MMC 188/40

Motion Controller Module

970303

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

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.

Printed in U.S.A.

TABLE OF CONTENTS

OVERVIEW

The MMC 188/40 motion controller module was designed to provide a complete 4-axis position control sub-system for Modicon® 800 series I/O racks. The axes are controlled by an on-board processor that provides a complete PID loop in addition to numerous special functions. Moves can be coordinated and position changes can be made on-the-fly. The module has four optically isolated magnetostrictive transducer interfaces and four optically isolated outputs configurable for hydraulic valves and electric servo motors.

The MMC 188/40 occupies one slot of the 800 series I/O rack. Communication between the 984® Programmable Controller and the MMC 188/40 is done over the OURBUS through 8 bidirectional registers. The on-board 80C188EB processor relieves the Programmable Controller of the overhead usually associated with servo control. The motion controller module updates the axis position and drive output every two milliseconds, assuring precise positioning even at high speeds.

If more than 4 axes of control are required, additional MMC 188/40 modules can be installed.

Features

General

- 800 I/O compatible (ModConnect®).
- OURBUS interface.
- 80C188EB processor operating at 13.875 MHz.
- 2 millisecond control loop time.
- Linear resolution to 0.001 inches.
- Maximum velocity in excess of 60 inches per second.
- Maximum position count 65535.
- Hardware fail-safe timers.
- Diagnostics and tuning software.
- Hot Swappable.
- Optional EEPROM for parameter and profile storage.

Position Inputs

- 4 optically isolated Magnetostrictive Transducer (Temposonics™) interfaces
- 1.5V return pulse sensitivity
- 27.75 MHz counters
- 1, 2 or 4 recirculations on board
- Positive or negative interrogating pulses
- Direct interface to magnetostrictive transducers

Drive Output

- 4 optically isolated, individually configurable servo drive outputs:
  - ± 100mA current output
  - ± 50mA current output

Modicon, ModConnect and 984 are registered trademarks of Modicon, Inc.

± 25mA current output

± 10 Volt output

± 5 Volt output

±2.5 Volt output

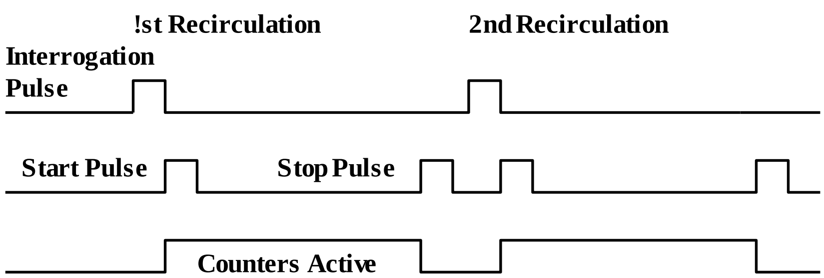
- 12 bit digital to analog converters

Description

Principle Of Operation

Position Measurement

The MMC 188/40 has the necessary circuitry to interface directly to magnetostrictive transducers (Temposonics™). To initiate a measurement, an interrogation pulse is sent from the MMC 188/40 to the transducer. The transducer responds by returning 2 pulses - a *start* pulse and a *stop* pulse. The time between the 2 pulses is proportional to the position of the axis. This is one "recirculation."



Counters on the MMC 188/40 count the number of cycles of a 27.75 MHz clock between the start and stop pulses. With this frequency, each clock cycle is equivalent to approximately 0.004" (4 thousandths of an inch) displacement of the axis. To increase the measurement resolution, 2 or 4 recirculations can be used. With 4 recirculations, the counts between 4 sets of start/stop pulses are added together for a single position measurement. This in effect increases the resolution 4 times - from 0.004" to 0.001".

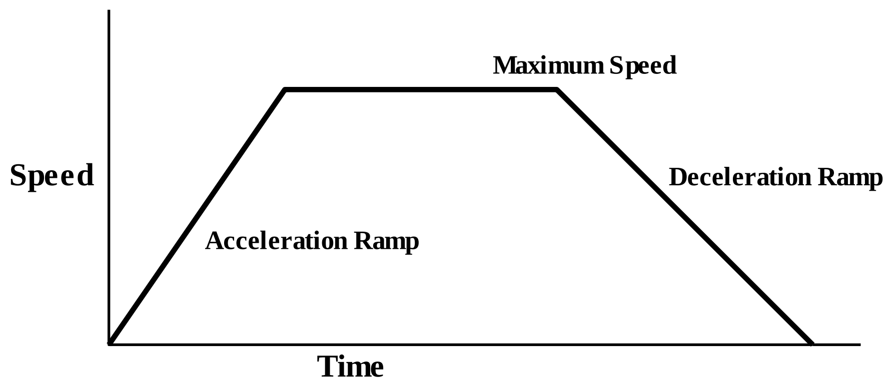
The MMC 188/40 converts the TRANSDUCER COUNTS read from the counters to an ACTUAL POSITION in user-defined Position Units (usually 0.001" using 4 recirculations). The ACTUAL POSITION is then used by the MMC 188/40 for all subsequent internal calculations.

Control Loop

The MMC 188/40 is a targeting controller. Every 2 milliseconds the TARGET POSITION is updated by the on-board microprocessor. For point-to-point moves, TARGET POSITIONS are generated such that the velocity follows a trapezoidal profile. The ACCELERATION, DECELERATION and MAXIMUM SPEED used to generate the profile are specified by the user, and can be changed while the axis is in motion.

