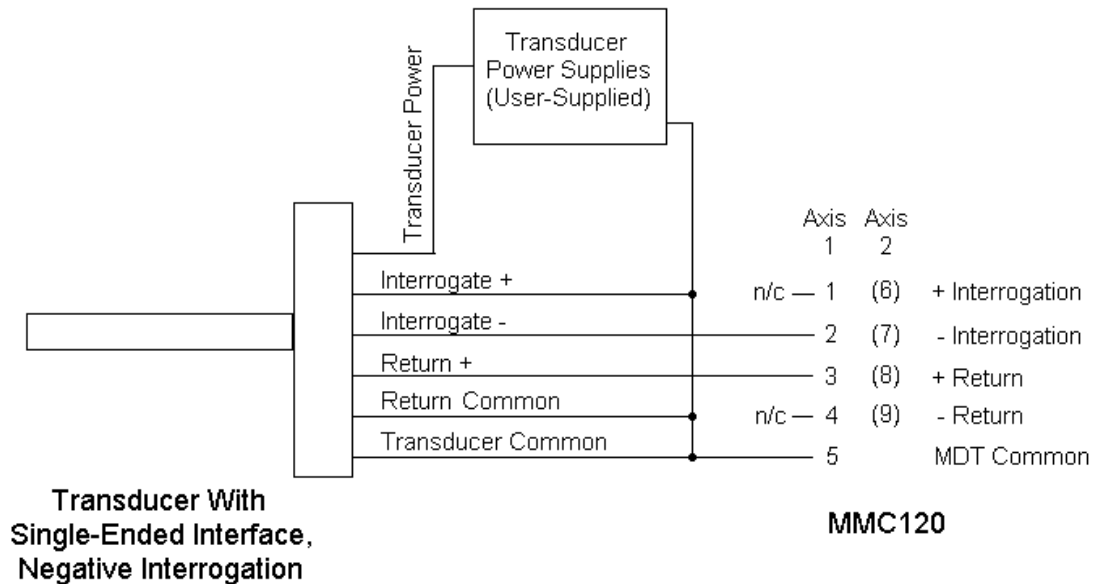


For single-ended transducer with positive interrogation, connect the transducer '- interrogation in' wire to the MDT Common pin and the transducer '+ interrogation in' wire to the '+ Int' pin. **CONNECT NOTHING TO THE '- INT' PIN OF THE MMC120.** Connect the transducer return plus wire to the '+' return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

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 which is as short as possible. Route it out the bottom of the module and against the back panel.

## Connector Information

Transducer interface:

### 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 transducer interface can be used with either differential (RS-422) or single-ended (TTL) transducer signals. For an RS-422 transducer, 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 a positive interrogation, single-ended transducer, connect the transducer '- interrogation in' wire to the MDT Common 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 '+' return pin on the MMC120 and the transducer return common wire to MDT Common on the MMC120. **CONNECT NOTHING TO THE '- RETURN' PIN OF THE MMC120.**

For a negative interrogation, single-ended transducer, connect the transducer '+ interrogation in' wire to the MDT Common 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 '+' return pin on the MMC120 and the transducer return common wire to MDT Common on the MMC120. **CONNECT NOTHING TO THE '- RETURN' PIN OF THE MMC120.**

Drive Output:

### Three Pin Plug-in Terminal Block

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

Serial Port:

### DB-9

Pin	Function
2	Receive
3	Transmit
5	Common

## Hydraulic System Notes

Hydraulic systems must have enough pressure and fluid volume (accumulator) to move the desired load the commanded distance at the commanded speed. Inadequate pressure or volume will cause the axis to lag the target position as the controller attempts to move the axis faster than the system can move.

There should be no flexible hose between the valve and the cylinder being controlled. Hose swells and contracts as the valve opens and closes, causing oscillation and loss of control.

If proportional valves are used they should have zero overlap and a linear (not curvilinear) response. Overlapped or nonlinear valves may cause oscillation or hunting.

Valves with slow response cause the Motion Control Module (MCM) to overcompensate for disturbances in the motion of the system. With a slow valve the system will not respond immediately to the control signal, so 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.

## STARTUP

**CAUTION:** Do not connect the drive outputs to the drives until the EXTEND and RETRACT limits have been determined and the MMC120 has been properly initialized with these limits by the Quantum Programmable Controller.

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.

When the MMC120 Motion Control Module is first turned on the parameters are set to default values. These values allow the module to maintain the present positions of the axes it controls; the parameters must be changed for the module to work in your system. The Quantum Programmable Controller program is responsible for initializing the parameters when the MMC120 is powered up by sending the correct initialization parameters to the module, then issuing a 'P' command. The following procedure will help find the optimum values for the initialization parameters.

**TIP:** Use Delta's DCS120 setup program to help determine initial parameters and operation. Also, 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. See APPENDIX A: DCS120 DIAGNOSTIC AND SETUP PROGRAM.

Configure DCS120 to communicate with the module. Make sure the transducers are connected and powered up. Disconnect the drive output from the MMC120 (3 pin connector), and arrange a way to manually move the axis, either physically or with a control box that can electrically drive the valve (a 'diddle box'). If neither of these options are usable to you, the procedure shown in 'Moving the Axis' (pages 21 and 22) uses DCS120 and the Open Loop mode to slowly move the axis.

## Initialization Parameters

If you want to get started quickly, go to the next section - Moving The Axis. Continue with this section now or return to it later to set the SCALE and OFFSET parameters.

The easiest way to set the SCALE and OFFSET is to use the P0 and P1 commands in DCS120. To use these commands, you need to know the physical position and the corresponding transducer counts at two points (usually the extend and retract limits). For example, if the first position is 0.000 inches and the counts are 2163, and the second position is 27.500 inches and the counts are 31626, then enter:

```
P0 0 2163 <ENTER>
P1 27500 31626 <ENTER>
```

After you enter values for both P0 and P1, the SCALE and OFFSET fields will be updated on the screen (notice the parameters turn gray). Use the alt-P command to activate the new values. See Appendix A for more information about the P0 and P1 commands.

**SCALE Parameter** (Default: 30300) Range: -32767 to 32767, excluding 0

The SCALE factor converts TRANSDUCER COUNTS to Position Units. The SCALE is defined as:

$$SCALE = \frac{\text{Position Units per Inch} \times 32768}{\text{Transducer Cal. Number } (\mu\text{s/in.}) \times 120 \text{ MHz}} \times \text{Sign}$$

Where 'Sign' equals +1 when an extend move yields increasing ACTUAL POSITION, and -1 when an extend move yields decreasing ACTUAL POSITION. An extend move **always** yields increasing TRANSDUCER COUNTS, but the sign of SCALE determines whether extending the axis will give increasing or decreasing ACTUAL POSITION.

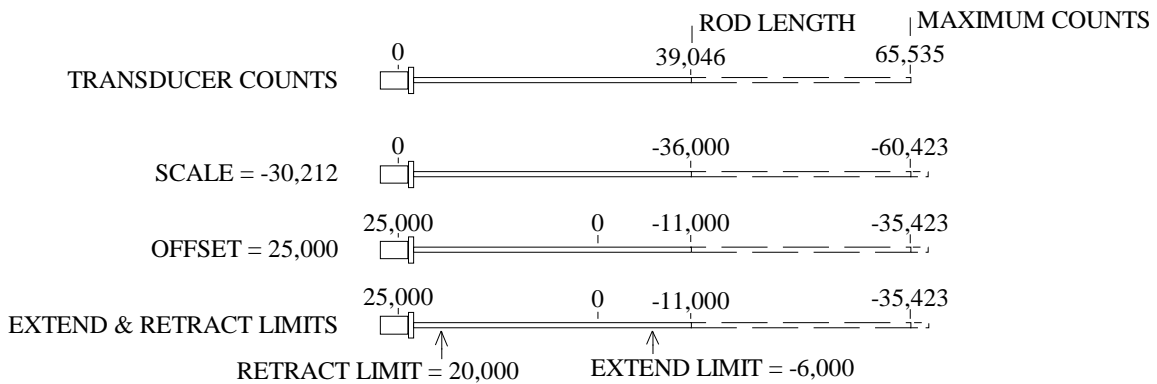
The calibration number must be specified in the same base units as the Position Units (inches, mm, etc.). Position Units are generally 0.001 inches or 0.1 mm. There are usually 1000 Position Units per inch when the calibration number is in microseconds per inch.

**OFFSET Parameter** (Default: 0) Range: -65535 to 65535 position units

Next, the axis OFFSET must be set. The axis OFFSET is added to the position of the axis when the ACTUAL POSITION is computed. When the OFFSET is changed make sure to change the EXTEND LIMIT and RETRACT LIMIT by the same amount. Use the following equation to adjust the current ACTUAL POSITION to be something else:

$$OFFSET = \text{old OFFSET} + \text{DESIRED POSITION} - \text{ACTUAL POSITION}$$

Both the EXTEND and RETRACT limits must be adjusted in the same way as the OFFSET, and another 'Alt-P' command must be issued. Move the axis to the EXTEND and RETRACT limits and make any adjustments necessary.



**Moving The Axis**

With the axis drive output still disconnected and the transducer on, move the axis close to the physical extend limit (move it either manually or with a 'diddle box') - be sure to leave some space for safety. If neither of these movement options are usable to you, the following procedure uses DCS120 and the Open Loop mode to slowly move the axis. (Remember: extending is the direction that returns increasing TRANSDUCER COUNTS.)

## Open Loop Procedure

**CAUTION - OPEN LOOP OPERATION IGNORES ALL LIMITS! BE PREPARED TO REMOVE DRIVE POWER!**

- A. Connect the MMC120 drive output to the valves.
- B. In DCS120, with the cursor on the axis you want to adjust, enter Alt-R to restore the null. Put 0 (zero) in the COMMAND VALUE field, enter Alt-Shift-O, and verify that DRIVE = 0. If the NULL DRIVE is not zero, enter Alt-N to zero it.
- C. Turn on the hydraulics (the axis will drift due to valve null errors).
- D. Put 500 in the COMMAND VALUE field and enter Alt-Shift-O, then put 0 (zero) in the COMMAND VALUE field. Verify that DRIVE = 500. The axis should extend. If the axis retracts, check the drive wiring polarity, hydraulic plumbing, and the value null. Correct any problems before continuing.
- E. When the axis gets to the extend limit, enter Alt-Shift-O, then Alt-P to stop the axis. Enter the ACTUAL POS. value in the EXTEND LIMIT parameter.
- F. Put -500 in the COMMAND VALUE field and enter Alt-Shift-O, then put 0 (zero) in the COMMAND VALUE field. Verify that DRIVE = -500. The axis should retract.
- G. When the axis gets to the retract limit, enter Alt-Shift-O, then Alt-P to stop the axis. Enter the ACTUAL POS. value in the RETRACT LIMIT parameter.
- H. Repeat Steps B through G for each axis in use.

Enter the ACTUAL POSITION reading as the value for the EXTEND LIMIT parameter. Now move the axis close to the physical retract limit and set the RETRACT LIMIT parameter. Issue an 'alt-P' command. (If you are using DCS120, be sure to read the parameters into the Programmable Controller's memory using Command Type B.)

## Powering Up

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

**NOTE:** To input hexadecimal numbers to DCS120, enter a zero as the first character: 0FFFF

Turn off the power to the MMC120 and connect the drive output. Turn the power back on and issue a 'P' command on both axes. Set the MODE to 00000, and the SPEED, ACCELERATION, and DECELERATION to about 20% of the expected maximum speed. To confirm that everything is correct, make some short moves by entering the COMMAND VALUE and issuing a GO ('G') command.

**NOTE:** DCS120 is extremely useful; we recommend you use it to configure, tune, and troubleshoot the system. Refer to Appendix A for information about creating plots of moves.

At this point AUTO STOP should be set to 0E0E0 so any transducer error on an axis will cause it to stop, but other errors will not. Check the STATUS word for errors after each move.



## 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 EXT FEED FORWARD and RET 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 1,000. 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 'alt-F' 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. See Appendix A for more information.

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 which 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 DCS120 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 MAXIMUM ERROR until errors start to occur then adjust the FEED FORWARD gains.

Increase the SPEED and ACCELERATION values gradually while making long moves. Use DCS120 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; if the error occurs on a retract move, increase the RETRACT REED 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 and DECEL 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 MAXIMUM ERROR and adjusting the PROPORTIONAL GAIN until the axis can be moved without getting an error. Remember: the parameters are not updated in the MMC120 until the 'P' 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 'G' and 'H' commands repeatedly. This causes the Motion Control Module to GO and HALT.

## Saving Parameters and Profiles

After the system is set up and tuned using DCS120, you need to store the parameter and profile values in the Programmable Controller. This is done using Command Types A and B. Only the parameters for the axes used need to be saved, and only the profiles used need to be saved.

## MOTION CONTROL PARAMETERS

Sixteen Parameter registers and 6 Command registers control the operation of the MMC120. The status of the module is shown in 10 internal Readback registers. Manipulating the contents of the Parameter and Command registers is key to the successful application of the MMC120.

Delta's DCS120 program (included with the module) gives you direct access to all the internal registers. This is helpful for setup and debugging.

The chapter 'Communicating With The MMC120' describes how the Quantum Programmable Controller gains access to the internal registers of the MMC120. This chapter describes these internal registers.

### Initialization Parameters

#### CONFIGURATION Word (Default: 0000)

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

##### 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 MMC120 continues to count, resulting in about 2700 extra counts per recirculation which are added to the transducer count. These extra counts create an offset; to correct this offset 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 should be slightly positive when the transducer is at its minimum position. If your minimum position count is negative, reduce the setting of these bits until the count is slightly positive.

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 10 - Reverse Drive Mode bit

When this bit is set the polarity of the axis' 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 and the displayed drive will be the opposite of what is output by the module (a positive display will output a negative voltage).

##### 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 this bit is set, two changes are made to the way Open Loop mode is used:

- 1) After completing a move the axis is automatically placed in Open Loop mode with the drive set to null.
- 2) When a "P" command is sent to the axis, it is placed in Open Loop mode with the drive set to null.

**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 which must complete to prevent the machine from stopping in an incomplete state (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 null 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.)

**Bit 14 - Divide by 4**

When this bit is set and bit 15 is cleared, the transducer counts are divided by four before being used by the MMC120. When bits 14 and 15 are both set, the transducer counts are divided by eight.

**Bit 15 - Divide by 2**

When this bit is set and bit 14 is cleared, the transducer counts are divided by two before being used by the MMC120. When bits 14 and 15 are both set, the transducer counts are divided by eight.

**Bit 16 - Integrator Limit**

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

### 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	1	1	1				
Bit Definition																
-----																
Transducer Type/8 Recirc -	01	-	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Transducer Type/4 Recirc -	02	--	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Transducer Type/2 Recirc -	03	---	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Transducer Type/1 Recirc -	04	----	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Reserved -----	05	-----	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Reserved -----	06	-----	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Reserved -----	07	-----	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Reserved -----	08	-----	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Reserved -----	09	-----	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Reverse Drive Mode -----	10	-----	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Absolute Mode -----	11	-----	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Continue Mode -----	12	-----	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Simulate Mode -----	13	-----	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Divide by 4 -----	14	-----	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Divide by 2 -----	15	-----	/	/	/	/	/	/	/	/	/	/	/	/	/	/
Integrator Limit -----	16	-----	/	/	/	/	/	/	/	/	/	/	/	/	/	/

**SCALE (Default: 30300) Range: -32767 to 32767, excluding 0**

See the Startup section (page 20) for additional information.

**TIP:** Use DCS120 commands P0 and P1 to help calculate SCALE and OFFSET.

The SCALE parameter is used to calculate the ACTUAL POSITION in Position Units from the TRANSDUCER COUNTS. Position Units can be 0.001", 0.1 mm, etc. The primary use for the SCALE parameter is to compensate for variations in magnetostrictive transducers. It can also be used to convert to non-standard units for special applications.

SCALE is defined as 32768 times the number of Position Units per TRANSDUCER COUNT:

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

Where P0 and P1 are two positions and C0 and C1 are the corresponding transducer counts.

Magnetostrictive displacement transducers have a calibration number (nominally 9.012 microseconds per inch) and the MMC120 has a 60 MHz oscillator with both edges counted, for an effective counter speed of 120 MHz. To determine the SCALE for a magnetostrictive transducer use the following formula:

$$\text{SCALE} = \frac{\text{Position Units per Inch} \times 32768}{\text{Cal. Number } (\mu\text{s/in}) \times 120 \text{ (MHz)}} \times \text{Sign}$$

Where Sign equals +1 when an extend move yield increasing ACTUAL POSITION counts, and it equals -1 when an extend move yields decreasing ACTUAL POSITION counts. Negative SCALE values indicate the axis moves in reverse. That is, the ACTUAL POSITION decrease as the TRANSDUCER COUNTS increase.

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.

The ACTUAL POSITION is determined as follows:

$$\text{ACTUAL POSITION} = \frac{\text{TRANSDUCER COUNTS} \times \text{SCALE}}{32768} + \text{OFFSET}$$

---

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

---

**SCALE Calculation Examples**

## Example 1

For a system using a magnetostrictive transducer with a calibration number of 9.0110  $\mu\text{s}$  per inch and a position unit of 0.001 inch, the SCALE will be:

$$\text{SCALE} = \frac{1000(\text{Position Units Per Inch}) \times 32768}{9.0110} = 30304$$

$$9.0110(\mu\text{s/in}) \times 120 \text{ (MHz)}$$

**Example 2**

For a system using a magnetostrictive transducer with a calibration number of 9.0110  $\mu\text{s}$  per inch and a position unit of 0.1 millimeter, the SCALE will be:

$$\text{SCALE} = \frac{254(\text{Position Units Per Inch}) \times 32768}{9.0110(\mu\text{s/in}) \times 120 \text{ (MHz)}} = 7697$$

**OFFSET (Default: 0)**

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:

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

The P0 and P1 commands can help calculate the OFFSET parameter. See page 82 for additional information.

**CAUTION:** When using OFFSET, you must be familiar with the limitations of 16 bit math.

**EXTEND LIMIT (Defaults to current position on power-up)**

The EXTEND LIMIT specifies the maximum requested position value that the MMC120 will allow as a COMMAND VALUE. (When the SCALE is negative, this is the minimum value.) A COMMAND VALUE that exceed this value will be set to the EXTEND LIMIT, and the PARAMETER ERROR bit (bit 5) in the STATUS word will be set. The EXTEND LIMIT is given in Position Units. On startup the EXTEND LIMIT defaults to the current position of the axis. New EXTEND and RETRACT LIMITS must be issued with a 'P' command before the axis will move.

---

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

---

**RETRACT LIMIT (Defaults to current position on power-up)**

The RETRACT LIMIT specifies the minimum value the MMC120 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. On startup the RETRACT LIMIT defaults to the current position of the axis. New EXTEND and RETRACT LIMITS must be entered, then a 'P' command sent, before the axis will move.

---

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

---

**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 millivolt per unit of Position Error

Proportional drive is defined as:

Proportional Drive = PROPORTIONAL GAIN x Position Error

### **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.

INTEGRAL GAIN is defined as:

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

Integral Drive is defined as:

Integral Drive = INTEGRAL GAIN x Accumulated Counts

### **DIFFERENTIAL GAIN (Default: 0) Range: 0 to 65535**

The DIFFERENTIAL GAIN field is used to control the differentiator. 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 millivolt per change in Position Error

Differential Drive is defined as:

$$\text{Differential Drive} = (E0_{(\text{TARGET POSITION} - \text{ACTUAL POSITION})} - E1_{(\text{TARGET POSITION} - \text{ACTUAL POSITION})}) \times \text{DIFFERENTIAL GAIN}$$

where E0 is the position error 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.

### **EXTEND FEED FORWARD (Default: 100) Range: 0 to 65535**

FEED FORWARD is an open loop compensation which 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:

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

**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.

### **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 COUNTS.

---



**EXTEND ACCELERATION FEED FORWARD (Default: 0) Range: 0 to 65535**

An axis will not respond to a change in the drive output until some time has passed. This causes the error between the actual and target positions to be larger for axes that are moving large masses or have slow valves. The MMC120 can help compensate for these factors by increasing the drive output. The increase in drive is controlled by the value of this parameter, which has units of millivolts of drive per 100,000 position units per second per second.

If the position unit is .001 inches, this equals millivolts of drive per 100 inches per second per second.

**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)**

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.

**CAUTION:** Do not make this value too large or the drive will oscillate.

**IN POSITION (Default: 50)**

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 STATUS word.

For 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. Note that if the IN POSITION parameter is 0 (zero), the In Position bit will never be set.

**FOLLOWING ERROR (Default: 250)**

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 STATUS word.

**AUTO STOP (Default: 0FFE0 - Soft Stops enabled, Hard Stop enabled for transducer errors)**

Bits 1 through 8 in the AUTO STOP field have a one-to-one correspondence with bits 1 through 8 in the STATUS word. The bits in this byte enable a Soft Stop whenever the corresponding error bit (1 through 8) is set in the STATUS word. During a Soft Stop, the target speed ramps down to zero using the current DECELERATION value. On startup these bits are all set, so any error will cause a Soft Stop. Bits 1, 2, and 3 (the transducer error bits) cannot be cleared. This means that a transducer error will always cause a Soft Stop, and will cause a Hard Stop if bits 9, 10, or 11 are set. If a transducer error causes a Soft Stop then the axis goes into open loop and ramps the drive output to null using the current open loop ramp rate.

---

**NOTE:** To input hexadecimal numbers using DCS120, enter a zero as the first character: 0FFFF

---

Bits 9 through 16 in the AUTO STOP field also have a one-to-one correspondence with bits 1 through 8 in the STATUS word. The bits in this byte enable a Hard Stop whenever the corresponding error bit (1 through 8) is

set in the STATUS word. During a Hard Stop, the drive output is set to zero and the axis placed in open loop mode, and held there until a new command is issued; bits 13 (Open Loop) and 14 (Halt) are also set. On startup the Hard Stop error bits are set to E0 hex, enabling only the transducer error bits.

<u>Soft Stop, Bit #</u>	<u>Error Bit</u>	<u>Hard Stop, Bit #</u>
1	1 - No Transducer	9
2	2 - Transducer Noise	10
3	3 - Transducer Overflow	11
4	4 - Overdrive	12
5	5 - Parameter Error	13
6	6 - Position Overflow	14
7	7 - Integrator Windup	15
8	8 - Following Error	16

If both Soft Stop and Hard Stop bits are set for a particular error condition, a Hard Stop will be executed.

## Motion Control Commands

The motion control command words (MODE, ACCELERATION, DECELERATION, SPEED, COMMAND VALUE, and COMMAND) can be changed while the axis is in motion.

### MODE (Default: 00000)(See page 36 for the MODE Word bit map)

Eight bits in the MODE word determine the way the MMC120 responds to control commands and parameters. Bit 1 is the MSB and bit 16 is the LSB.

#### Bit 1 - Graph Disable

When this bit is set on an axis of the MMC120, 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 MMC120 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 about 1.5 times the straight-line acceleration. 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, then 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 times in milliseconds.

#### Bit 12- Sync

When this bit is set, it causes the axes to move in synchronization. The axes are synchronized as follows:

- A) The axis with the longest move is the master.
- B) The profile of the other axis is determined by the length of its move relative to the master axis. Both 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.
- C) If either axis is halted by an error condition, the other axis will also be halted.

#### Bits 13 and 14 - Integrator Mode Select

These two bits define four integrator modes:

	Bit #
	<u>13</u> <u>14</u>
Mode 0 - Integrator always active -----	0 0
Mode 1 - Integrator active only during DECEL and IN POSITION ----	0 1
Mode 2 - Integrator active only during IN POSITION -----	1 0
Mode 3 - Integrator never active -----	1 1

**Bits 15 and 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

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 to accelerate or decelerate to the specified SPEED.

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

**ACCELERATION (Default: 1000)**

This parameter defines the acceleration ramp rate used by the axis for a move. It has four meanings depending on the setting of the acceleration/deceleration mode.

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 .001", then a value of 200 would represent an acceleration rate of 0.200 inches/sec/sec.

The relationship between the ramp distance, ACCELERATION and SPEED is:

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

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

In Mode 1, the ramp rate is one thousand times larger:

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

In Mode 1, if one Position Unit equals .001", an ACCELERATION of 200 would represent an acceleration rate of 200 inches/sec/sec.

In Mode 2, this parameter defines the distance to the specified SPEED, and in Mode 3, it defines the time to the specified speed.

### 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

Bit Definition		1   2   3   4				5   6   7   8				9   0   1   2				3   4   5   6			
-----		_ _ _				_ _ _				_ _ _				_ _ _			
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 and Decel Mode	15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	/	/
Accel and Decel Mode	16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	/

## DECELERATION (Default: 1000)

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

## SPEED (Default: 1000)

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

---

## COMMAND VALUE

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:

<u>For this command:</u>	<u>The COMMAND VALUE field is:</u>
'A' (Change ACCEL)	Acceleration value
'D' (Change DECEL)	Deceleration value
'E' (Start Events)	Event number
'G' (Go To Position)	Requested Position
'I' (Set Integral Drive)	Integral Drive value
'J' (Go Relative)	Relative Position
'M' (Set Mode)	Mode
'N' (Set Null)	New Null value
'O' (Open Loop Output)	Requested Drive
'U' (Update FLASH)	The section of FLASH memory to update
'V' (Set SPEED)	Speed

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

## COMMAND

The COMMAND word tells the MMC120 what to do. See Appendix A for more information about commands.

## Move Profiles

The module stores 16 user-programmable move profiles in its memory. The following table lists the default values for these profiles. Profiles 0 through 7 can be changed by either a PLC program or DCS120; profiles 8 through 15 can be changed only by DCS120.

Profile Parameter	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 Parameter	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 Parameter	8	9	0Ah	0Bh
MODE	1	1	1	1
ACCEL	100	100	100	200
DECEL	100	100	100	200
SPEED	20000	25000	30000	35000

Profile Parameter	0Ch	0Dh	0Eh	0Fh
MODE	1	1	1	1
ACCEL	200	200	200	100
DECEL	200	200	200	100
SPEED	40000	45000	50000	10000

**MMC120 Initial Move Profiles**

A 'Go' or 'Open Loop' command uses the profile of the last move. If these commands are issued immediately after power up, when there is no 'last move' profile, the module uses the following profiles for these commands:

<b>Profile Parameter</b>	<b>Open Loop</b>	<b>Go</b>
<b>MODE</b>	1	0
<b>ACCEL</b>	100	1000
<b>DECEL</b>	100	1000
<b>SPEED</b>	0	1000



## Readback Parameters

### ACTUAL POSITION

The ACTUAL POSITION is the measured position of the axis at any moment. This position is updated every millisecond. The ACTUAL POSITION is calculated from the TRANSDUCER COUNTS as follows:

$$\text{ACTUAL POSITION} = \frac{\text{TRANSDUCER COUNTS} \times \text{SCALE}}{32768} + \text{OFFSET}$$

### AXIS STATUS Word (See map on page 42)

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. (See the AUTO STOP parameter on page 31.)

Error bits 2 through 8 are cleared whenever a 'G' command is given.

---

**NOTE:** Bit 1 is the MSB, Bit 16 is the LSB.

---

#### 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 (page 31). 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 250 counts, 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. 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.

**Bit 6 - Position Overflow**

This bit is set when the ACTUAL POSITION is greater than its 16 bit limit (above 65535 counts). The ACTUAL POSITION will display 65535, indicating an invalid value. The Position Overflow bit is not latched. It causes no action, a Soft Stop, or a Hard Stop, depending on the setting of AUTO STOP bits 6 and 14. This bit will stay on until a new command is given to the axis.

**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 a valid command or Status Area Request is received by the MMC120 from the Programmable Controller.

**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.

**Bits 11 & 12 - State Bits**

Bits 11 and 12 show the state of the target generator:

<u>STATE</u>	<u>Bit 11 (State Bit B)</u>	<u>Bit 12 (State Bit A)</u>
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' 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 Soft 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.

### 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. See the IN POSITION parameter.

---

**NOTE:** When bit 16 is set, an axis is at the commanded position. This bit can be monitored by the ladder program to determine when an axis has arrived at the commanded position.

---

## 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 a 'Go', 'Go Using Profile', or 'P' command is issued using the COMMAND parameter.

## TARGET POSITION

The TARGET POSITION is the position calculated by the target generator, using the ACCEL, DECEL, SPEED, COMMAND VALUE, and MODE parameters, showing where the axis should be at any moment.

When the axis is moving, the TARGET POSITION is updated each millisecond as it moves toward the COMMAND POSITION. The value can be read by the Quantum Controller on every scan.

---

**NOTE:** When an axis is stopped, the TARGET POSITION should be the same as the COMMAND POSITION unless an error or HALT has occurred.

---

## TRANSDUCER COUNTS

TRANSDUCER COUNTS is the axis position read directly from the transducer counters with no scaling. TRANSDUCER COUNTS can overflow the 16-bit value; status bit 6 is set when this happens.

## 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 millisecond.

## 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  $\pm 10$  volt output in steps of 0.005 volts. Note that there are no pots to electronically null the output; this is done in software. Null drive will be near 0 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.

### 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.

Hexadecimal To	F	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1
Binary Conversion	E	1 1 1 0	1 1 1 0	1 1 1 0	1 1 1 0
Table	D	1 1 0 1	1 1 0 1	1 1 0 1	1 1 0 1
	C	1 1 0 0	1 1 0 0	1 1 0 0	1 1 0 0
	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

Bit Definition																			
-----																			
**No Transducer	01	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
**Transducer Noise	02	--	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
**Transducer 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.

## FLASH MEMORY

The MMC120 contains FLASH memory that has two user-accessible sections. Section 1 contains the parameters and profiles for both axes. Section 2 contains the event step table for both axes. The two sections have separate checksums; the checksums are tested on module power-up and on reset.

If the parameters initialize bit is off, check the parameter error bits, take the action described below to correct the problem, then issue a 'P' command. Even if the parameter error bits are not set, you still must issue a 'P' command to an axis before using it.

If the checksum for Section 1 is incorrect, the axis 1 parameter error bit will be turned on. If the checksum for Section 2 is incorrect, the axis 2 parameter error bit will be turned on. If either of these bits is set, you must update all the data for both axes in that section. For example, if the axis 2 parameter error bit is on, you must update the event step table information for *both* axes.

To update the FLASH information, you must first download correct data to the MMC120 volatile memory. Use the SET PARAMETERS command (type 3) to download the parameters, the SET PROFILES command (type 2) to download the profiles, and the EVENT STEP TRANSFER command (type Extended E) to download the event data. (See page 47 for command types.)

After the correct data has been downloaded to the MMC120's volatile memory, transfer it to the FLASH memory with the 'U' command (55h). To transfer data to the parameter section (1), put a '1' in the COMMAND VALUE field then issue a 'U' command. To transfer data to the event step table section (2), put a '2' in the COMMAND VALUE field then issue a 'U' command. The 'U' command uses the Acknowledge bit (9) of the Status word to report the result of the update. If the FLASH updated successfully, bit 9 of axis 1 is toggled; if the update was unsuccessful, bit 9 of axis 2 is toggled.

The disadvantage to storing configuration data in the MMC120 is that when one module is replaced with another, the stored data must be loaded into the new module. Because of this, all parameters, profiles, and event step table data must be stored either in the ladder logic program or in a DCS120 file so they can be transferred to an MMC120 when needed.

## COMMUNICATING WITH THE MMC120

### Quantum Bus Configuration

Select four 3XXXXX registers for inputs and four 4XXXXX registers for outputs of each module.

The MMC120 must be configured as a `DCS MMC 120 0x` module in the I/O map.

---

**NOTE:** You must confirm that the following line is present in the `\MODSOFT\RUNTIME\GCNFTCOP.SYS` file. If this line is not present it must be added; if it is present but does not EXACTLY match this line (except for the sequence number) it must be changed:

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

The sequence number (156 in the example above) must not be repeated. It must be set to the next value available in your file.

---

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 steps storage. If you also want to store the parameters in the Programmable Controller's memory you must reserve 32 words of memory (registers).