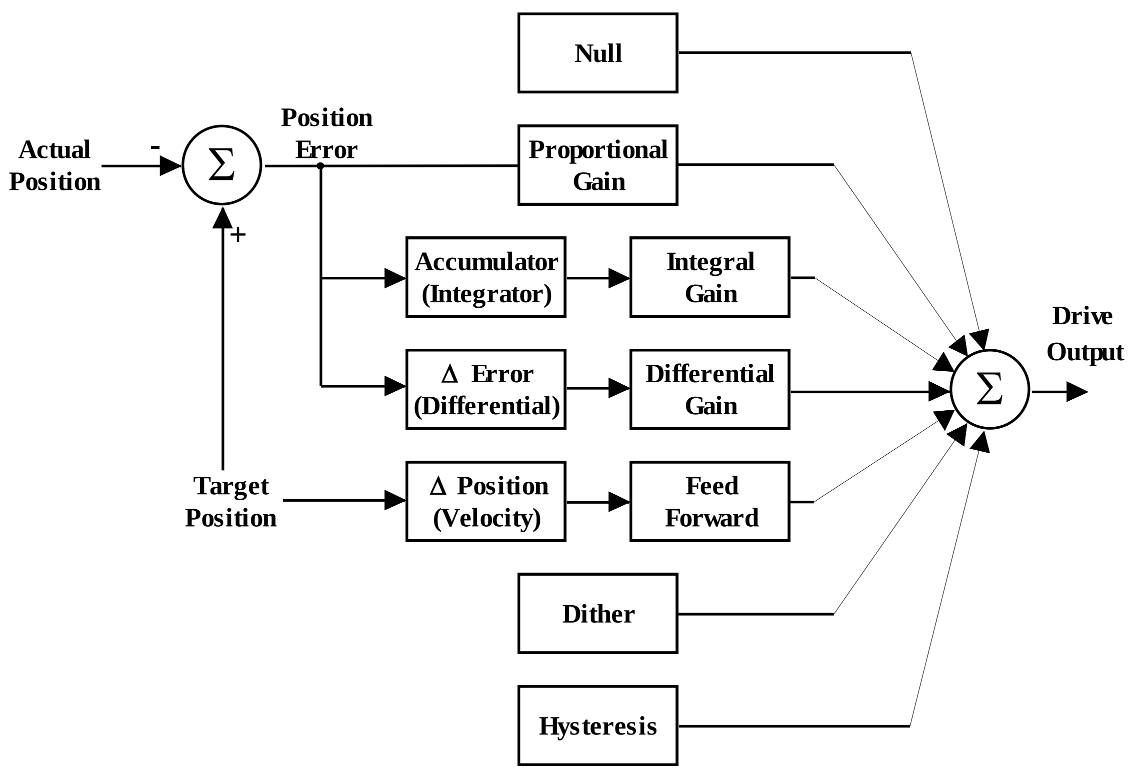
The ACTUAL POSITION measured from the magnetostrictive transducer is compared to the TARGET POSITION to determine the position error. The position error is used to calculate the closed loop component of the drive output every 2 milliseconds. It is multiplied by either the PROPORTIONAL STATIC GAIN, PROPORTIONAL EXTEND GAIN or PROPORTIONAL RETRACT GAIN to calculate the proportional component of the drive output. The *change* in position error is used, along with the DIFFERENTIAL GAIN, to calculate the differential portion 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 position error is also use in conjunction with the NULL UPDATE to

Temposonics is a trademark of MTS Systems Corporation.



correct for null drift in hydraulic valves.

In addition to the closed loop drive, the MMC 188/40 has 3 open loop components to its drive output: HYSTERESIS, DITHER, and Feed Forward (EXTEND FEED FORWARD, RETRACT FEED FORWARD and FEED FORWARD ADVANCE),. These open-loop components of drive help keep the ACTUAL POSITION tracking the TARGET POSITION during various parts of the move or at rest.



Drive Output

The drive calculated by the MMC 188/40 is sent through optocouplers to a 12 bit Digital-to-Analog converter (DAC). The output from the DAC is amplified to provide either current or voltage output to the hydraulic valve or servo amplifier. The output mode (current or voltage) is jumper selectable as is the output level - ±25mA, ±50mA or ±100mA in current mode, or ±2.5V, ±5V or ±10V in voltage mode.

OURBUS Interface

The MMC 188/40 communicates with the 984 controller over the OURBUS. It is traffic coped as a B886, 8 register, binary bidirectional I/O module. Commands and status for all four axes are transferred back and forth in groups of 8 16-bit words.

Programming

The commands for the MMC 188/40 are sent to the module by writing to the output registers. The first register is always a command. The following 7 registers are either commands or data depending on the first command's type. Details on programming the MMC 188/40 are given in the section "COMMUNICATING WITH THE MMC 188/40".

Custom Loadable Function Block (FN10)

A Custom Loadable Function Block (FN10) is provided with the MMC 188/40 to facilitate communications with the module. FN10 consists of 29 sub functions which include initialization blocks and various math functions.

EEPROM (Optional)

An Electrically Erasable Programmable Read Only Memory (EEPROM) option is available on the MMC 188/40 for parameter and profile storage. This reduces the memory requirements in the Programmable Controller and eliminates the need to transfer initialization parameters back and forth.

Specifications

Ambient

Temperature

Operating: 0 ^{o}C to +60 ^{o}C.

Storage: -40 ^{o}C to +85 ^{o}C

Humidity

90% non condensing.

Vibration Resistance

1 g at 60 to 500 Hz for 23 min.

0.075 mm displacement from 10 to 60 Hz.

Shock Resistance

30 g for 11 milliseconds.

ESD Immunity

8 KV

Power Requirements

Backplane

+5V @ 500 milliamps

Magnetostrictive Transducer Section

+5V @ 100 milliamps

±15V @ 500 milliamps max. (Actual current requirements vary depending on the application. When powering Temposonic I and neuter Temposonic II transducers from the MMC 188/40, sufficient current must be provided. The total required current is found as follows: Current = 100mA + 55mA per Temposonic I + 100mA per neuter Temposonic II. Other transducers must be connected directly to the power supply.

Drive Output Section

±15V @ 500 milliamps max. (Actual current requirements vary depending on the application, and is calculated as follows: Current = 100mA + 10mA x number of axes used in voltage mode + output range of axes used in current mode.)