<p><b>TIP:</b> Use DCS120 software to set up and modify the parameters, profiles, and event steps, then move them to the Programmable Controller memory.</p>
--

Each axis requires 16 words of memory in the Quantum Programmable Controller. This memory contains the initialization parameters for the axis. The parameters are arranged in the following order:

- 0 CONFIGURATION
- 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 registers and each module can store up to eight profiles in the PLC's memory, for a total of 32 registers. These first eight profiles can be changed from the PLC and another eight, for a total of 16, can be changed from DCS120. 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 in the PLC, you must reserve eight registers for each step used. If you use all 256 event steps, you must reserve 2048 registers.

The demo program on the DCS120 disk shows an example of ladder logic for transferring parameters, profiles, and Event Control steps to and from the MMC120.

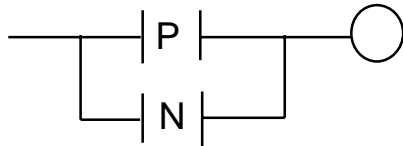
## I/O Register Overview

The MMC120 communicates with the Quantum Programmable Controller through four input (to the programmable controller) and four output (to the MMC120) registers.

Input Registers		Output Registers	
3TTTTT + 0	Axis 1 Status Bits	4TTTTT + 0	Axis 1 SAR/Command
3TTTTT + 1	Axis 1 Data In	4TTTTT + 1	Axis 1 Data Out
3TTTTT + 2	Axis 2 Status Bits	4TTTTT + 2	Axis 2 SAR/Command
3TTTTT + 3	Axis 2 Data In	4TTTTT + 3	Axis 2 Data Out

See page 46 for Output Register information and page 62 for Input Register information

Whenever you issue a command to a module that changes the status area's definition, you must wait for the ACK bit to toggle before reading the input registers. This guarantees that the data in the input registers is valid. The following ladder logic will detect whenever the ACK bit changes state.



The MMC120 can accept commands as fast as the P/C can send them, but there could be as much as 2 ms of latency before the MMC120 toggles the ACK.

I/O scan time must not be faster than segment scan. To keep data synchronized in a single-segment program, round the minimum scan time up to the next 10ms boundary. For example, if your program has a 5 ms scan time, set the minimum scan time to 10 ms; if it is 12ms, set the minimum to 20 ms.



## Output Registers

Four 16-bit registers are sent to the MMC120 each time the I/O drop is accessed. The registers contain commands and data:

<u>Register Number</u>	<u>Contents</u>
4TTTTT + 0	Command - Axis 1
4TTTTT + 1	Data Out - Axis 1
4TTTTT + 2	Command - Axis 2
4TTTTT + 3	Data Out - Axis 2

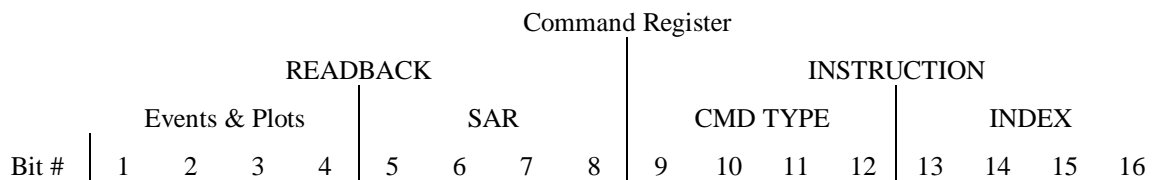
The commands sent to the two axes are independent of each other.

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.

## COMMANDS

Command registers are divided into two bytes. The most significant byte (bits 1 - 8) specifies the Readback mode; it is called the READBACK byte. The least significant byte (bits 9 - 16) specifies the instruction the MMC120 is to follow; it is called the INSTRUCTION byte.

The two bytes in the command register are divided into 4-bit nibbles (a "nibble" is half a byte). In the INSTRUCTION byte the most significant nibble (bits 9 - 12) specifies the Command Type; the least significant nibble (bits 13 - 16) specifies an index indicating what the command acts on. In the Readback byte, bits 5 - 8 are the Status Area Request (SAR) field indicating what status information the MMC120 is to return, and bits 1 - 4 are used by plots and Event Step commands. (See the Input Register section.)



## Command Types

The Command Types available for controlling the MMC120 are:

Command Type	Binary Representation				Description
0	0000	XXXX	0000	NNNN	Reserved
1	0000	XXXX	0001	NNNN	Go using profile
2	0000	XXXX	0010	NNNN	Set profiles
3	0000	XXXX	0011	NNNN	Set parameters
4	0000	XXXX	0100	NNNN	ASCII commands
5	0000	XXXX	0101	NNNN	ASCII commands
6	0000	XXXX	0110	NNNN	ASCII commands
7	0000	XXXX	0111	NNNN	ASCII commands
8	0000	XXXX	1000	NNNN	Reserved
9	0000	XXXX	1001	NNNN	Open Loop using profile
A	0000	XXXX	1010	NNNN	Get Profiles
B	0000	XXXX	1011	NNNN	Get Parameters
C	0000	XXXX	1100	NNNN	Reserved
D	0000	XXXX	1101	NNNN	Reserved
E	0000	XXXX	1110	NNNN	Event Step Edit
F	0000	XXXX	1111	NNNN	Diagnostics
Extended E	1110	NNNN	NNNN	NNNN	Event Step Transfer

In the following discussion, an **X** represents a bit whose value does *not* affect the meaning of the rest of the bits in a word. An **N** indicates a bit that can be either 1 or 0 and whose value *does* affect the meaning of the word.

## Command Type 1 - Go Using Profile (0000 XXXX 0001 NNNN)

The Go Using Profile command allows the Quantum Programmable Controller to tell the MMC120 to move the axis using stored profiles. These are the most commonly used commands. The Go Using Profile command toggles the Acknowledge bit and clears the In Position bit in the STATUS word.

Each stored profile contains MODE, ACCELERATION, DECELERATION, and SPEED parameters, which define the profile of a move.

The format of the Command Register is:

HEX VALUE	BIT #					
	1234	5678	9012	3456		
	SAR	CMND	INDX			
XX10	0000	XXXX	0001	0000	- GO USING PROFILE 0	
XX11	0000	XXXX	0001	0001	- GO USING PROFILE 1	
XX12	0000	XXXX	0001	0010	- GO USING PROFILE 2	
XX13	0000	XXXX	0001	0011	- GO USING PROFILE 3	
XX14	0000	XXXX	0001	0100	- GO USING PROFILE 4	
XX15	0000	XXXX	0001	0101	- GO USING PROFILE 5	
XX16	0000	XXXX	0001	0110	- GO USING PROFILE 6	
XX17	0000	XXXX	0001	0111	- GO USING PROFILE 7	
XX18	0000	XXXX	0001	1000	- GO USING PROFILE 8	Predefined profiles - modify from DCS120 only
XX19	0000	XXXX	0001	1001	- GO USING PROFILE 9	
XX1A	0000	XXXX	0001	1010	- GO USING PROFILE 10	
XX1B	0000	XXXX	0001	1011	- GO USING PROFILE 11	
XX1C	0000	XXXX	0001	1100	- GO USING PROFILE 12	
XX1D	0000	XXXX	0001	1101	- GO USING PROFILE 13	
XX1E	0000	XXXX	0001	1110	- GO USING PROFILE 14	
XX1F	0000	XXXX	0001	1111	- GO USING PROFILE 15	

The output registers are defined as follows:

4TTTTT+0	Axis 1 Command - Go using profile
4TTTTT+1	Axis 1 Requested Position
4TTTTT+2	Axis 2 Command - Go using profile
4TTTTT+3	Axis 2 Requested Position

**Command Type 2 - Set Profiles (0000 XXXX 0010 NNNN)**

These commands allow the programmer to change motion profiles stored in the MMC120. Only one value in one profile can be changed each time the drop is accessed, 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. New GO commands can also be given while the axis is moving.

HEX VALUE	BIT #				4TTTTT+0 Axis 1	4TTTTT+2 Axis 2	
	SAR	CMND	INDX				
0X20	0000	XXXX	0010	0000	0	4	MODE
0X21	0000	XXXX	0010	0001	0	4	ACCEL
0X22	0000	XXXX	0010	0010	0	4	DECEL
0X23	0000	XXXX	0010	0011	0	4	SPEED
0X24	0000	XXXX	0010	0100	1	5	MODE
0X25	0000	XXXX	0010	0101	1	5	ACCEL
0X26	0000	XXXX	0010	0110	1	5	DECEL
0X27	0000	XXXX	0010	0111	1	5	SPEED
0X28	0000	XXXX	0010	1000	2	6	MODE
0X29	0000	XXXX	0010	1001	2	6	ACCEL
0X2A	0000	XXXX	0010	1010	2	6	DECEL
0X2B	0000	XXXX	0010	1011	2	6	SPEED
0X2C	0000	XXXX	0010	1100	3	7	MODE
0X2D	0000	XXXX	0010	1101	3	7	ACCEL
0X2E	0000	XXXX	0010	1110	3	7	DECEL
0X2F	0000	XXXX	0010	1111	3	7	SPEED

Setting one complete profile requires four scans with one command written each scan.

Example: Set profiles 2 and 7

Commands to axis 1 set profile 2 while commands to axis 2 set profile 7, so these two profiles can be set simultaneously.

```

First Scan    4TTTTT + 0  0X28h (Set profile 2 MODE)
                + 1  0001h (MODE for profile 2)
                + 2  0X2Ch (Set profile 7 MODE)
                + 3  0001h (MODE for profile 7)

Second Scan   4TTTTT + 0  0X29h (Set profile 2 ACCEL)
                + 1   100 (ACCEL for profile 2)
                + 2  0X2Dh (Set profile 7 ACCEL)
                + 3   150 (ACCEL for profile 7)

Third Scan    4TTTTT + 0  0X2Ah (Set profile 2 DECEL)
                + 1    70 (DECEL for profile 2)
                + 2  0X2Eh (Set profile 7 DECEL)
                + 3    70 (DECEL for profile 7)

```

```
Fourth Scan  4TTTTT  + 0    0X2Bh (Set profile 2 SPEED)
              + 1    12000 (SPEED for profile 2)
              + 2    0X2Bh (Set profile 7 SPEED)
              + 3    20000 (SPEED for profile 7)
```

These four scans result in:

```
Profile 2 -  MODE = 0001h
              ACCEL = 100
              DECEL = 70
              SPEED = 12000
```

```
Profile 7 -  MODE = 0001h
              ACCEL = 150
              DECEL = 70
              SPEED = 20000
```

**Command Type 3 - Set Parameters (0000 XXXX 0011 NNNN)**

These commands allow the Quantum Programmable Controller to download new initialization parameters to the MMC120. New parameters can be downloaded at any time, but they do not take effect until a 'P' command is issued (see Command Type 5). An axis must be stopped when issuing a 'P' command.

Set Parameter commands are defined as follows:

		BIT #				
		1234	5678	9012	3456	
HEX	VALUE	SAR	CMND	INDX		
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 ACCELERATION FEED FORWARD
0X3B	0000	XXXX	0011	1011		- Set RETRACT ACCELERATION 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 output registers are defined as follows:

4TTTTT + 0	Axis 1 command - set parameter
4TTTTT + 1	Axis 1 parameter value
4TTTTT + 2	Axis 2 command - set parameter
4TTTTT + 3	Axis 2 parameter value

Example: Set the extend limit for both axes

4TTTTT + 0	0X33	Set Axis 1 Extend Limit
4TTTTT + 1	24000	Extend Limit value
4TTTTT + 2	0X33	Set Axis 2 Extend Limit
4TTTTT + 3	12000	Extend Limit value

These commands set the Extend Limit for Axis 1 to 24000 and Axis 2 to 12000.

**Command Types 4, 5, 6, and 7 - ASCII Commands (0000 XXXX 01NN NNNN)**

These commands usually don't require parameters. The 'E', 'I', 'G', 'J', 'M', 'N', 'O', 'U', and 'V' commands are exceptions. The MMC120 acknowledges these commands by setting the Acknowledge bit in the STATUS word until the next time the drop is accessed.

		BIT #					
		0000	0000	0111	1111		
		1234	5678	9012	3456		
		-----					
HEX		SAR	CMND	INDX			
VALUE		-----					
0X41	0000	XXXX	0100	0001	- A	Change ACCELERATION	
0X44	0000	XXXX	0100	0100	- D	Change DECELERATION	
0x45	0000	XXXX	0100	0101	- E	Start event	
0X46	0000	XXXX	0100	0110	- F	FEED FORWARD Adjust	
0X47	0000	XXXX	0100	0111	- G	Go (using last profile given)	
0X48	0000	XXXX	0100	1000	- H	HALT Command	
0X49	0000	XXXX	0100	1001	- I	Set Integral Drive	
0X4A	0000	XXXX	0100	1010	- J	Relative move	
0X4B	0000	XXXX	0100	1011	- K	Disable drive output	
0X4D	0000	XXXX	0100	1101	- M	Set MODE	
0X4E	0000	XXXX	0100	1110	- N	Set NULL	
0X4F	0000	XXXX	0100	1111	- O	OPEN LOOP using last profile	
0X50	0000	XXXX	0101	0000	- P	Initialize Parameters	
0x51	0000	XXXX	0101	0001	- Q	Quit events	
0X52	0000	XXXX	0101	0010	- R	Restore NULL	
0X53	0000	XXXX	0101	0011	- S	Save NULL	
0X55	0000	XXXX	0101	0101	- U	Update FLASH	
0X56	0000	XXXX	0101	0110	- V	Set SPEED	
0X69	0000	XXXX	0110	1001	- i	Set Integral Drive to null drive value	
0x6E	0000	XXXX	0110	1110	- n	Set NULL to Integral Drive value	
0x72	0000	XXXX	0111	0010	- r	Restore Integral Drive	
0x73	0000	XXXX	0111	0011	- s	Save Integral Drive	
0x79	0000	XXXX	0111	1001	- y	Start a graph	

NOTE: Any hex values not listed are reserved

Example:

```
4TTTTT + 0   0X47
4TTTTT + 1   12000
```

Axis 1 moves to 12000 using last profile given.

**NOTE:** When issued by the PLC, these commands must be entered in Hex or Binary. The ASCII representation is used by the DCS120 software package.

---

**'A' - Change ACCELERATION (41h)**

This commands sets the ACCELERATION parameter to the value in the data register of the PLC or the COMMAND VALUE field in DCS120.

**'D' - Change DECELERATION (44h)**

This commands sets the DECELERATION parameter to the value in the data register of the PLC or the COMMAND VALUE field in DCS120.

**'E' - Start Events (45h)**

This commands starts the Events sequence at the step specified in the data register of the PLC or the COMMAND VALUE field in DCS120.

**'F' - Set Feed Forward (46h)**

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 will allow the system to adjust for changing system dynamics. This will also make setup easier.

---

**NOTE:** The 'F' command should be used **only** after the axis is moving smoothly.

---

**'G' - Go to COMMAND VALUE (47h)**

When the Quantum Programmable Controller issues an ASCII 'G' command with a requested position specified in the data register of the PLC or the COMMAND VALUE field in DCS120, the MMC120 executes the move with the last profile used. When no previous Go Using Profile commands have been executed, the last profile used defaults to profile 0. In general, the PLC should use the Go Using Profile commands rather than the ASCII GO command.

When the GO command is issued from DCS120 the profile information (MODE, ACCEL, DECEL, and SPEED) are taken from the DCS120 screen (the COMMAND VALUE contains the requested position). New GO commands with different MODE, ACCEL, DECEL, SPEED, or positions can be issued while the axis is in motion.

**'H' - HALT (48h)**

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.

---

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

---

**'I' - Integral Drive Set (49h)**

This command sets the Integral Drive to the value in the COMMAND VALUE field.

**'i' - Integral Drive Clear (69h)**

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

**'J' - Relative Move (4Ah)**

This command changes the COMMAND POSITION by the amount specified in the data register of the PLC or the COMMAND VALUE field in DCS120.



**'K' - Disable Drive Output (4Bh)**

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 until a new command is issued. When a 'K' command is issued from DCS120 it is issued to all axes simultaneously. When issued from the Quantum PLC, it goes only to the specified axis.

**'M' - Set Mode (4Dh)**

This commands sets the Mode to the value in the data register of the PLC or the COMMAND VALUE field in DCS120.

**'N' - Set Null (4Eh)**

The 'N' command sets the Null Drive to the value in the data register of the PLC or the COMMAND VALUE field in DCS120.

**'n' - Null Drive Clear (6E)**

The 'n' command sets the Null Drive to the current Integral Drive value.