For example, if 1 axis is used in voltage mode, one axis is in current mode with a range of ±25mA, another is in current mode with a range of ±100mA, an the fourth axis is unused, the current required would be 100mA + 10mA + 25mA +100mA = 235mA.

Transducer Inputs

Direct interface to Magnetostrictive Transducers

1.5V input sensitivity. (A differential to single-ended converter is available for connecting transducers with differential outputs.)

Start and Stop pulse required

Outputs

±100 milliamps into 25 to 75 ohm load

±50 milliamps into 25 to 150 ohm load

±25 milliamps into 25 to 300 ohm load

±10 Volts into 1000 ohm or greater load

±5 Volts into 1000 ohm or greater load

±2.5 Volts into 1000 ohm or greater load

Isolation

All inputs and outputs used for control purposes are optically isolated from each other and from the rest of the module. The withstand voltage is rated at 2500VDC.

The RS-232 port used for DCSMON is not optically isolated.

I/O Protection

All inputs and outputs are protected against transients with transorbs.

Bus Compatibility

ModConnect partners

MODICON 800 series I/O

Direct connection to OURBUS backplane

B886 with 8 bidirectional binary registers

128 input and 128 output points per module

Fail Safe Timers

The MMC 188/40 has four fail safe timers:

Drive Output Disable - 15 microsecond

The fastest fail-safe timer on the MMC 188/40 is set for 15 microseconds and is re-triggered every time there is activity on the module's internal bus. If the timer fails to be re-triggered within 15 microseconds, the drive outputs will be disabled. When internal bus activity resumes, the drive outputs are re-enabled.

Software Restart - 15 milliseconds

The 15 millisecond fail safe timer is connected to the module non-maskable interrupt. Normally, the fail safe timer is re-triggered by the on-board processor every two millisecond. If the processor fails to re-trigger the fail safe timer, a software restart will occur after 15 milliseconds if the non-maskable interrupt vector is not corrupted.

Hardware Reset - 50 milliseconds

If the software restart fails, the 50 millisecond hardware reset timer will expire and a hardware reset will be executed. This has the same effect on the MMC 188/40 as powering up.

OURBUS Reset - 50 milliseconds

If the CPU fails to re-trigger the OURBUS fail-safe timer, it will expire after 50 milliseconds causing a reset on the interface circuitry. The Programmable controller will detect this as a failed I/O module.

FRONT PANEL INDICATORS

There are six multicolored light emitting diodes (LED) on the front panel of the MMC 188/40. The LEDs provide status information about the module and each of the four axes.

The LED labeled "Run" is the run indicator for the MMC 188/40 microprocessor. When the Run indicator is green, the drive outputs are enabled. If the Run indicator is red (processor dead), the drive outputs are disabled and the axes will not move unless the axis is out of null. The run indicator is controlled by a 15 microsecond fail-safe timer that is reset by activity on the processor bus.

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

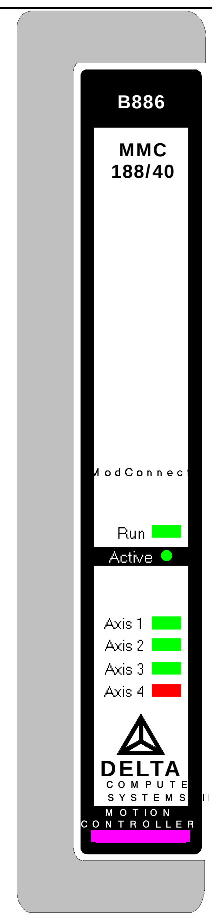
The LEDs labeled "Axis 1" through "Axis 4" indicate whether the input transducers are working. The motion controller module checks the status of the transducer channels every 10 milliseconds. Should a control error occur on an axis other than a transducer error, the LED for the axis in error will flash red repetitively.

NOTE: If a transducer fails to respond, the output to the corresponding axis will be set to the current drive null and its LED will be red. If the axis magnetostrictive transducer comes on line, it will respond to commands and its LED will turn green. The Programmable Controller will see a change in the STATUS bits when either event occurs.

Run indicator constant red	On-board processor halted, output drives disabled
Run indicator constant green	On-board processor running

Active indicator constant red	OURBUS disabled
Active indicator green	OURBUS functional

Axis 1, Axis 2, Axis 3, Axis 4 indicator constant red	Transducer consistently not responding for axis
Axis 1, Axis 2, Axis 3, Axis 4 indicator flashing red-green to orange-green	Program has detected control error for axis (color varies with axis position)
Axis 1, Axis 2, Axis 3, Axis 4 indicator green	Axis transducer working. The intensity is directly proportional to the axis position



MOTION CONTROL PARAMETERS

Following is a list of all the parameters needed to configure and communicate with the MMC 188/40.

Initialization Parameters

DIRECTION (Default: 0) Possible options: 0 and -1 (0FFFF H)

The DIRECTION parameter determines whether the ACTUAL POSITION increases or decreases as the TRANSDUCER COUNTS increase. A zero produces a direct relation between ACTUAL POSITION and TRANSDUCER COUNTS while a -1 (65535 or 0FFFFH) results in an inverse relationship. The ACTUAL POSITION is determined as follows:

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

XOR (exclusive OR) has the effect of multiplying by -1 when DIRECTION = -1 and multiplying by 1 when DIRECTION = 0.

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

NOTE: Extending is always the direction which increases TRANSDUCER COUNTS.

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

SCALE (Default: 32768) Range 0 to 65535

TIP: Use DCSMON commands P0 and P1 to help calculate SCALE, OFFSET and DIRECTION.

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. It can also be thought of as the number of Position Units for every 32768 TRANSDUCER COUNTS.

When using 4 recirculations for the position measurement, the most common Position Unit size is 0.001". With 2 recirculations Position Units are usually 0.002", and with 1 recirculation they are 0.004" or 0.1 mm.

Linear transducers have a calibration number (typically 9.1 microseconds per inch) and the MMC 188/40 has a 27.75 MHz oscillator for transducer position measurement counters. To determine the SCALE for a magnetostrictive transducer use the following formula:

$$\text{SCALE} = \frac{32768}{\text{Position Unit} \times \text{Cal. Number} \times 27.75 \times \text{Number of Recirculations}}$$

The calibration number must be specified in the same base units as the Position Units (inches, mm, etc).

The SCALE parameter can also be adjusted or fine tuned by using position readings from the MMC 188/40 and actual physical measurements in the following formula:

$$\text{SCALE} = \text{Old Scale} \times \frac{\text{Measured Position 1 - Measured Position 2}}{\text{Position Reading 1 - Position Reading 2}}$$

Where Old Scale is the default SCALE or a previously calculated SCALE, and Measured position is in Position Units (thousandth of an inch, etc.) from a physical measurement of the system. Position readings are taken using the ACTUAL POSITION parameter.

See Startup Section for more information on setting the scale.

NOTE: If a negative value is obtained for SCALE, the absolute value must be used and the DIRECTION parameter changed from 0 to -1 or vice versa.

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

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.

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

EXTEND LIMIT (Defaults to ACTUAL POSITION on power-up)

The EXTEND LIMIT specifies the maximum value that the MMC 188/40 will allow as a COMMAND POSITION. (When DIRECTION = -1, this is a minimum value.) REQUESTED POSITIONS that exceed this value will be truncated, and the Parameter Error bit (bit 8) in the STATUS word will be set. The EXTEND LIMIT is given in Position Units. On start-up 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.

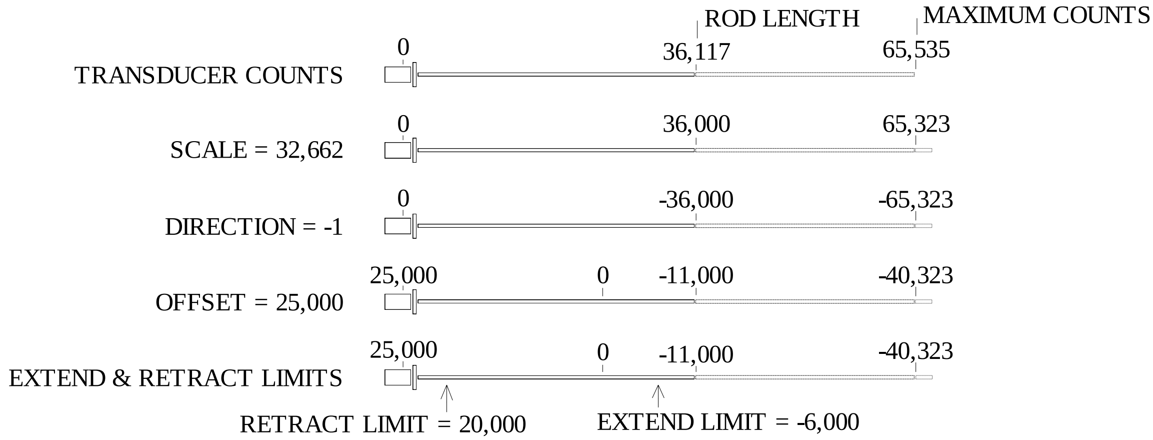
NOTE: The EXTEND LIMIT must be changed when the DIRECTION, SCALE, or OFFSET parameters are changed.

NOTE: Extending is the direction that returns increasing TRANSDUCER COUNTS.

RETRACT LIMIT (Defaults to ACTUAL POSITION on power-up)

The RETRACT LIMIT specifies the minimum value the MMC 188/40 will allow as a COMMAND POSITION. (When DIRECTION = -1, this is a maximum value.) REQUESTED POSITIONS below this value will be truncated. The RETRACT LIMIT is given in Position Units. On start-up the RETRACT 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 RETRACT LIMIT must be changed when the SCALE, OFFSET or DIRECTION parameters are changed.



STATIC GAIN (Default: 50) Range: 0 to 12,500

STATIC GAIN specifies the amount of drive added to the output for a given amount of position error while the axis is at rest. STATIC GAIN is given in number of counts of drive per 100 units of position error. The STATIC GAIN should be set as high as possible without causing the axis to oscillate. If a gain greater than 12500 is entered, the parameter error bit will be set in the STATUS word, and the value will be truncated to 12500. See MAXIMUM POSITION ERROR parameter.

CAUTION: Increase the STATIC GAIN with care as oscillations could damage both man and machine.

EXTEND GAIN (Default: 50) Range: 0 to 12,500

EXTEND GAIN is similar to the STATIC GAIN except it applies when the axis is extending. The gain while moving can usually be a little higher than at rest. This will allow tighter control of the axis while in motion.

RETRACT GAIN (Default: 50) Range: 0 to 12,500

The RETRACT GAIN is similar to the EXTEND GAIN except it is used for retracting only.

NOTE: Retracting is the direction that returns decreasing TRANSDUCER COUNTS.

EXTEND FEED FORWARD (Default: 100)

Feed forward is an open loop compensation which is proportional to the TARGET SPEED of the axis. This value is expressed in terms of counts of drive per 10,000 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:

$$\text{Feed Forward Drive} = \frac{\text{EXTEND FEED FORWARD} \times \text{TARGET SPEED}}{10{,}000}$$

The maximum value that may be used for the EXTEND FEED FORWARD is given by:

$$\text{Maximum Feed Forward} = \frac{2048000}{\text{MAX SPEED}}$$

NOTE: Use the 'F' command to automatically set the feed forward value after the axis has made a complete move without oscillations or output overdrive errors.

RETRACT FEED FORWARD (Default: 100)

Same as EXTEND FEED FORWARD, excepts it is used only when retracting.