**'O' - Open Loop (4Fh)**

<b>CAUTION:</b> Use this command with care! Open Loop operation disables all safety features on the MMC120!
---

The Open Loop command allows the Quantum 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 amount of drive to be added to (or subtracted from) the current null value.

The ACCELERATION and DECELERATION fields control the rate at which the drive output ramps to the requested value. The ACCEL and DECEL parameters are in millivolts per millisecond.

To command the MMC120 to output 10% of negative drive you would enter -1000 in the COMMAND VALUE field and 79 (O) in the COMMAND field. A '25' in the ACCELERATION field would cause the module to ramp the output at 25 millivolts every millisecond, so the output would ramp to -1000 in 40 milliseconds.

See also Command Type 9.

**'P' - Set Parameters (50h)**

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 MMC120 will copy the ACTUAL POSITION of the axis into the TARGET and COMMAND POSITIONS.

**'Q' - Quit Events (51h)**

This command stops the event control sequence.

**'R' - Restore Null (52h)**

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

**'r' - Restore Integral Drive (72h)**

This command restores a previously saved Integral Drive value.

**'S' - Save Null (53h)**

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

**'s' - Save Integral Drive (73h)**

This command saves the current value of the Integral Drive so it can be recalled later with a Restore Integral Drive command ('r').

**'U' - Update FLASH (55h)**

This command instructs the MMC120 to write all parameters and profiles to FLASH for storage in case of power loss or reset. Issuing 'U' with a 1 in the COMMAND VALUE field updates the Parameters and Profiles, while 'U' with a 2 in the COMMAND VALUE field updates the Event Step Table. Do not use a COMMAND VALUE greater than 2 with the 'U' command.

The 'U' command uses the Acknowledge bit (9) of the Status word to report the result of the update. If the FLASH updated successfully, bit 9 of axis 1 is toggled; if the update was unsuccessful, bit 9 of axis 2 is toggled.

**'V' - Set SPEED (56h)**

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

## Command Type 9 - Open Loop Using Profile (0000 XXXX 1001 NNNN)

The Open Loop Using Profile command allows 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 selected 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.

---

The information stored in the profile table is used as follows:

**MODE** -- Only mode bit 1 (Graph Disable) is valid. All other bits are ignored.

**ACCELERATION** -- The rate of change of the drive output is controlled with this parameter. It is specified in millivolts of drive per millisecond. It applies while the drive is ramping away from null.

**DECELERATION** -- This parameter is the same as the acceleration, except it applies while the drive is ramping toward null.

**SPEED** -- The speed parameter is used as a drive limit. A value of 1000 will let the drive ramp up or down to plus or minus 1000 (1 volt) from null regardless of the value in the COMMAND VALUE field.

The Open Loop command sets the state bits in the Status word (bits 11 and 12) to indicate whether the drive is ramping up, ramping down, or at constant speed (other than null).

The format of the Command Register is given below:

HEX VALUE	BIT #				
	1234	5678	9012	3456	
	SAR	CMND	INDX		
0X90	0000	XXXX	1001	0000	- OPEN LOOP USING PROFILE 0 \
0X91	0000	XXXX	1001	0001	- OPEN LOOP USING PROFILE 1 \ Modify by
0X92	0000	XXXX	1001	0010	- OPEN LOOP USING PROFILE 2 / writing to
0X93	0000	XXXX	1001	0011	- OPEN LOOP USING PROFILE 3 / Axis 1
0X94	0000	XXXX	1001	0100	- OPEN LOOP USING PROFILE 4 \
0X95	0000	XXXX	1001	0101	- OPEN LOOP USING PROFILE 5 \ Modify by
0X96	0000	XXXX	1001	0110	- OPEN LOOP USING PROFILE 6 / writing to
0X97	0000	XXXX	1001	0111	- OPEN LOOP USING PROFILE 7 / Axis 2
0X98	0000	XXXX	1001	1000	- OPEN LOOP USING PROFILE 8 Predefined
0X99	0000	XXXX	1001	1001	- OPEN LOOP USING PROFILE 9 profiles -
0X9A	0000	XXXX	1001	1010	- OPEN LOOP USING PROFILE 10 changed only
0X9B	0000	XXXX	1001	1011	- OPEN LOOP USING PROFILE 11 from DCS120
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 4 output registers are defined as follows:

```

4TTTTT+0  Axis 1 Command - Open Loop using profile
4TTTTT+1  Axis 1 Requested Drive

4TTTTT+2  Axis 2 Command - Open Loop using profile
4TTTTT+3  Axis 2 Requested Drive
    
```

### Command Type A - Get Profiles (0000 XXXX 1010 NNNN)

These commands allow the Programmable Controller to retrieve from the MMC120 profile data that was set up with DCS120.

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

Retrieving one complete profile requires four scans with one command retrieved each scan.

```

Example: Retrieve Profile 5 from the MMC120 -  MODE = 0002h
                                           ACCEL = 50
                                           DECEL = 35
                                           SPEED = 8000
    
```

```

First Scan  4TTTTT + 0  xxxx
              + 1  xxxx (Don't care - Axis 1 parameters)
              + 2  00A4h (Get profile 5 MODE)
              + 3  xxxx (Don't care)
    
```

## Second Scan:

Returns: 3TTTTT + 0 xxxx (Axis 1 status)  
 + 1 xxxx  
 + 2 xxxx (Axis 2 Status)  
 + 3 0002d (Profile 5 MODE)

Sends: 4TTTTT + 0 xxxx  
 + 1 xxxx  
 + 2 00A5h (Get Profile 5 ACCEL)  
 + 3 xxxx

## Third Scan:

Returns: 3TTTTT + 0 xxxx (Axis 1 status)  
 + 1 xxxx  
 + 2 xxxx (Axis 2 status)  
 + 3 0050d (Profile 5 ACCEL)

Sends: 4TTTTT + 0 xxxx  
 + 1 xxxx  
 + 2 00A6h (Get Profile 5 DECEL)  
 + 3 xxxx

## Fourth Scan:

Returns: 3TTTTT + 0 xxxx (Axis 1 status)  
 + 1 xxxx  
 + 2 xxxx (Axis 2 status)  
 + 3 0035d (Profile 5 DECEL)

Sends: 4TTTTT + 0 xxxx  
 + 1 xxxx  
 + 2 00A7h (Get Profile 5 SPEED)  
 + 3 xxxx

## Fifth Scan:

Returns: 3TTTTT + 0 xxxx (Axis 1 status)  
 + 1 xxxx  
 + 2 xxxx (Axis 2 status)  
 + 3 8000d (Profile 5 SPEED)

When this command is used, the MMC120 ignores the SAR field and the parameter register.

**Command Type B - Get Parameters (0000 XXXX 1011 NNNN)**

These commands allow the Programmable Controller to read the current state of the MMC120 initialization parameters. These commands are similar to the Type A commands. The requested parameter will be returned in the corresponding axis status area, and it will be available the next scan after the ACK bit toggles. It will take 17 scans to get all 16 parameters.

		BIT #				
		1234	5678	9012	3456	
HEX	VALUE	SAR	CMND	INDX		
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 ACCELERATION FEED FORWARD
0XBB	0000	XXXX	1011	1011	-	Get RETRACT ACCELERATION 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 output registers are defined as follows:

4TTTTT + 0	Axis 1 command - Get parameter
4TTTTT + 1	Don't care
4TTTTT + 2	Axis 2 command - Get parameter
4TTTTT + 3	Don't care

When using this command the SAR field and the data register are ignored by the MMC120.

## Command Type E - Event Step Edit (0000 XXXX 1110 00NN)

The "Event Step Edit" commands let you change an event parameter's value across a range of steps in a single scan.

```
E0 = 0000|XXXX|1110|0000    Beginning Step Number (0 to 255)
E1 = 0000|XXXX|1110|0001    Ending Step Number (0 to 255; > E0)
E2 = 0000|XXXX|1110|0010    Parameter to Modify (0 to 7)
E3 = 0000|XXXX|1110|0011    Value to be Used
```

```
E2 Parameter Values
0    MODE
1    ACCEL
2    DECEL
3    SPEED
4    POSITION
5    COMMAND
6    LINK TYPE/NEXT
7    LINK VALUE
```

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 an E3 command with the new parameter value.

For 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 using the E007 extended command type. 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.

## Command Type F - Diagnostics (0000 XXXX 1111 XXXX)

The "Diagnostic" commands are for testing the MMC120 under safe conditions. The command echoes the output registers in the input registers. This tests the Quantum-to-MMC120 communications.

```
4TTTTT+0 = 0000|XXXX|1111|XXXX (XXFXh)  3TTTTT+0 = Echo 4TTTTT+0
4TTTTT+1 =      don't care                3TTTTT+1 = Echo 4TTTTT+1
4TTTTT+2 =      "      "                  3TTTTT+2 = Echo 4TTTTT+2
4TTTTT+3 =      "      "                  3TTTTT+3 = Echo 4TTTTT+3
```

When using this command the SAR field is ignored by the MMC120.

## Extended Command Type E - Event Step Transfer (1110 DSSS SSSS SNNN)

The "Event Step Transfer" commands are used to move event step information to and from the MMC120. The 'D' bit indicates the direction of data transfer (0 = Put Step, 1 = Get Step), the 'S' bits specify the step number, and the 'N' bits specify the field of the step to be transferred. Commands E000 to E7FF put (write) data to the MMC120, while commands E800 to EFFF get (read) data from the MMC120. While writing steps, the new command value is echoed to the SAR.

To write event step information to the MM120 :

Cmd Reg 1	Parameter Register 1
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 and LINK NEXT
E007	Step 0 LINK VALUE
E008	Step 1 MODE
E009	Step 1 ACCELERATION
.	.
.	.
E7FD	Step 255 COMMAND
E7FE	Step 255 LINK TYPE and LINK NEXT
E7FF	Step 255 LINK VALUE

To update two words every PLC scan, you can also use Command and Data Parameter Register 2 for axis 2. By using Register 1 for the first half of a table and Register 2 for the second half, transfer time can be cut in half.

To read event step information from the MMC120:

Cmd Reg 1	Status Word 1
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 and LINK NEXT
E807	Step 0 LINK VALUE
E808	Step 1 MODE
E809	Step 1 ACCELERATION
.	.
.	.
EFFD	Step 255 COMMAND
EFFE	Step 255 LINK TYPE and LINK NEXT
EFFF	Step 255 LINK VALUE

The data will be valid when the ACK bit in the Status word toggles (on the next scan).



## Input Registers

Status Information can be selectively read by the Quantum Programmable Controller by using the Status Area Request (SAR) field of the command words. The status word returned per axis is specified in the SAR field of the command for the axis.

Offset	AXIS	REGISTER DESCRIPTION
3TTTTT+1	1	specified by Command 1, 3TTTTT + 0 Axis 1 status bits
3TTTTT+3	2	specified by Command 2, 3TTTTT + 2 Axis 2 status bits

The register indexes are as follows:

0	0000	Command Position
1	0001	Target Position
2	0010	Actual Position
3	0011	Transducer Counts
4	0100	Status
5	0101	Drive Output
6	0110	Actual Speed
7	0111	Drive Null
8	1000	Step
9	1001	Link Value
10 - 15		Reserved

To get the STATUS and ACTUAL POSITION, you would set SAR to 2 (0010). In the example below, the registers are selected by SAR.

	SAR	CMND	INDX	(HEX)	
4TTTT+0	0000	0010	0XXX	XXXX	(02XX) 3TTTT+0 AXIS 1 STATUS
+1	xxxx	xxxx	xxxx	xxxx	3TTTT+1 AXIS 1 ACTUAL POSITION
4TTTT+2	0000	0101	0XXX	XXXX	(05XX) 3TTTT+2 AXIS 2 STATUS
+3	xxxx	xxxx	xxxx	xxxx	3TTTT+3 AXIS 2 DRIVE OUTPUT

The STATUS is always reflected into 3TTTTT + 0 and 3TTTTT + 2. You can monitor the ACK bit in the status word (bit 9) to confirm that a command has been received.

## QUANTUM PROGRAMMING HINTS

The Programmable Controller is responsible for storing the initialization parameters used by the MMC120 and initializing the Motion Control Module with those parameters.

The MMC120 provides a STATUS word for each axis. If an error bit is set 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 MMC120 only once per segment, otherwise the newest data will overwrite the previous data.

See the demo program on the DCS120 disk for ladder logic examples.

### Error Handling

The MMC120 reports errors to the Quantum Controller within one millisecond of detection. Errors are reported by setting bits in the affected axis' STATUS word and turning on the appropriate Front Panel 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 which 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 MMC120 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 'H' (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 for error detection by setting the appropriate bits in the SOFT STOP the HARD STOP bytes corresponding to the error bits in the STATUS word. See AUTO STOP (page 31) and STATUS (pages 38-40).

## TROUBLESHOOTING

### Problems and Solutions

#### Ladder 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' 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  
Stopped

#### Transducer counts field not indicating transducer location

See 'Axis indicator constant red' above.

#### Transducer counts field changes but output drive does not work

See '...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 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 which 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 PLC issuing unintended commands to the module. Use the DCS120 command queue to monitor commands sent by the PLC to the module. Confirm that only the expected commands are being sent. See **Appendix A: DCS120 Diagnostic and Setup Program** for more information about the command queue.

## HYDRAULIC SYSTEM PROBLEMS

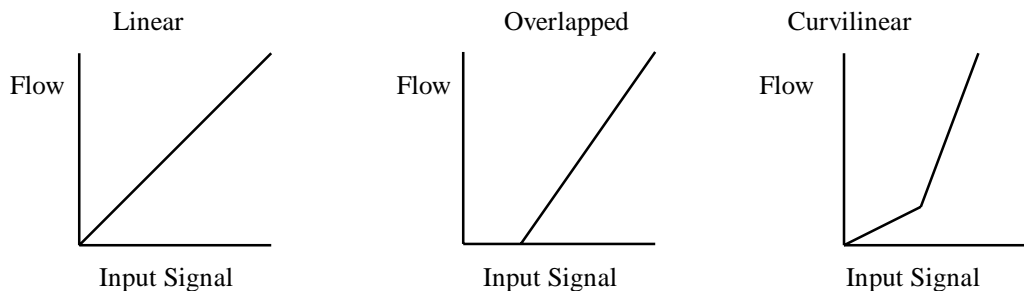
These hydraulic system problems can make system tuning difficult or impossible.

### 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 nonlinearity:

**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

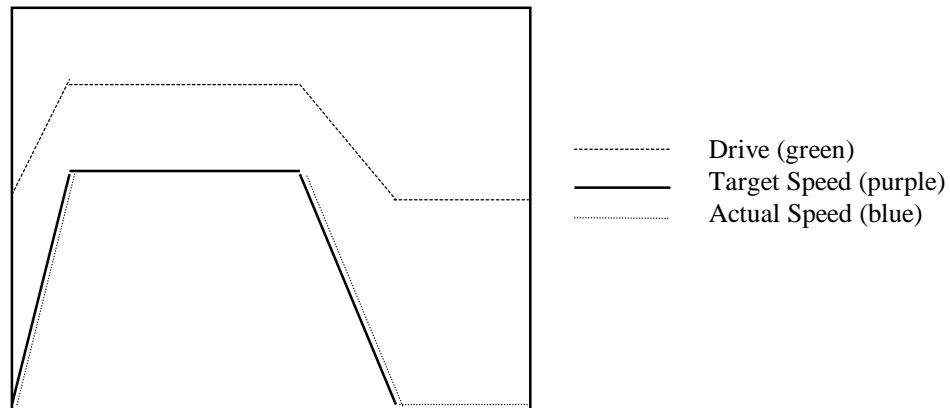
Insufficient pump and/or accumulator capacity will cause the system response to degrade during a move because the effective pressure drops.

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.

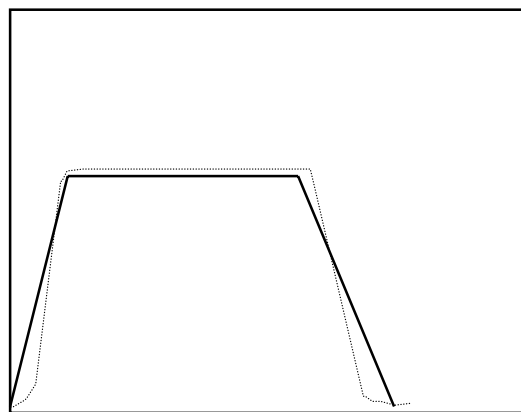
Insufficient pump capacity can result in inadequate control when moving many axes simultaneously or when making long moves. In these cases pressure can drop so much a fully open valve cannot maintain the requested speed.

## 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 with DCS120. The graph (ignoring the Position terms) should show:

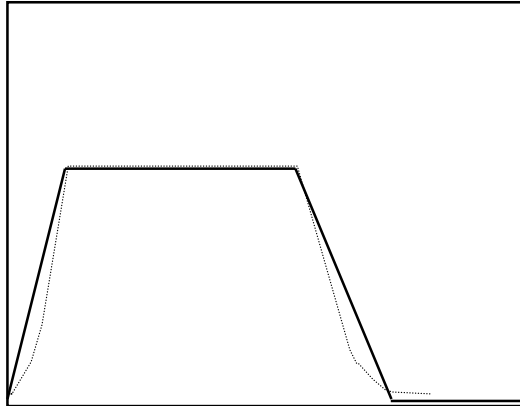


If the actual speed and target speed show:



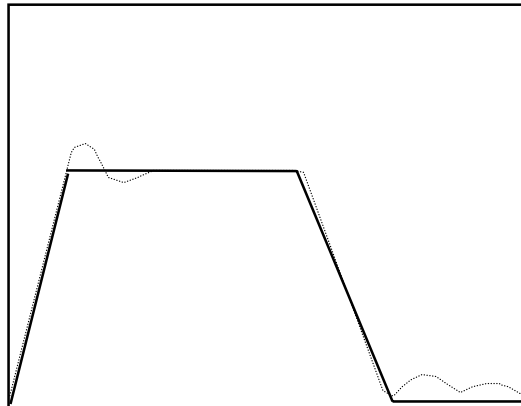
Your valve probably has overlap. Replace the valve with a linear one or try increasing the DEAD BAND ELIMINATOR value.

If the speeds show:



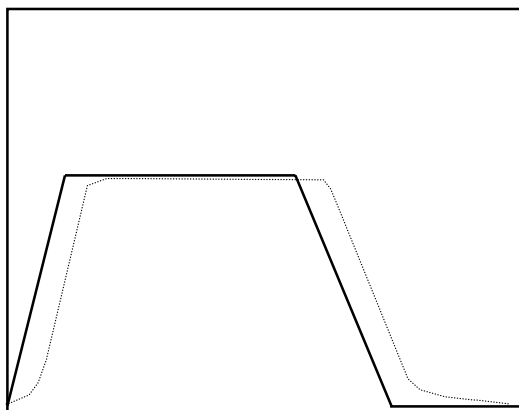
Your valve is probably curvilinear. Replace the valve with a linear one or increase the proportional gain and tune the system for high-speed stability; expect poor control at low speed and when stopped.

If the speeds show:



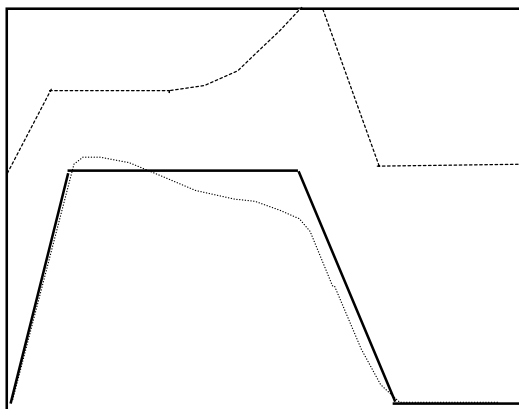
You may have too much hose between the valve and the cylinder. Reduce the amount of hose or add differential gain (usually less than 5).

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.



## REPAIRS AND RETURNS

### Warranty

The MMC120 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 the MMC120 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

If you need to return an MMC120 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.**  
**11719 NE 95th St. , Suite D**  
**Vancouver, Washington 98682-2444**

**Phone: 360-254-8688**  
**Fax: 360-254-5435**  
**Pager: 360-699-7784 (24 hr)**

### Returns

Returned items (in new condition and less than 120 days from shipment from Delta) must be approved by Delta and are subject to a 25% restocking fee. An RMA number must be issued before an item is returned.

### Delta Web Page and Email

Delta also has a site on the World Wide Web. To get product information and application notes, point your browser to:

**<http://www.deltacompsys.com>**

And you can send us email at:

**[email@deltacompsys.com](mailto:email@deltacompsys.com)**

## GLOSSARY OF TERMS

**Actual Position** - The scaled measured axis position expressed in Position Units.

**Circulation** - The process of sending an interrogation signal to a transducer and counting a high frequency clock while waiting for the transducer to respond.

**Cleared** - Refers to a FALSE or logic zero value.

**Counts** - Raw number of digital counts for a given physical distance. The number has NOT been scaled to represent the actual physical distance.

**Extending** - Moving or turning the axis such that the transducer counts increase.

**Feed Forward** - Control industry term used to express the amount of output drive needed to reach a desired speed, or the open loop gain of a control system.

**MCM** - Motion Control Module.

**Overlap** - Overlap is a region in a hydraulic valve where no fluid can flow in either direction; it is usually specified in percent. Valves with large overlap are very difficult to control.

**PID** - Control industry term used for Proportional, Integral, and Differential drive compensation.

**Position Unit** - Basic unit of distance measurement for a specific application (i.e., thousandths of an inch, tenths of millimeters, thousandths of revolutions, etc.). Position units are calculated using the SCALE and COUNTS parameters.

**Retracting** - Moving or turning the axis such that the transducer counts decrease.

**Scale** - A multiplier used to translate Transducer Counts to Position Units. It can also be used to compensation for differences in magnetostrictive transducers.

**Set** - When referring to STATUS bits it refers to a TRUE or logic one value. It also refers to a physical positions of the axes.

**Target Position**- A calculated position at which an axis is expected to be at any given time.

**Transducer** - A general term for a device that converts a physical distance to an electrical signal.

**Transducer Counts** - See Counts.

## ASCII TABLE

This is a list of all the standard ASCII control characters, listing the keyboard character, hex value, and decimal value of the character.

Key	Hex	Dec	Key	Hex	Dec	Key	Hex	Dec
^@	00	00	+	2B	43	V	56	86
^A	01	01	,	2C	44	W	57	87
^B	02	02	-	2D	45	X	58	88
^C	03	03	.	2E	46	Y	59	89
^D	04	04	/	2F	47	Z	5A	90
^E	05	05	0	30	48	[	5B	91
^F	06	06	1	31	49	\	5C	92
^G	07	07	2	32	50	]	5D	93
^H	08	08	3	33	51	^	5E	94
^I	09	09	4	34	52	_	5F	95
^J	0A	10	5	35	53	`	60	96
^K	0B	11	6	36	54	a	62	97
^L	0C	12	7	37	55	b	62	98
^M	0D	13	8	38	56	c	63	99
^N	0E	14	9	39	57	d	64	100
^O	0F	15	:	3A	58	e	65	101
^P	10	16	;	3B	59	f	66	102
^Q	11	17	<	3C	60	g	67	103
^R	12	18	=	3D	61	h	68	104
^S	13	19	>	3E	62	i	69	105
^T	14	20	?	3F	63	j	6A	106
^U	15	21	@	40	64	k	6B	107
^V	16	22	A	41	65	l	6C	108
^W	17	23	B	42	66	m	6D	109
^X	18	24	C	43	67	n	6E	110
^Y	19	25	D	44	68	o	6F	111
^Z	1A	26	E	45	69	p	70	112
^[	1B	27	F	46	70	q	71	113
^\	1C	28	G	47	71	r	72	114
^]	1D	29	H	48	72	s	73	115
^^	1E	30	I	49	73	t	74	116
^-	1F	31	J	4A	74	u	75	117
space	20	32	K	4B	75	v	76	118
!	21	33	L	4C	76	w	77	119
"	22	34	M	4D	77	x	78	120
#	23	35	N	4E	78	y	79	121
\$	24	36	O	4F	79	z	7A	122
%	25	37	P	50	80	{	7B	123
&	26	38	Q	51	81		7C	124
'	27	39	R	52	82	}	7D	125
(	28	40	S	53	83	~	7E	126
)	29	41	T	54	84	rub	7F	127
*	2A	42	U	55	85			

## Appendix A: DCS120 Diagnostic and Setup Program

### Description

DCS120-DOS is a DOS-based software package that allows you to access, display, troubleshoot, and control features of Delta's MMC120 TSX Quantum motion control products. Used with an IBM PC or compatible, DCS120 allows you to adjust the Motion Control Module's parameters and make simple movements. You can display a motion trajectory using DCS120's graphic or ASCII formats.

DCS120-Win is also available and recommended for Windows 95/98/NT users. DCS120-Win has all the features of DCS120-DOS plus a Windows interface, online help, plot printing, and popup editors that save you from memorizing hexadecimal codes and commands. DCS120-Win is available on the setup disks that came with the MMC120 module and on Delta's web site.

This appendix covers only DCS120-DOS because DCS120-Win has an extensive online manual.

### General Features

- Provides context-sensitive help.
- Provides graphic display of axis movement.
- Provides utilities to calculate motion parameters Scale and Offset
- Provides access to auto feed forward adjust function.
- Allows user to direct simple moves from the keyboard.
- Permits user to change control parameters from the keyboard.
- Displays parameter and status information for multiple axes.
- Permits graphic diagnostic information to be saved and retrieved from disk.
- Provides a display-only mode to display previously-saved diagnostic graphic information using only a PC.
- Saves parameters to disk for backup.
- Includes an Event Control spreadsheet with 256 steps.
- Uses spreadsheet format for editing parameters.
- Provides look only mode (non-editing) for operator reference.
- Provides sum of errors squared ( $\sum e^2$ ) tuning aid; it's displayed on the top line of the plot screen.

To install DCS120 on your hard drive follow the instructions on page 79.

A demo program is installed with DCS120-DOS and DCS120-Win.

### Initial Startup Using Display-Only Mode

For a quick demonstration of the program, setup DCS120-DOS from the disks provided, then type **MMCDOS PC** <Enter> from the installed directory. Since no hardware needs to be connected, DCS120 will display the PC Mode parameters.

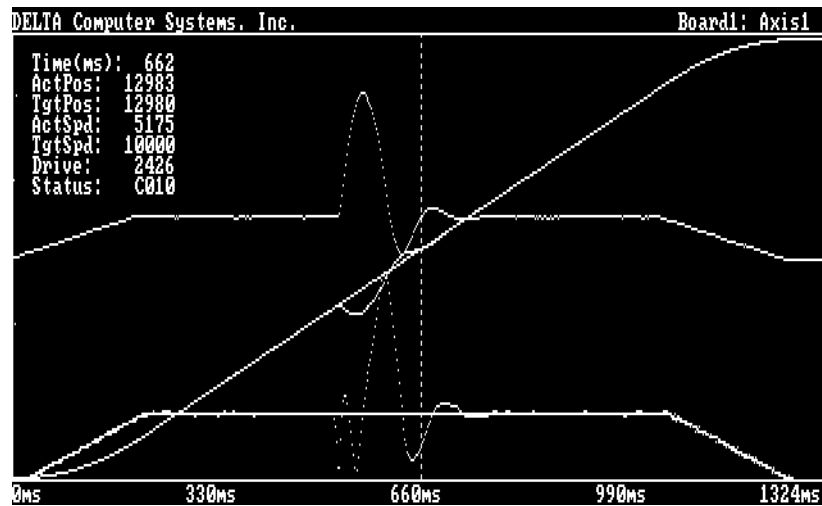
Note "PC AT" in the upper left corner; this indicates PC (display only) mode. See pages 81-79 for the list of available commands.

PC AT 0.99		DELTA Computer Systems, Inc.						MMC120	
19990709									
			PLOT TYPE	0	0	3X+0	3X+1	3X+2	3X+3
			PLOT TIME	1	1				
	Axis1	Axis2		Axis1	Axis2	4X+0	4X+1	4X+2	4X+3
COMMAND POS.	0	0	CONFIG	00000	00000	00000	0	00000	0
TARGET POS.	0	0	SCALE	30300	30300	00000	0	00000	0
ACTUAL POS.	0	0	OFFSET	0	0	00000	0	00000	0
COUNTS	0	0	EXTEND LIMIT	50000	50000	00000	0	00000	0
STATUS	00000	00000	RETRACT LIMIT	1000	1000	00000	0	00000	0
DRIVE	0	0	PRO. GAIN	100	100	00000	0	00000	0
ACTUAL SPEED	0	0	INT. GAIN	100	100	00000	0	00000	0
NULL DRIVE	0	0	DIF. GAIN	10	10	00000	0	00000	0
STEP	0	0	EXT. FEED FWD.	100	100	00000	0	00000	0
LINK VALUE	0	0	RET. FEED FWD.	100	100	00000	0	00000	0
MODE	00001	00001	EXT. ACCEL FF.	50	50	00000	0	00000	0
ACCEL	1000	1000	RET. ACCEL FF.	0	0	00000	0	00000	0
DECEL	1000	1000	DEAD BAND ELIM	0	0	00000	0	00000	0
SPEED	1000	1000	IN POSITION	5	5	00000	0	00000	0
COMMAND VALUE	0	0	FOLLOW ERROR	50	50	00000	0	00000	0
COMMAND			AUTO STOP	00000	00000	00000	0	00000	0
AXIS 1 COMMAND POS. :	0		INPUT>	_					W F

DCS120 Default Startup Screen

	Initialization parameters	White = active; gray = non-active
	Real time read only parameters	Green
	Real time command parameters	Red or yellow depending on mode

You can also display graphic information previously saved on the disk. The DCS120 distribution disk has two demonstration files: stall.plt and multi.plt. You can use the RP (Read Plot) command to display them. Example: type **RP stall <Enter>**.



Stall.plt

## Initial Startup Using Motion Control Module

Before DCS120 can be used with a Motion Control Module you must connect the appropriate hardware. Refer to pages 17-19 for hardware connection information.

Once you have connected the hardware, you must configure DCS120 to match it. Insert the DCS120 disk in drive A and type **a:install <Enter>** (you can substitute drive B for A). Select the directory where you want to install DCS120. You may need to select a different COMM port. Please see explanations of the various configuration files later in this appendix. When you finish the installation, change to the DCS120 directory (default: c:\DCS120) and type **MMCDOS <Enter>** to start the program.

The following is a sample of a typical system screen:

```

MMC120 0.99                      DELTA Computer Systems, Inc.                      MMC120
19990709

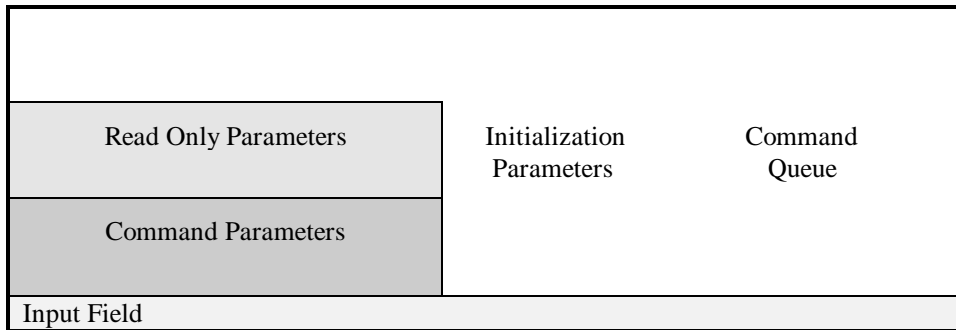
                                PLOT TYPE          0      0      3X+0  3X+1  3X+2  3X+3
                                PLOT TIME          1      1
                                Axis1 Axis2          Axis1 Axis2  4X+0  4X+1  4X+2  4X+3
COMMAND POS.      0      0  CONFIG          00000 00000 00000  0 00000  0
TARGET POS.      0      0  SCALE          30300 30300 00000  0 00000  0
ACTUAL POS.      0      0  OFFSET          0      0 00000  0 00000  0
COUNTS          0      0  EXTEND LIMIT  50000 50000 00000  0 00000  0
STATUS           00000 00000  RETRACT LIMIT  1000  1000 00000  0 00000  0
DRIVE            0      0  PRO. GAIN      100    100 00000  0 00000  0
ACTUAL SPEED     0      0  INT. GAIN      100    100 00000  0 00000  0
NULL DRIVE       0      0  DIF. GAIN      10     10 00000  0 00000  0
STEP             0      0  EXT. FEED FWD.  100    100 00000  0 00000  0
LINK VALUE       0      0  RET. FEED FWD.  100    100 00000  0 00000  0
MODE             00001 00001  EXT. ACCEL FF.  50     50 00000  0 00000  0
ACCEL            1000  1000  RET. ACCEL FF.  0      0 00000  0 00000  0
DECEL            1000  1000  DEAD BAND ELIM  0      0 00000  0 00000  0
SPEED            1000  1000  IN POSITION      5      5 00000  0 00000  0
COMMAND VALUE    0      0  FOLLOW ERROR    50     50 00000  0 00000  0
COMMAND          00000 00000  AUTO STOP      00000 00000 00000  0 00000  0
AXIS 1 COMMAND POS. :      0      INPUT> _                                W F
    
```

### Screen After Installation

**NOTE:** After installation, if the screen contains all zeros in the Read Only (green) section, there is a problem with the hardware connections and/or DCS120's configuration information. Recheck the hardware and re-install DCS120 while double checking the specified setup parameters.