NOTE: Retracting is the direction that returns decreasing TRANSDUCER COUNTS.

FEED FORWARD ADVANCE (Default: 0)

Due to inertia and valve response times, an axis will not respond to a change in the drive output until some time has passed. This will cause the error between the ACTUAL POSITION and TARGET POSITION to be larger for axes that are moving large masses or have slow valves. The MMC 188/40 can time shift the feed forward portion of its drive output by the number of milliseconds specified by the FEED FORWARD ADVANCE.

This time shifting is implemented by delaying the TARGET POSITION and TARGET SPEED of the axis with respect to the feed forward drive output. The time delay is specified by the FEED FORWARD ADVANCE. This can compensate for an axis with a slow response time so the target and actual positions track more closely. Always use the minimum amount to get the desired results. One could also call this a target delay time.

NOTE: If the motion on two or more axes is being coordinated, FEED FORWARD ADVANCE should be set the same on those axes.

HYSTERESIS (Default: 0)

HYSTERESIS is used to compensate for drive deadband. Some valves and drives will not react to small changes in output around the null drive value. The HYSTERESIS value is added or subtracted from the drive output, depending on the direction of travel. The drive output will then be outside of the dead band.

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

DITHER (Default: 0)

The DITHER value is the amplitude, in percent of full drive, of a 250 Hz square wave that is superimposed on the normal drive output. A value of 10 will cause a square wave of 10% of full drive to be applied on top of the normal output. This keeps the valve free and improves valve response. Leave this parameter at zero unless the valve manufacturer recommends that DITHER be used. DITHER is applied only when the axis is at rest.

DIFFERENTIAL GAIN (Default: 0)

The DIFFERENTIAL GAIN field is used to control the differentiator. The differentiator should be used on systems that have poor response and a system frequency of 5 Hz or less. The differentiator maintains an exponential average of the *change* in error between the TARGET and ACTUAL POSITIONS. This average is multiplied by the DIFFERENTIAL GAIN value to get the differentiator drive term.

Differential Drive = (Error(t) - Error(t-20)) x DIFFERENTIAL GAIN

INTEGRAL GAIN (Default: 0)

The INTEGRAL GAIN is used to control the amount of drive provided by the integrator while the axis is moving. When a 'G' command is issued, the integrator starts adding the position error to an accumulator every 2 milliseconds. As long as the error is not approaching zero or has not changed sign, it is added to the accumulator. When the axis starts ramping down, the accumulator is decremented at a rate calculated to bring the value to zero when the axis stops. The drive provided by the integrator is given by:

$$\text{Integral Drive} = \frac{\text{INTEGRAL GAIN x Accumulator}}{20{,}000}$$

Note: The integral drive is only applied while the At Commanded Position bit is on.

NULL UPDATE (Default: 500) Range: 10 - 65535, 0 = disabled

NULL UPDATE is the time interval in milliseconds at which the drive null is checked and the value for NULL DRIVE is updated. NULL DRIVE on an ideal system should be 0 Volts or 0 milliAmps; equivalent to 2048 output to the digital-to-analog converter. In a real system, however, a 0 Volt or current output may cause the axis to drift slowly so the real value of NULL DRIVE is something other than 2048.

When the axis is at command position, (not Halted), the ACTUAL POSITION is compared with the TARGET POSITION at the interval specified by the NULL UPDATE. If the ACTUAL POSITION is greater than the TARGET POSITION, the value of NULL DRIVE is decreased by one. If the ACTUAL POSITION is less than the TARGET POSITION, the value of NULL DRIVE is increased by one.

NEW NULL (Default: 0 - use previous null) Range: 1844 to 2252

Sometimes it is desirable to set the internal null value of an axis. The motion controller will set the initial null to 2048 on start-up which should be close to neutral drive. In reality, the true value for null may be slightly different and therefore cause a small bump in the position of the axis. If the real value for null is between 1844 and 2252, the value can be entered here on start-up so the bump in position will not occur. (See NULL UPDATE parameter description.) If a zero (default is used, the current value of null is not canged.

If a value outside the range is used, it will be ignored (set to zero).

MAX POSITION ERROR (Default: 250)

The MAX POSITION ERROR determines how large the difference between the TARGET POSITION and ACTUAL POSITION can get before the Lag Error or Lead Error bits are set in the STATUS word. The value MAX POSITION ERROR is limited by the largest of the three proportional gains:

$$\text{MAX POSITION ERROR} < \frac{204800}{\text{Largest Proportional Gain}}$$

HALT MASK (Default: 0000H - Halt enabled)

The bits in the HALT MASK field have a one-to-one correspondence with the bits in the STATUS word. The bits in this field are used to mask out the automatic halt that occurs when an error bit (2-4, 7-8, and 10) is set in the STATUS word. On start-up, this field is 0 (zero - all bits cleared), and any error will cause a halt. For simple applications this field should be left at zero. In applications where two or more axes are synchronized together using sync bits in the MODE field (page 14), when one axis is halted, all synchronized axes halt with it.

NOTE: In DCSMON use a zero as the first character when entering hexadecimal numbers.

ESTOP MASK (Default: FFFFH - Emergency stop disabled)

Like the HALT MASK, the bits in the ESTOP MASK field have a one-to-one correspondence with the bits in the STATUS word. The bits in this field are used to mask out the emergency stop that occurs when an error bit (2-4, 7-8, and 10) is set in the STATUS word. During an emergency stop, the drive output is immediately set to null and held there until a new command is issued.

NOTE: To enable an emergency stop on an error condition, the appropriate bit must be cleared in both the HALT MASK and the ESTOP MASK.

AT COMMAND POSITION (Default: 50)

AT COMMAND POSITION specifies the size of a window around the COMMAND POSITION. When the TARGET POSITION is equal to the COMMAND POSITION and the ACTUAL POSITION gets within this window, the At Command Position bit is set in the STATUS word.