## The DCS120 Screen

The DCS120 main screen is divided into five parts:



### Read Only Parameters

Read Only parameters contain status information about the Motion Control Module. On color monitors this area is green. These parameters are updated automatically by the controller. Do not attempt to write into this section; entries are ignored.

### Command Parameters

These Parameters are written to the module by DCS120. They contain the commands and related parameters for execution by the module. This area can be either red or yellow. When DCS120 is in Readback mode the area is red, and when in Write mode it is yellow. (Note: Readback mode is used to monitor the Motion Control Module as the PLC is giving commands.)

### Input Field

This line is where you enter parameters and commands. If you hold down the *Alt*, *Ctrl* or *Shift* keys, it also displays the available commands. See 'Summary of Keyboard Commands' for more information.

### Initialization Parameters

The initialization parameters contain all the information necessary for the Motion Control Module to control the axes. If you change these parameters, the new values are written to the module only when a 'P' command (*Alt-P*) is issued. These parameters are gray before an *Alt-P* is issued and white after (gray indicates that the parameters have not been transferred to the Motion Control Module).

### Command Queue

The Command Queue is a record of the commands and data sent to the module by the PLC. It shows the most recent 256 events (16 per page, with 16 pages), with the newest at the top of the queue. You can scroll through the record with the Page Up/Page Down keys; a pointer (!) shows which page you are on in the Queue. You can use the queue to confirm that the ladder logic is sending only the expected commands to the MMC120. You can save the Command Queue to a .log file by using the *SQ filename* command (do not enter the '.log' extension).

There are four columns in the Command Queue, with the following meanings:

Column:	4X+0	4X+1	4X+2	4X+3
Meaning:	Cmd to Axis 1	Data Out to Axis 1	Cmd to Axis 2	Data Out to Axis 2



## Summary of Keyboard Commands

When the *Shift*, *Ctrl* or *Alt* keys are pressed, a menu of the available commands is displayed at the bottom of the screen. Pressing the key multiple times will scroll through the entire list of commands available with that key.

To change most parameters, cursor to the field and enter a new value. To see the help file of a parameter, press *Ctrl-H* when the cursor is on that field. To enter hex numbers, type a "0" (the number zero) as the left-most digit of the number. After typing the desired new value, DCS120 will accept the input as soon as you hit *Enter* or move the cursor to a new location.

---

**NOTE:** New values are not automatically transferred to the Motion Control Module. An *Alt-P* command must be issued to transfer initialization parameters.

---

On color monitors the initialization parameters are displayed in gray to indicate parameters have not been transferred. Use an *Alt-P* command on each axis to transfer the parameters to the Motion Control Module. (The screen display of the parameters changes from gray to white after an *Alt-P* is issued.)

### SHIFT Key Commands

---

SHIFT ? Display full help information using text editor

### Ctrl Key Commands

---

Ctrl	B	Display Status bits in expanded format (Toggle)
	C	View configuration
	D	Temporary exit to DOS; return to DCS120 by typing EXIT
	E	Edit .INI configuration file (TE editor is included as default)
	F	Enter table editor for Function keys (F1-F10)
	H	Context-sensitive help
	P	Enter table editor for profiles
	Q	Quit DCS120
	S	Enter Event Control editor
	T	Toggle between Write mode and Readback mode
	V	View datalog raw data (after graph taken)
F1-F10		Executes Command saved for function key. Uses all 6 command parameters stored: MODE, ACCEL, DECEL, SPEED, COMMAND VALUE, and COMMAND

### Function Keys

---

F1-F10	Executes command stored for function key. Uses only COMMAND VALUE and COMMAND parameters
--------	--

### Command Line

---

P0	Pos 0	Cnt 0	Scale and Offset field calculation. Place cursor on axis of interest. See page 82 for more information.
P1	Pos 1	Cnt 1	information.
	R	<i>filename</i>	Reads the project parameters for current motion controller module
	S	<i>filename</i>	Saves current setup using the <i>filename.BDn</i> where <i>n</i> represents current module location; execute for each controller module. Do not enter the .BDn extension.
	RP	<i>filename</i>	Displays graph previously saved using SP command
	SP	<i>filename</i>	Saves current graph in a plot file specified by <i>filename</i> . Do not enter the .plt extension.
	SQ	<i>filename</i>	Saves Command Queue to .log file specified by <i>filename</i> . Do not enter the .log extension.
	T	<i>n</i>	Sets the plot timebase to <i>n</i> milliseconds
	PT	<i>n</i>	Sets the type of the extra plot information. <i>n</i> can be one of the following:
		0	Extra precision positions
		1	Command and Command Value
		2	Event Step and Link Value
		3	Raw Transducer Counts

**Alt Key Commands**

---

Alt	A	Change ACCELERATION
	D	Change DECELERATION
	E	Start Events
	F	Feedforward adjust (auto tuning)
	G	Issue a Go command to the selected axis
	H	Halt command to axis
	I	Integral Drive set
	i	Integral Drive clear
	J	Relative move
	K	Issue Emergency Stop command to all axes
	M	Set MODE
	N	Set NULL
	n	Null Drive clear
Shift-O		Open Loop
	P	Activate control parameters (initializes the axis the cursor is on)
	Q	Quit events
	R	Issue a Restore Null command to the selected axis
	r	Restore Integral Drive
	S	Issue a Save Null command to the selected axis
	s	Save Integral Drive
	U	Update FLASH
	V	Set SPEED

**Additional keys**

---

Esc	Return to previous screen
Ins	Retrieve graphic information from module
Shift-Ins	Retrieve graphic information and Command Queue data from module
Alt-Shift-O	Open loop
Tab	Move the graph data table around the graph
Page Up	Move up through the Command Queue
Page Down	Move down through the Command Queue

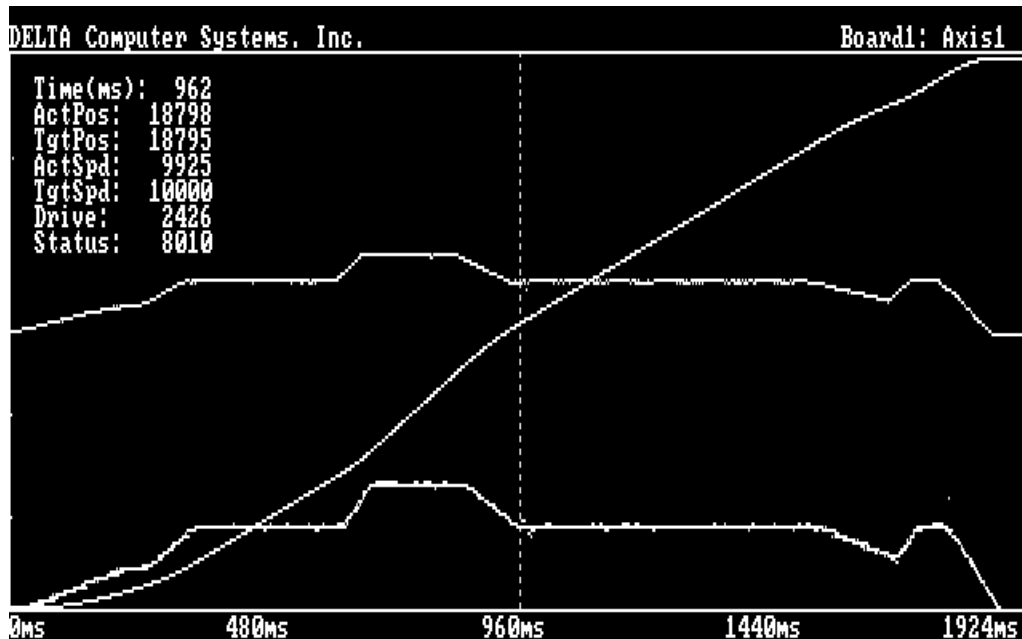
## Diagnostic Graphs

You can see a diagnostic graph after each move started with a “G” or “O” command. The MMC120 acquires data for each move automatically. Press the INS key to display the graph. Press SHIFT-INS to display the graph with Command Queue data included.

The graph displays information with the time base specified in the Plot Time field. Entering a 2 in the Axis 1 Plot Time field causes the module to acquire data for that axis each 2 milliseconds. A value of 10 specifies 10 milliseconds between data points.

The Home, End, PgUp, PgDn, up arrow and down arrow keys move a vertical cursor across the graph, showing the values at that point in time for the Actual Position, Target Position, Actual Speed, Target Speed, Status field and Drive Output. The Tab key moves this text to different locations on the screen (or removes it from the screen) to avoid covering the plot. The lines on the plot are not labeled, but instead are color coded.

Two plot files are included for demonstration purposes: multi.plt and stall.plt. Multi.plt shows you can change the requested position, Speed and Acceleration on the fly without first stopping the axis or causing a discontinuity in the motion. Stall.plt shows how the motion control board tries to keep the Actual and Target Positions together, even after the Actual Position has stalled.



Multi.plt

### Display axis graphic information (uses axis where cursor is currently located)

INS Key	Display axis move datalog in graph form (use <i>Ctrl-V</i> command after INS key to display raw data)
SHIFT INS	Graph display plus command data
Home	Move cursor to start of graph
End	Move cursor to end of graph
← ↑	Move cursor back one time interval (fine adjust)
→ ↓	Move cursor forward one time interval (fine adjust)
PgUp	Move cursor back multiple time intervals (coarse adjust)
PgDn	Move cursor forward multiple time intervals (coarse adjust)

## Keyboard Commands

### Getting Help

To display the help file of a particular parameter, place the cursor on the line of that parameter then press *Ctrl-H*. You can also press *Shift-?* to display the DCS120.HLP file, which contains general help and information about DCS120. The help files are displayed using the editor specified in the .ini file. Follow the editor's instructions for exiting back to the original DCS120 screen.

A *Ctrl-C* will display the configuration information for the Motion Control Module. The *Esc* key returns DCS120 to the original screen.

### Function Keys

You can use function keys F1 through F10 to issue commands to the Motion Control Module. They operate on the cursor's current axis. The function keys must first be programmed with the Function key table editor (*Ctrl-F* to start).

To use the function keys press the function key either by itself or in conjunction with the *Ctrl* or *Alt* keys. Function keys by themselves will issue the Command and Command Value to the Motion Control Module. Using *Ctrl-Fn* will issue the Command, Command Value, Speed, Deceleration, Acceleration, and Mode to the module. *Alt-Fn* will issue the command and retrieve graphic information for bus controllers used with VMC and MC modules.

### Exiting DCS120 Screens

*Ctrl-Q* entered at the main screen terminates DCS120 and returns the PC to the DOS prompt.

The *Esc* key will return you to the previous DCS120 screen.

*Ctrl-D* entered at the main screen will cause DCS120 to temporarily terminate and provide you with a DOS prompt. Type 'Exit' at the DOS prompt to resume execution of DCS120.

### Displaying Axis Status Information

The *Ctrl-B* command displays the expanded-format Status word from the module on the right side of the DCS120 screen; this lets you easily identify the individual status bits.

Enter *Ctrl-B* again to return to the original screen.

### Editing the .INI File

Typing *Ctrl-E* will start the editor program and allow you to modify the initialization file that is being used by DCS120. You must restart DCS120 for any changes to take effect.

### Motion Control Commands

The standard motion control commands (Go, Halt, etc.) can be executed by pressing the *Alt* key and the character key simultaneously. A 'Go' command would be *Alt-G*. See the keyboard command summary in the previous section for available commands.

## Open Loop Command

**CAUTION: OPEN LOOP MODE DISABLES ALL SAFETY CHECKS OF THE MOTION CONTROL MODULES! USE THIS COMMAND WITH CARE!**

You can execute the "O" (open loop) command by pressing *Alt-Shift-O* or entering the ASCII value for O (decimal 79) in the COMMAND field. (First you must specify the millivolts of drive output required in the COMMAND VALUE field.)

## Readback and Write Mode Toggle

*Ctrl-T* toggles between the Readback and Write modes ('R' or 'W' is displayed at the right end of the input line). When in the Readback mode, the Command parameters on the DCS120 screen are red, and all fields are continuously updated from the module. This mode is useful for observing the operation of the Programmable Controller or Bus Master.

When in the Write mode, the control parameters are not updated unless a parameter error occurs; when values are entered from DCS120 they overwrite the values written by the Programmable Controller.

## SCALE & OFFSET Calculation Commands

The P0 and P1 commands provide an easy way to set the motion controller's Scale and Offset parameters. To use these commands, you must know the physical distance and the corresponding number of the transducer COUNTS at two points. Measure the physical distance in position units (for example, thousandths of inches, millimeters) while noting the corresponding COUNTS on the DCS120 main screen. Once you have these numbers, simply enter the "zero" position information using P0 and the other position using P1.

For example, if at position 0 the actual position is 0.000 inches and the COUNTS are 2163, then enter:

**P0 0 2163.** If at position 1 the actual position is 27.500 inches and the COUNTS are 31626, then enter:

**P1 27500 31626.**

---

**Note:** If the COUNTS are not entered, the current value in the COUNTS field will be used.

---

After you enter values for both P0 and P1, the SCALE and OFFSET fields will be updated on the screen (notice the parameters turn gray). Use the *Alt-P* command to activate the newly calculated numbers. (In the above example SCALE changes from 32767 to 30564 and OFFSET changes from 0 to -2018.) The RETRACT and EXTEND LIMITS are updated to their new actual positions using the new SCALE and OFFSET values.

## OFFSET Parameter

The OFFSET and other parameters may be displayed incorrectly on the DCS120 screen under some circumstances. This is because the offset and retract limits specify 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 simply enter the correct value for the Offset and save the parameters to a file (using the *S filename* command). Whenever you start DCS120, read the file (using the *R filename* command) and DCS120 will display the parameters correctly.

## Variable Datalogging Rates

The Plot Time (on the main parameter screen) shows the time base of the graphic display; it can be changed with the T *n* command, where *n* is the time in milliseconds. Each axis is set independently. For example, if you set this field to 10 on axis 1, the plot will represent 1024 samples of 10ms each (10.24 seconds of motion). The default time is 1 millisecond.

## Moving Axes Simultaneously

Axes can be moved simultaneously with DCS120. When you set the sync bit (bit 12) in the MODE word on both axes, the axes will only respond to G (go) commands when the last axis is given a command. This allows you to configure both axes with requested positions then make them move in unison.

## Table Editors

Three Table Editors are available in DCS120. They are used to modify files containing information for the operation of the Motion Control Module. The Table Editors are:

- Function Key Editor (*Ctrl-F*).
- Profiles Editor (*Ctrl-P*).
- Event Steps Editor (*Ctrl-S*).

The Table Editors use commands similar to those used in an Excel spreadsheet:

*Shift*+Arrow keys highlight the work-area to be edited.

*End*, *Home* and Arrow keys move the cursor around the screen.

*Ctrl-C* stores a highlighted area in the buffer for pasting.

*Ctrl-S* will save the table to disk.

*Ctrl-R* will read the table from disk.

*Ctrl-X* deletes the selection and stores it in the buffer for pasting.

*Ctrl-V* pastes the contents of the buffer into the work area.

In addition the following commands are available in the Event Steps Table Editor:

*Ctrl-G* will move the cursor to the next step in the chain.

*Ctrl-Z* will zero the entire Step Table

*Alt-S* will save the Event Steps to the module.

*Alt-R* will read the Event Steps from the module.

## Advanced Features

### DOS Command Line Options

Syntax: MMCDOS [ini\_file] [project\_file] [-options]

When you start DCS120, you can include up to two command line parameters and several startup options. The first parameter is the name of the initialization file DCS120 uses to configure itself. The default file is DCS120.INI. The second parameter is the name of the project files that contain information about each of the Motion Control Modules in the system. The file names are DCS120.BDn (where n is the module number: 1, 2, ...). The ini\_file and project\_file are described later in this section.

The available startup options are:

- w starts DCS120 in the write mode
- r starts DCS120 in the readback mode
- d creates a debug file (debug.txt) with a history of all commands executed from DCS120  
Note: this creates large files that will fill your hard drive if DCS120 runs too long
- f forces initialization of the module when DCS120 starts
- l starts DCS120 in the look-only mode

These options can all be specified in the .ini file, but when specified on the command line they supersede the .ini file. The following examples illustrate the use of command line parameters and options.

You can specify a different board file for each different project. This permits easy use of different control parameters for the same axis.

These are examples of Command Line option use:

- |                   |   |
|-------------------|---|
| MMCDOS            | Uses DCS120.INI and <i>filename</i> .BD1  |
| MMCDOS PROJ -w -f | Uses PROJ.INI and <i>filename</i> .BD1, starts in the write mode, and forces initialization |
| MMCDOS PROJ PROJ2 | Uses PROJ.INI and PROJ2.BD1   |

### Setting Up Advanced Features

To create special project and board files, copy and rename the existing DCS120.INI and DCS120.BDn files. You must keep the new files in the same directory as DCS120's default files.

### Description of Initialization File and Project Files

DCS120 needs several files to run. These files must all be in the same directory.

- |              |  |
|--------------|--|
| MMCDOS.EXE   | Executable program code  |
| filename.INI | Initialization file used to configure DCS120 (ASCII text format)   |
| filename.BDn | Project file used to define startup parameters for both axes on each module. Each module used in a system will have a .BDn file to define axis-specific parameters. <i>Filename</i> for the .BDn files are defined in the filename.INI file [PROJECT] parameter. |

## Initialization File

To operate, DCS120 must have a *filename*.INI file. The purpose of the .INI file is to configure DCS120 to work in the display only mode (PC mode) or work with a TSX Quantum PLC. The .INI file must also specify the type of communications used (port and baud rate).

---

**NOTE:** If the keyword in brackets is not the first item on the line then that line is a comment. All keywords are in brackets and parameters must be separated by at least one space or tab. A sample .INI file follows:

---

```
[TARGET]          PC           ;Used if DCS120 for display mode only (default)
***[TARGET]      MMC120       ;Enabled for TSX Quantum MMC120
[COMM]           COM1        ;Serial port
[EDITOR]         EDIT        ;Name of your ASCII editor (EDIT default)
[PROJECT]        MYPROJ      ;Name of file to store configuration and
                        ; function keys
[FORCEINIT]      NO          ;Initialize the module with parameters stored
                        ; on disk
[READBACK]       YES         ;Select Readback mode or Write mode
[LOOKONLY]       NO          ;Limited access to DCS120 capabilities

[BOARD]          MMC120
[AXIS]           Axis1
[AXIS]           Axis2
[ENDDATA]
```

\*\*\* the line is ignored by DCS120

### [TARGET]

The [TARGET] keyword is used to specify the target system that is being used. The target systems available are listed above with a brief description. Any data after the target name is a comment. Select one of the targets and disable the rest by placing "\*\*\*\*" in front of each of them.

### [COMM]

When communicating through an RS-232 port, the [COMM] line must also be set. The [COMM] line has one parameter: the serial (COM) port. The options for the communication port are COM1 and COM2. These must be capitalized and used exactly as shown. The baud rate is fixed at 9600.

### [EDITOR]

This keyword has one parameter: the name of the text editor you use when *Ctrl-E* is entered. This defaults to EDIT, which is available in DOS 5.0 or later, but you can change this to use your favorite editor. The editor must be in the current directory or specified in the PATH environment variable in your autoexec.bat file.

### [PROJECT]

This keyword has one parameter: the name of the project. This name must not be longer than 8 characters as it is used to read and write the .BDn files. You must use this name when you save the .BDn files with the 'S' command.

### [FORCEINIT]

If set to YES, the initialization parameters will be copied from the .BDn file to the Motion Control Module. This can help during initial setup, but may not be desirable after the system is in operation since it may change operational parameters. If [FORCEINIT] is set to NO, DCS120 will read the parameters from the module.



**[READBACK]**

The [READBACK] parameter determines whether DCS120 starts up in the Readback mode or Write mode. The following configuration allows you to execute DCS120 and to come up in the readback mode without initializing the axis:

```
[FORCEINIT]      NO           ; DEFAULT
[READBACK]       YES          ; DEFAULT
```

This configuration allows DCS120 to initialize all axes and come up in the readback mode:

```
[FORCEINIT]      YES
[READBACK]       YES          ; DEFAULT
```

**[LOOKONLY]**

This keyword controls access to the initialization and command parameters. When [LOOKONLY] is set to YES, DCS120 can only monitor status and plot graphs. When it is NO all capabilities are enabled.

**[BOARD]**

This keyword has one parameter: the module name. This name is displayed in the upper right corner of the main and graph screens. This name is limited to six characters. It can be edited to reflect the function of the module, such as edger, or qdbnd.

**[AXIS]**

This keyword has only one parameter: the name of the axis. The axis name can be up to five characters long; longer names are truncated. The name can be chosen to indicate the function of the axis. There must be two axis keywords immediately after each board keyword.

## Project Files

The project files are used to set the initialization parameters for each axis. They also contain function key definitions and profile definitions. The name of the file is set in DCS120.INI file in the [PROJECT] field but will always use the .BDn extension. Each module in the system has a .BDn file. DCS120 software is shipped with one .BDn file. These files can be edited to match different configurations. If edited, the files should be renamed.

There are three other file types which can be shared by the modules in a system:

.FNn	Function Key settings
.PRn	Move Profiles
.STn	Event steps

Each .BDn file contains the following Initialization Parameters:

;FIELD	AXIS0	AXIS1	AXIS2	AXIS3	; FIELD DESCRIPTION
[FIELD] 10	00000	00000	00000	00000	; MODE
[FIELD] 11	1000	1000	0	0	; ACCEL
[FIELD] 12	1000	1000	0	0	; DECEL
[FIELD] 13	1000	1000	0	0	; SPEED
[FIELD] 14	0	0	0	0	; COMMAND VALUE
[FIELD] 15					; COMMAND
[FIELD] 16	00000	00000	00000	000000	; CONFIG
[FIELD] 17	30300	30300	0	0	; SCALE
[FIELD] 18	0	0	0	0	; OFFSET
[FIELD] 19	0	0	0	0	; EXTEND LIMIT
[FIELD] 20	0	0	0	0	; RETRACT LIMIT
[FIELD] 21	1	1	0	0	; PRO. GAIN
[FIELD] 22	1	1	0	0	; INT. GAIN
[FIELD] 23	0	0	0	0	; DIF. GAIN
[FIELD] 24	100	100	0	0	; EXT. FEED FWD.
[FIELD] 25	100	100	0	0	; RET. FEED FWD.
[FIELD] 26	0	0	0	0	; EXT. ACCEL FF.
[FIELD] 27	0	0	0	0	; RET. ACCEL FF.
[FIELD] 28	0	0	0	0	; DEAD BAND ELIM
[FIELD] 29	50	50	0	0	; IN POSITION
[FIELD] 30	250	250	0	0	; FOLLOW ERROR
[FIELD] 31	00000	00000	00000	00000	; AUTO STOP

The values in Axis2 and Axis3 are not used by the MMC120.

.FNn (Function Key) example:

	KEY#	AXIS	MODE	ACCEL	DECEL	SPEED	POSITION	COMMAND
	[FNKEY] 1	0	00001	100	100	10000	4000	71
	[FNKEY] 2	0	00001	100	100	10000	8000	71
	[FNKEY] 3	0	00001	100	100	10000	12000	71
	[FNKEY] 4	0	00001	100	100	10000	16000	71
	[FNKEY] 5	0	00001	100	100	10000	20000	71
	[FNKEY] 6	0	00001	100	100	10000	24000	71
	[FNKEY] 7	0	00001	100	100	10000	28000	71
	[FNKEY] 8	0	00001	100	100	10000	32000	71
	[FNKEY] 9	0	00001	100	100	10000	36000	71
	[FNKEY] 10	0	00001	100	100	10000	40000	71
	[FNKEY] 1	1	00001	100	100	10000	4000	71
	[FNKEY] 2	1	00001	100	100	10000	8000	71
	[FNKEY] 3	1	00001	100	100	10000	12000	71
	[FNKEY] 4	1	00001	100	100	10000	16000	71
	[FNKEY] 5	1	00001	100	100	10000	20000	71
	[FNKEY] 6	1	00001	100	100	10000	24000	71
	[FNKEY] 7	1	00001	100	100	10000	28000	71
	[FNKEY] 8	1	00001	100	100	10000	32000	71
	[FNKEY] 9	1	00001	100	100	10000	36000	71
	[FNKEY] 10	1	00001	100	100	10000	40000	71

PRn (Profile) example:

	PROF#	MODE	ACCEL	DECEL	SPEED
[PROF]	0	00001	25	25	5000
[PROF]	1	00001	50	50	10000
[PROF]	2	00001	75	75	15000
[PROF]	3	00001	100	100	20000
[PROF]	4	00001	125	125	25000
[PROF]	5	00001	150	150	30000
[PROF]	6	00001	25	150	10000
[PROF]	7	00001	150	25	10000
[PROF]	8	00001	100	100	10000
[PROF]	9	00001	100	100	10000
[PROF]	10	00001	100	100	10000
[PROF]	11	00001	100	100	10000
[PROF]	12	00001	100	100	10000
[PROF]	13	00001	100	100	10000
[PROF]	14	00001	100	100	10000
[PROF]	15	00001	100	100	10000

Sample .STn (Event Step) file:

```

;      STEP#  MODE  ACCEL  DECEL  SPEED  POSITION  COMMAND  LINKTYPE  LINKVALUE  LINKINDEX
[STEP]   1  08001  100   100   5000  14000    G         D           500         17
[STEP]   2  08001  100   100  10000  15000    G         D           5000        25
[STEP]   3  08001   0     0     0     0     0         0           0           1
:
:
[STEP]  17  00001   5     5    500  11000    G         D           1500        18
[STEP]  18  08001   5     5    250   1000    G         D           1100        21
[STEP]  21  08001   5     5    250  11000    G         D           8000        22
[STEP]  22  08001  25    25   2000  15000    G         D           500         23
[STEP]  23  08001  50    50   1000  32767    C         0            0           1
[STEP]  25  00001  50    10  25000  30000    G         D            40         27
[STEP]  26  08001  50    50  25000  30000    G         D            20         27
[STEP]  27  08001  200   200  25000  30000    G         D            100        28
[STEP]  28  08001  50    200  25000  28000    G         D            100        29
[STEP]  29  08001  100   200  25000  28000    G         B            32         30
[STEP]  30  08001  100   200  25000  28000    G         D            20         31
[STEP]  31  08001  100   100  25000  29000    G         D            0          32
[STEP]  32  00001   1     1     1   49151    C         D            0           0

```

## User Notes

### Negative Offset

DCS120 displays both positive and negative numbers in the position, extend limit, and retract limit fields. The way position and limit fields are displayed is determined by the offset field. Most applications requiring an offset use a negative offset. The rule is: all numbers between the negative offset and 0 are displayed as negative numbers.

Example:

OFFSET	0	position range	0 to 65535 (65536 total steps)
OFFSET	-1000	position range	-1000 to 64535
OFFSET	-6000	position range	-6000 to 59535
OFFSET	-10000	position range	-10000 to 55535
OFFSET	-20000	position range	-20000 to 45535
OFFSET	-30000	position range	-30000 to 35535

For any negative offset, values greater than the maximum position but less than the maximum number of steps are treated and displayed as negative numbers:

OFFSET -30000	position range	-30000 to 35535	(65536 total steps)
---------------	----------------	-----------------	---------------------

A position request of 45000 is displayed as:  $45000 - 65536 = -20536$

### View Data Command

The View Data command (*Ctrl-V*) displays raw Counts, Target, Status, and time information. (Speed values are calculated by DCS120.)

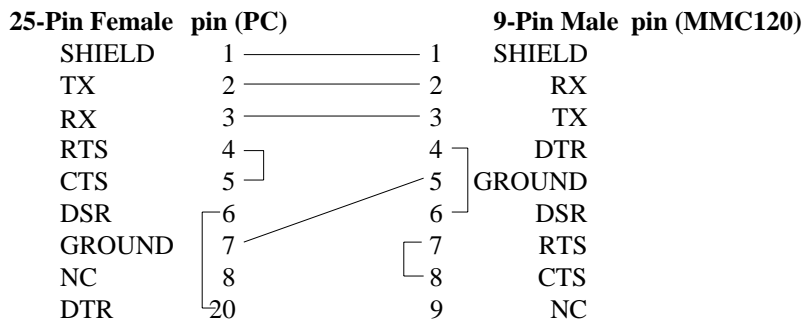
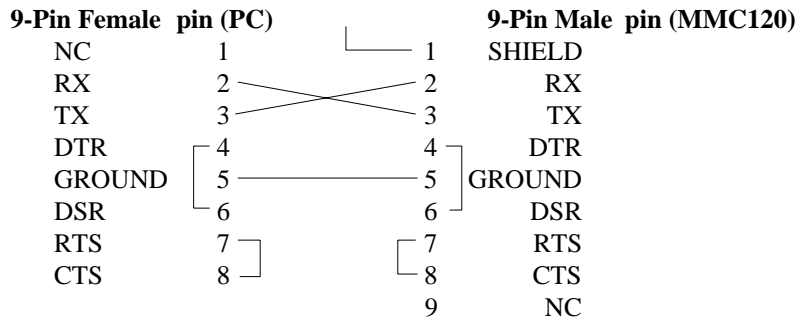
### Sum of Errors Squared ( $\sum e^2$ )

A sum of errors squared term is included on the top line of the plot screen. You can use this term by noting if it's getting smaller or larger as you change parameters; the smaller the number the better-tuned the axis. Repeat the tuning steps in the MMC120 manual using the same profile terms until you get the lowest number.

**MMC120 Communication Cable**

DCS120 communicates with the MMC120 through the RS-232 port on the module.

**RS-232 Cable (same as Modbus cables for PC)**



The communication rate between DCS120 and the MMC120 is 9600 baud.

A demo program is available on the DCS120 distribution disk.

## Appendix B: Event Control

### Introduction

The Event Control feature in the MMC120 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 ladder logic programming required.

### Overview

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

Both axes can execute any of the sequences at any time. After the Event Control Steps are configured, the Programmable Controller need only start a Step sequence to execute a complicated motion profile. Steps in the sequence can respond to internal conditions - positions, speeds, and status - in one millisecond rather than at the Programmable Controller scan time.

### 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:

Mode	Mode Word
Accel	Acceleration
Decel	Deceleration
Speed	Requested Speed
Command Value	Requested Position or command value
Command	Any valid ASCII command
Link Type	Condition that triggers execution of next Step
Link Value	Parameter associated with Link Type
Link Next	Number of next Step in sequence

### Commands

Any command which can be issued by the Programmable Controller can be used in a Step.

### Link Word

The Link word, which indicates the next Step to be executed (the Link Next byte), also controls when the Step gets executed (the Link Type byte). The Link Next and Link Type bytes are stored in the two bytes of a 16 bit word:

Link Next (8 bits)	Link Type (8 bits)
-----------------------	-----------------------

The complete Step is stored in memory in the following format:

MODE
ACCEL
DECEL
SPEED
COMMAND VALUE
COMMAND
LINK NEXT   LINK TYPE
LINK VALUE

## Step Table

Steps are stored in MMC120 memory. There are eight 16 bit words per Step (6 for the command and 2 for the link). The table has a maximum length of 2048 words (8 x 256).