For example, if an axis COMMANDED POSITION is 10,000, and the AT COMMAND POSITION parameter is 30, the At Command Position bit will be set when the TARGET POSITION is 10,000 and the ACTUAL POSITION is between 9,971 and 10,029. Note that if AT COMMAND POSITION is 0 (zero), the At Command Position bit will never be set.

NEAR COMMAND POSITION (Default: 0)

The NEAR COMMAND POSITION is similar to the AT COMMAND POSITION. The NEAR COMMAND POSITION window should be wider (larger value) than the AT COMMAND POSITION window unless the NEAR COMMAND POSITION is to be ignored, in which case it should be set to 0. This parameter is useful in letting the programmable controller know the axis is out of the way of some other equipment.

Motion Control Commands

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

MODE (Default: 0000H)

5 bits in the MODE word are used to determine the way the MMC 188/40 responds to control commands and parameters.

Bit 16 - Acceleration and Deceleration Mode Select.

When this bit is set, ACCELERATION and DECELERATION are given in ramp rates. when the bit is cleared, ACCELERATION and DECELERATION are given in ramp lengths.

Bit 14 - Unipolar Mode.

When this bit is set, the analog output from the MMC 188/40 is restricted to positive values only. i.e. the absolute value of the output is used.

Bit 13 - Simulation Mode.

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 purposes.

Bit 12 - SyncA (Synchronization bit A).

When this bit is set, it causes the axis to move in a synchronized manner with other axes on the module that also have the bit set. The axes are synchronized as follows:

a) The axis with the longest specified move is the master
b) The profile the remaining axes follow is determined by the length of their move relative to the master axis. All axes will accelerate together, move at constant speed together and decelerate together, arriving at their COMMAND POSITION simultaneously.
c) If any axis is halted by an error condition, all axes will halt together.

Bit 11 - SyncB (Synchronization bit B)

This bit has the same effect as bit 12 above. It is used if a second pair of axes must be synchronized on the same module.

ACCELERATION (Default: 1000)

This field determines the rate at which the axis will accelerate. The field can be interpreted in two ways depending on bit 16 in the MODE field. If the bit is cleared (zero), the value in this field is a distance expressed in Position Units. If the SCALE were set so one Position Unit equaled .001", then a value of 5000 would represent a ramp of 5.000".

If bit 16 in the MODE field is set (one), this field is interpreted as acceleration rate and is expressed in 1000 Position Units/sec/sec. If SCALE were set so one Position Unit equaled .001", then a value of 200 would represent an acceleration rate of 200 inches/sec/sec.

Using ramp length is simpler for applications where the maximum speed does not change. The relationship between the ramp length, ACCELERATION and MAXIMUM SPEED is:

$$\text{Ramp Length} = \frac{\text{MAXIMUM SPEED x MAXIMUM SPEED}}{\text{Acceleration Rate x 2000}}$$

$$\text{Acceleration Rate} = \frac{\text{MAXIMUM SPEED x MAXIMUM SPEED}}{\text{Ramp Length x 2000}}$$

DECELERATION (Default: 1000)

This field is similar to the ACCELERATION field except it specifies the deceleration ramp length or deceleration rate.

MAXIMUM SPEED (Default: 1000)

The MAXIMUM SPEED parameter sets the constant speed to be achieved after acceleration. The maximum speed is expressed in Position Units/second. If the SCALE is set so one position unit equals .001", a maximum speed of 25 inches per second is expressed as 25000.

NOTE: When bit 16 in the MODE field is cleared, changing the MAXIMUM SPEED without changing the ACCELERATION and DECELERATION distances will change the acceleration and deceleration rates.

NOTE: Should the MAXIMUM SPEED be set to zero, the axis will do the same as a HALT, except that the halt bit will not be set.

REQUESTED POSITION

The Programmable Controller sets this to the position the axis is to move to. This value is bounds checked using the EXTEND and RETRACT LIMITS and then placed in the COMMAND POSITION field.

COMMAND

The COMMAND word tells the MMC 188/40 what to do and what information to return to the 984 Programmable Controller. See the section on COMMUNICATING WITH THE MMC 188/40 for details on valid commands.

Readback Parameters

ACTUAL POSITION

The ACTUAL POSITION is the measured position of the axis at any moment. This position is updated every two milliseconds. The ACTUAL POSITION is obtained from the TRANSDUCER COUNTS with the following formula:

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

XOR has the effect of multiplying by -1 when DIRECTION = -1 and multiplying by 1 when DIRECTION = 0.

STATUS

The axis STATUS word contains 16 bits of information about the condition of the axis. Six error bits (2-4, 7-8, and 10) can be used to trigger a halt or emergency stop on the axis.

The error bits 2-5, 8, 10 and 14-16 are cleared whenever a 'G' command is given.

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

Bit 1 = Parameters Initialized

This bit is set after a Set Parameter ('P') command is successfully executed. Until this bit is set, the axis will not respond to any Go ('G') commands. This bit is cleared whenever the module is reset.

Bit 2 = Lag Error

Lag is a condition where the ACTUAL POSITION falls behind the TARGET POSITION. This bit will be set when the ACTUAL POSITION lags the TARGET POSITION by more than the amount specified by the MAXIMUM ERROR parameter. The LAG ERROR bit is latched and is cleared when a new command is issued.

Bit 3 = Lead Error

Lead is a condition where the ACTUAL POSITION gets ahead of the TARGET POSITION during a move. This bit will be set when the ACTUAL POSITION leads the TARGET POSITION by more than the amount specified by the MAXIMUM ERROR parameter. The Lead Error bit is latched and is cleared when a new command is issued.

Bit 4 = Overdrive

This bit is set when the calculated drive output would exceed the 12 bit range of the A/D converter. Usually this error is an indication the system does not have enough power to drive the axis at the requested speed. The computer will truncate the drive to 12 bits. The OVERDRIVE bit is latched and is cleared when a new command is issued.