The table is not necessarily 2048 words long. It is only long enough to hold all the steps up to the highest Step number used. If the highest Step number used is 49 then only 400 words of memory will be used  $[(49 \times 8) + 8]$ .



## Linking Steps

### Link Next

Link Next is the number of the Step to be executed in the sequence as soon as the conditions specified by Link Type and Link Value are met. Its range is 0 to 255.

### Link Type and Link Value

Link Type and Link Value specify the condition which causes the MMC120 to execute the next Step in a sequence. There are two link types: Basic and Enhanced. The Basic Link Types issue commands to an axis based on its own position, speed, or status. The Extended Link Types issue commands to an axis based on the position, speed, or status of the other axis. The Basic link types are case-significant ASCII values, while the Extended link types are specified with a hex number in which the bits specify the axis number and link type.

#### Basic Link Types

Link Type	Description	Link Value	Range
0 (zero)	End of chain	None	None
B	Status bits set	Bit pattern	Any bits
b	Status bits cleared	Bit pattern	Any bits
D	Delay	Time in milliseconds	0 to 65,535
L	Programmable Limit switch (Target Position set-point)	Position in Position Units	0 to 65,535
l	Actual position	Position in Position Units	0 to 65,535
N	Relative from Command Position	Position in Position Units	0 to 65,535
n	Relative from Actual Position	Position in Position Units	0 to 65,535
R	Target Position relative from Start	Position in Position Units	0 to 65,535
r	Actual Position relative from Start	Position in Position Units	0 to 65,535
S	Target Speed	Speed in Position Units per second	0 to 65,535
s	Actual Speed	Speed in Position Units per second	0 to 65,535

#### Link Type 0 (zero) - End of sequence

When this Link Type is encountered the sequence of steps is terminated. No additional steps are executed unless they are started as a new sequence.

#### Link Type B - Status bit on

This Link Type causes the MMC120 to execute the next Step in the sequence as soon as one of the selected status bits is set to 1. The Link Value for Link Type B is a bit pattern expressed in hexadecimal. For example, bit pattern 0000 0000 0001 0000 binary is 0010 hex, and specifies bit 12 (State Bit A, indicating acceleration). If this bit is set the next Step will execute.

**Link Type b - Status bit off**

This Link Type causes the MMC120 to execute the next Step in the sequence as soon as one of the selected status bits is set to 0. The Link Value for Link Type b is a bit pattern expressed in hexadecimal. For example, bit pattern 0000 0000 0010 0000 binary is 0020 hex, and specifies bit 11 (State bit B). If this bit is set to 0 the next Step will execute.

**Link Type D - Delay**

When a Step containing this Link Type is entered, the command is executed and a timer is started. When the timer reaches the value specified by the Link Value, the next Step will execute. The delay time specified by link value is in milliseconds. This Link Type allows delays between steps of 0 to 65 seconds in one millisecond increments.

**Link Type L - Limit switch**

When the Target Position of the axis reaches the value specified by Link Value, the next Step in the chain executes. This Link Type can be used to change speeds on-the-fly or trigger events on the other axis.

**Link Type l (lower case L) - Actual Position**

When the Actual Position of the axis reaches the value specified by the Link Value, the next step in the chain executes. This link type functions like the 'L' link type.

**Link Type N – Target Position Relative from Command Position**

This type is similar to the 'L' link type but the target position is relative to the Command Position rather than an absolute position.

**Link Type n – Actual Position Relative from Command Position**

This link type is similar to the 'N' link type but the distance is from the Command Position.

**Link Type R – Target Position Relative from Start**

The next Step in the sequence executes when the Target Position reaches the value specified by Link Value.

**Link Type r – Actual Position Relative from Start**

The next Step in the sequence executes when the Actual Position reaches the value specified by Link Value.

**Link Type S - Target Speed**

The next Step in the sequence executes when the Target Speed (calculated by the Target Generator) reaches the value specified by Link Value.

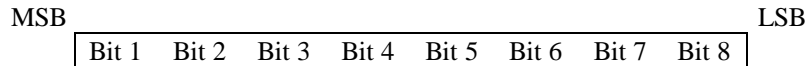
**Link Type s - Actual Speed**

The next Step in the sequence executes when the Actual Speed reaches the value specified by Link Value.

### Enhanced Link Types

Enhanced Link Types provide a way to trigger an event on one axis from the other axis' parameters. The Enhanced Link Types have the same link description, value, and range as the Basic Link Types, plus additional link types.

The bits in the Enhanced Link Type byte are defined as follows:



Bit 1 - Enhanced Link Type flag. 1 = Enhanced Link Type  
0 = Basic Link Type

Bits 2 and 3 - Define target axis:  
01 = Axis 2  
00 = Axis 1

Bit 4 - Direction flag: 1 - Enables Step when monitored parameter is greater than link value  
0 - Enables Step when monitored parameter is less than link value

Bit 5 - Reserved

Bits 6, 7, and 8 - Define link type:  
111 = Reserved  
110 = Status Word bit  
101 = Actual Speed  
100 = Target Speed  
011 = Reserved  
010 = Reserved  
001 = Actual Position  
000 = Target Position

For example:

Parameter	Link Type Byte		Link Type Byte		Go to Link Next when:
			Binary	Hex	
Target Pos.	2	1	1011 0000	B0	Axis 2 Target Position >= Link Value
Actual Speed	1	0	1000 0101	85	Axis 1 Actual Speed <= Link Value

### Step Execution

When a Step is started the command is executed immediately, then the link condition (Link Type and Link Value) is evaluated. If the condition is met, the next Step in the sequence (Link Next) executes on the next millisecond control loop interval. If the condition is not met, it will be re-evaluated every millisecond until the condition is met, then the next Step will execute.

## 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
<b>Mode</b>	00001	00001	00001
<b>Accel</b>	100	100	100
<b>Decel</b>	100	100	100
<b>Speed</b>	10000	10000	10000
<b>Command Value</b>	15500	10000	3000
<b>Command</b>	G		G
<b>Link Type</b>	B	D	0
<b>Link Value</b>	00001	500	0
<b>Link Next</b>	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 MMC120 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 entered, 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” to 3 inches. The Link Type 0 (zero) causes the axis to terminate the sequence.

## Starting a Sequence

### Triggering Events

To trigger or start an event sequence, place the Step number you want to execute (0 to 255) in the Command Value field, then issue an 'E' command.

Mode	00001	Not used (can be any value)
Accel	100	Not used (can be any value)
Decel	100	Not used (can be any value)
Speed	10000	Not used (can be any value)
Command Value	4	Execution will begin at Step 4
Command	E	69 Decimal (45 Hex)

### Ending a sequence

A **Halt**, **Quit**, or **Kill** command will cause the axis to stop executing Steps. A **Quit** or **Kill** command also sets the Step # to 0, and a **Kill** command also puts the axis in the emergency stop state. The preferred way to end a sequence is with a Link Type 0.

## Using DCS120

Step information for the MMC120 is generally programmed using DCS120, but you can modify the Step table directly with the P/C by using Command Type E and Extended E (see pages 63 and 64).

MMC120 0.99		DELTA Computer Systems, Inc.									
19990709		0	1	2	3	4	5	6	7	8	9
MODE		00000	00001	08081	00080	00000	00000	00000	00000	00000	00000
ACCEL		0	100	100	0	0	0	0	0	0	0
DECEL		0	20	100	0	0	0	0	0	0	0
SPEED		0	5000	1000	0	0	0	0	0	0	0
POSITION		00000	20000	2000	0	0	0	0	0	0	0
COMMAND			G	G	G						
LINK TYPE			L	0A0							
LINK VALUE		0	10000	0	0	0	0	0	0	0	0
LINK NEXT		0	2	3	0	0	0	0	0	0	0
LINK TYPES		LINK VALUES			LINK NEXT						
0	End of Chain	N/A			N/A						
B	Status Bits	bits in hex			next Step number						
D	Delay 0	milliseconds			next Step number						
L	Limit Switch	position units			next Step number						
S	Speed	position units/sec			next Step number						
E	Enhanced	parameter dependent			next Step number						

### Accessing the DCS120 Step Editor screen

To go to the Step Editor screen from the main DCS120 screen, press Ctrl-S. To return to the main screen, press Esc.

### Step Editor Commands

To cut and paste regions of the table, hold down the Shift key and move around with the cursor keys to select the desired region. After you select it, you can cut it with Ctrl-X or copy it with Ctrl-C and paste it with Ctrl-V.

Ctrl-G will 'go' to the next Step in the sequence. You can use Ctrl-G to step through a sequence and confirm it operates as expected. If you enter an 'e' in the link type field, the Enhanced Event Editor will start. This editor will assemble the needed link type value based on your selections in this editor.

Ctrl-Z will set all values in the table to zero.

### Reading Step files from disk and module

To read a Step file from disk to DCS120, press Ctrl-R. A list of available step files will be displayed; enter the desired file name and press Enter (but do not enter the .ST1 extension). To read the Step file from the MMC120 to DCS120, press Alt-R.

### Sending Step files to disk, module, and Flash memory

To save a Step file from DCS120 to disk using the current file name, press Ctrl-S; to save it with a new file name, enter "*filename*" on the command line (do not enter the '.BD1' extension; DCS120 automatically adds the correct extension). To send a Step file from DCS120 to the MMC120, press Alt-S. To save the Step file currently in the MMC120 to FLASH memory, enter '2' in the Position field, then press Alt-U. The 'U' command uses the Acknowledge bit (9) of the Status word to report the result of the update. If the FLASH updates successfully, bit 9 of axis 1 is set; if the update was unsuccessful, bit 9 of axis 2 is set.

Step Table Send/Receive Commands	Disk	DCS120	MMC120	FLASH
Alt-R		X <-----X		
Alt-S		X ----->X		
Ctrl-R	X -----> X			
Ctrl-S	X <----- X			
Alt-U(2)			X -----> X	

## Notes

### Reading the Current Step

The Step field value indicates which step the axis is current executing; the P/C can read this field to find the step being executed. When the axis is not executing steps this field is 0.

### Terminating a Loop

Step chains which are not endless loops should end with Step zero. The last Step which executes a command should have a link type 0 (zero).

### Interrupting a Sequence

You can exit out of a sequence before it reaches its end (if it has an end) either by issuing a command from the Programmable Controller or DCS120, or by starting another event sequence. Programmable Controller commands and DCS120 commands have priority over all Step commands.

### Multiple Axes Can Use the Same Steps

Both axes can be made to follow identical patterns simultaneously by using the same steps on both.

### No Conditional Branching

Step sequences cannot branch. An axis cannot wait for more than one condition (Link Type) at a time.

## Index

### A

ACCELERATION, 14, 15, 24, 25, 26, 32, 34, 35, 37, 50, 53, 54, 55, 56, 58, 61, 63, 81  
ACTUAL POSITION, 13, 14, 23, 24, 27, 29, 30, 31, 32, 40, 41, 42, 64  
ACTUAL SPEED, 43, 76, 78  
ASCII, 49, 54, 55, 74, 75, 84, 86, 87, 94, 96  
Auto Stop, 16, 24, 33, 40, 41, 44, 46, 53, 61, 65, 76, 78, 89  
AXIS STATUS, 40

### C

Checksum, 45, 66  
Circulation, 73  
COMMAND, 14, 24, 26, 30, 31, 32, 34, 37, 42, 43, 45, 55, 56, 57, 58, 63, 76, 78, 80, 84, 89, 90, 91, 95, 100  
COMMAND VALUE, 14, 24, 26, 30, 31, 34, 37, 42, 45, 55, 56, 57, 58, 63, 76, 78, 80, 84, 89, 95  
Configuration, 27, 45, 78, 80, 83, 87, 88

### D

DCS120, 15, 17, 22, 23, 24, 25, 26, 27, 29, 33, 38, 45, 46, 47, 50, 55, 56, 58, 59, 65, 67, 69, 75, 76, 77, 78, 79, 80, 83, 84, 85, 86, 87, 88, 89, 92, 93, 100, 101, 102  
DCSMON, 69, 80, 101  
Dead band, 32, 67, 68  
DIFFERENTIAL GAIN, 25, 31, 46, 53, 61  
Disable Drive Output, 56  
DRIVE, 24, 43, 64, 76, 78  
Drive Output, 10, 15, 18, 20, 56, 64, 82

### E

Event Control, 15, 47, 75, 80, 94  
EXTEND FEED FORWARD, 25, 26, 32, 53, 61  
EXTEND LIMIT, 23, 24, 26, 30, 37, 42, 46, 53, 61, 84

### F

Feed Forward, 32, 55, 71, 73  
Following error, 25, 26

### G

G, 24, 26, 37, 40, 41, 54, 55, 74, 81, 82, 83, 85, 91, 99, 100  
GO, 24, 26, 41, 50, 51, 55  
Graph, 34, 36, 58, 69, 82

### H

H see HALT, 24, 26, 42, 54, 55, 65, 74, 80, 81, 83  
HALT, 26, 43, 54, 55, 65

### I

I/O, 11, 15, 17, 46, 47, 48, 66  
IN POSITION, 32, 34, 42, 53, 61, 76, 78, 89  
Initialization, 15, 22, 40, 46, 53, 56, 61, 65, 79, 80, 83, 86, 88, 89  
INTEGRAL, 14, 25, 31, 46, 53, 61, 67  
INTEGRAL GAIN, 14, 25, 31, 46, 53, 61  
Interrogation, 13, 18, 19, 20, 73

### L

Link Next, 94, 96, 98, 99  
Link Type, 94, 96, 97, 98, 99, 102  
Link Value, 64, 94, 96, 97, 98, 99  
Link Word, 94

### M

Magnetostrictive, 9, 11, 13, 14, 17, 18, 27, 29, 30, 73  
MODE, 14, 24, 28, 34, 35, 36, 38, 39, 50, 51, 52, 54, 55, 58, 59, 60, 63, 76, 78, 80, 81, 84, 85, 89, 90, 91, 95, 100  
MODE WORD, 94

### N

NULL DRIVE, 24, 76, 78

### O

OFFSET, 22, 23, 29, 30, 31, 40, 46, 53, 61, 76, 78, 84, 89, 92  
Open Loop, 16, 33, 34, 37, 39, 42, 44, 49, 56, 58, 59, 81, 84

### P

P (Initialization), 22, 23, 24, 26, 31, 41, 42, 45, 47, 53, 54, 56, 74, 79, 80, 81, 84, 85, 94, 100, 102  
P/C see Programmable Controller, 47, 94, 100, 102  
PARAMETER ERROR, 30  
PID, 9, 15, 25, 73  
PID loop, 15  
Pressure, 69, 71  
Programmable Controller, 9, 13, 15, 17, 22, 24, 26, 27, 41, 46, 47, 50, 53, 55, 59, 61, 64, 65, 66, 84, 94, 102  
PROPORTIONAL GAIN, 14, 25, 31, 46, 53, 61



**R**

Requested Position, 14, 30, 34, 42, 55, 85  
Restore Null, 57, 81  
RETRACT FEED FORWARD, 15, 25, 32, 53, 61  
RETRACT LIMIT, 23, 24, 26, 30, 31, 37, 46, 53, 61, 76,  
78, 89

**S**

Save Null, 57, 81  
SCALE, 22, 23, 29, 30, 31, 35, 37, 40, 46, 53, 61, 73, 76,  
78, 84, 89  
Set, 17, 22, 24, 27, 37, 41, 49, 51, 52, 53, 54, 55, 56, 73,  
81  
Set Feed Forward, 55  
Set Parameters, 53, 56

Simulation, 36  
SPEED, 14, 24, 25, 26, 32, 34, 35, 37, 38, 40, 43, 50, 51,  
52, 54, 55, 56, 58, 59, 60, 63, 76, 78, 80, 81, 89, 90,  
91, 95, 100  
Status Area Request, 65  
STATUS WORD, 44  
Step, 43, 45, 47, 55, 63, 65, 85, 95, 97, 99, 100, 101, 102  
Step Editor, 100  
Step Table, 57, 85, 95, 101  
Steps, 24, 85, 94, 95, 96, 102  
Sync, 34

**T**

TARGET, 14, 27, 31, 32, 41, 42, 43, 76, 78, 87  
TRANSDUCER COUNTS, 23, 29, 30, 40, 43  
Triggering Events, 99