Bit 5 = Valve Out Of Null

Normally, when the drive output is zero volts or zero mA, the axis will not move. In hydraulic applications valves sometimes get dirty and the axis will tend to move even when the drive output is 0. The motion controller module will try to compensate for this problem, but if more than 10 percent of minimum or maximum drive output is required to hold a static position this error bit will be set.

Bit 6 = Transducer Not Responding

This bit will be set and the corresponding front panel axis LED (light emitting diode) will glow red if the transducer does not respond with valid data at least once every 10 milliseconds. The data received is compared with previous data. If the two readings differ by more than 500 counts, the new reading is assumed to be an error. When the Transducer Not Responding bit is set, an emergency stop is made. This bit will be on for at least 50 milliseconds or until the problem is corrected. When the bit is cleared, the module will respond to the next command.

CAUTION: If the transducer is not responding, an emergency stop is made.

Bit 7 = Position Overflow

This bit is set when the ACTUAL POSITION is about to overflow its 16 bit limit. This occurs when the ACTUAL POSITION tries to go beyond 65500 counts. The Position Overflow bit is latched and is cleared when a new command is issued.

Bit 8 = Parameter Error

This bit will be 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 hard and fast limits may not always be available. However, the motion controller does try to replace the offending value with another that is within range. By comparing the values written with the values after the error bit was set, the offending parameter can be determined. This bit is cleared when a new command is issued.

Bit 9 = Active

This bit will be set after a valid command or Status Area Request is received by the MMC 188/40 from the Programmable Controller. It will remain on until the next time the I/O drop is accessed by the Programmable Controller.

Bit 10 = Stopped

This bit is set when the average speed of the axis is less than 2000 TRANSDUCER COUNTS per second. It is intended for use as an axis obstruction indication. The Stopped bit is not latched.

Bit 11 = Decelerating

This bit will be set while the axis is decelerating.

Bit 12 = At Max Speed

This bit is set while the axis is moving at the speed specified by the MAXIMUM SPEED parameter.

Bit 13 = Accelerating

This bit will be set while the axis is accelerating.

Bit 14 = Halted

This bit is set when the axis has been given a Halt ('H') command or an error has caused a halt and the axis has come to a stop. While the axis is halted, the null update timer will not run. The Halted bit is cleared when a new command is issued to the axis.

NOTE: If the MMC 188/40 is left on while the axis power is off or the drive output is disconnected, a Halt command should be issued to keep the null value from getting very far off.

Bit 15 = Near Command Position

This bit is similar to the At Command Position bit except that it gets set before or at the same time as the At Command Position bit - never later. The Near Command Position bit is latched and is only cleared by a 'G' command. If the axis is near the COMMAND POSITION when the 'G' command is given, the bit may be set again before the Programmable Controller can detect it.

See NEAR COMMAND POSITION parameter.

Bit 16 = At Command Position

This bit will be set when the difference between the ACTUAL POSITION and COMMAND POSITION is less than the value in the AT COMMAND POSITION field and the TARGET POSITION is equal to the COMMAND POSITION. This bit is latched and is only cleared by a 'G' command. If the axis is near the COMMAND POSITION when the 'G' command is given, the bit may be set again before the Programmable Controller can detect it.

See AT COMMAND POSITION parameter.

NOTE: This bit will not be set if the axis is Halted outside the AT COMMAND POSITION window.

COMMAND POSITION

The COMMAND POSITION is the REQUESTED POSITION with bounds checking applied. If the REQUESTED POSITION is outside the RETRACT or EXTEND LIMIT, the COMMAND POSITION will be truncated to the value of the limit, and the axis will go only to the limit. The COMMAND POSITION is updated when a Move command is issued using the COMMAND parameter.

TARGET POSITION

The TARGET POSITION is the calculated position where the axis should be at any moment. When the axis is moving, the TARGET POSITION is updated every two milliseconds as it moves toward the COMMAND POSITION.

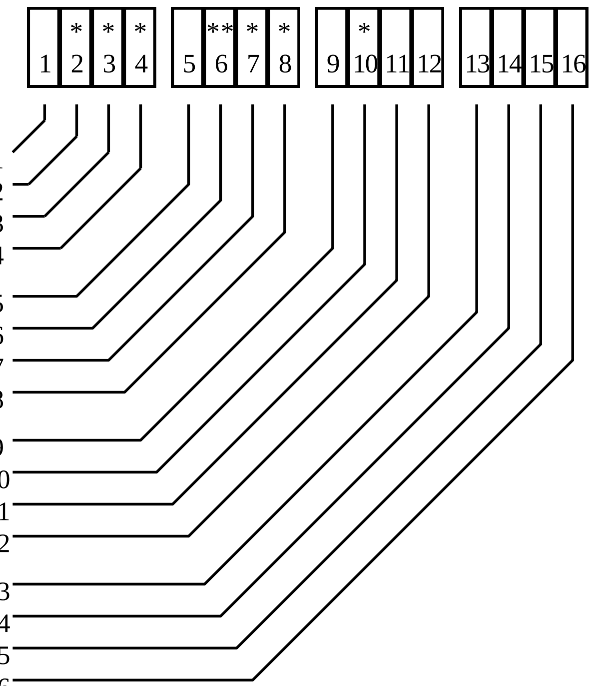
NOTE: When an axis is stopped, the TARGET POSITION should be the same as the COMMAND POSITION unless an error has occurred.

STATUS Word Bit Map

The axis STATUS word contains 16 bits of information about the status of the axis. The hexadecimal table provides an easy method to convert hexadecimal numbers to bit patterns.

Hexadecimal To Binary Conversion Table

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
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 Definitions

Parameters Initialized		1
Lag Error	*	2
Lead Error	*	3
Overdrive	*	4
Valve Out Of Null		5
Transducer Not Responding	**	6
Position Overflow	*	7
Parameter Error	*	8
Active		9
Stopped	*	10
Decelerating		11
At Max Speed		12
Accelerating		13
Halted		14
Near Command Position		15
At Command Position		16

* Can cause a halt or emergency stop if the corresponding bits are clear in the appropriate mask register.

** Will cause an emergency stop no matter what the mask register is set to.

TRANSDUCER COUNTS

TRANSDUCER COUNTS is the axis position read directly from the transducer counters with no scaling.

TARGET SPEED

The TARGET SPEED is the calculated speed at which the axis should be moving at any point in time.

DRIVE

DRIVE is the output to the actuator. The 12 bit digital value output - 0 (full negative drive) to 4095 (full positive drive) - will cause a ±10 Volt, ±5 Volt, ±2.5 Volt, ±100 milliAmp, ±50 milliAmp or ±25 milliAmp output. Null drive should be about 2048. Note that there aren't any pots to adjust to electronically null the output. All this is done in software (see NEW NULL and NULL UPDATE).

SETUP CHECKLIST

Wiring

Connect inputs (use AMP/10 where necessary).

Connect utputs (+out = axis extends, correct load).

Connect power supplies.

Insert module keys in rack.

MMC 188/40 configuration

Set interrogation pulse polarity.

Set number of recirculations.

Set drive output mode (current, voltage) and level

Installation

Define inputs and outputs.

Traffic cop the MMC 188/40.

Reserve memory for the module.

Install Custom Loadable Function Blocks.

Start-up and Tuning

Run DCSMON.

Set EXTEND and RETRACT LIMITS.

Tune axes.

Save parameters.

Read parameters into 984.

PREPARING FOR INSTALLATION

Wiring Notes

NOTE: When wiring the system, it is important that the drive extends when a positive voltage or current is sent to the drive. The extend direction is defined as the direction that causes the transducer to return increasing counts. The extend direction on a magnetostrictive transducer is away from the transmitting end.

NOTE: If the transducer is not supplied with a 200 ohm termination resistor installed between the interrogation pin and common, it may be necessary to install one as close to the transducer as possible to reduce electrical noise in the system -- see wiring diagram.

NOTE: Use differential drivers (AMP/10 cards from DELTA) when running transducer cables over 100 feet or in noisy environments.

NOTE: Use shielded twisted pairs for all connections to inputs and outputs.

NOTE: If possible, connect the transducer's ±15 volts directly to the MMC 188/40. The module provides current limiting resistors to protect the transducer.

System Notes

Hydraulic systems must have enough pressure and fluid volume (accumulator) to move the desired load. 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 is capable.

There should be no flexible hose between the valve and the cylinder being controlled in hydraulic applications. The hose swells and contracts as the valve opens and closes causing loss of control and oscillations.

Cable & Connector Information

Cabling from the MMC 188/40 transducer inputs and servo drive outputs to the terminal blocks are included with the module. The wire information below has wire color code data for the Temposonics brand of magnetostrictive transducers and cables supplied by DELTA Computer Systems Inc.

Delta Computer Systems, Inc. wire color codes. (PART NO MCCBS-02) DB25 to transducer.

Pin	Function	Color
1	+15 input	RED
2	power supply common	BLACK
3	-15 input	WHITE
4	+5 input	GREEN
5	+12 output	ORANGE
6	common	GRAY
7	interrogation pulse 1	BROWN
8	+15v axis 1	PINK
9	return pulse 1	YELLOW
10	-15v axis 1	VIOLET
11	common	TAN
12	interrogation pulse 2	BLUE
13	+15v axis 2	RED/BLACK

14	return pulse 2	RED/YELLOW
15	-15v axis 2	RED/GREEN
16	common	WHITE/BLACK
17	interrogation pulse 3	WHITE/BLUE
18	+15v axis 3	WHITE/RED
19	return pulse 3	WHITE/YELLOW
20	-15v axis 3	WHITE/GREEN
21	common	WHITE/GRAY
22	interrogation pulse 4	WHITE/BROWN
23	+15v axis 4	WHITE/ORANGE
24	return pulse 4	WHITE/BLACK/RED
25	-15v axis 4	WHITE/VIOLET

Typical Temposonics I Connector

MS310614A-P Connector Pins	Function	Color
A	+12 to +15 Vdc	Green
B	DC Common	Black
C	Return Pulse	Orange
D	-13.5 to -15 vdc	Blue
E	Interrogation Pulse	White
F	+11.5 to +12 vdc	Red

DB15 to valves or motor controller. PART NO MCCS-01.

Pin	Function	Wire Color
1	+15 input	RED
2	power supply common	BLACK
3	-15 input	WHITE
4	common	GREEN
5	drive output 1	ORANGE
6	common	BLUE
7	common	BROWN
8	drive output 2	YELLOW
9	common	RED/BLACK
10	drive output 4	RED/YELLOW
11	common	RED/GREEN
12	common	TAN
13	drive output 3	PINK
14	common	GRAY
15	common	VIOLET

Power Supply Requirements

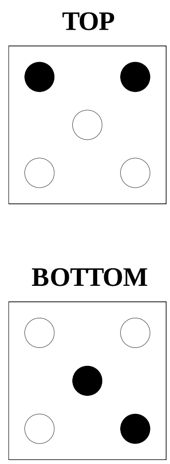
5 Volt and ±15 Volt power supplies must be provided for the transducer inputs and drive outputs. The power supplies should provide at least 500 mA and should not be connected to any other loads. This precaution will separate the MODICON system from any noise or electrical faults on the wires going to the transducers or drives. We also recommend using separate ±15V power supplies for the transducer and drive sections.

Module Keying

The keying pattern on the 800 series I/O rack must be as shown in the following figure to match the keying on the MMC 188/40.

The 800 Series I/O Rack

Mating Key Locations



When facing the rack, pins are placed into the holes indicated by black circles

CONFIGURING AND INSTALLING THE MODULE

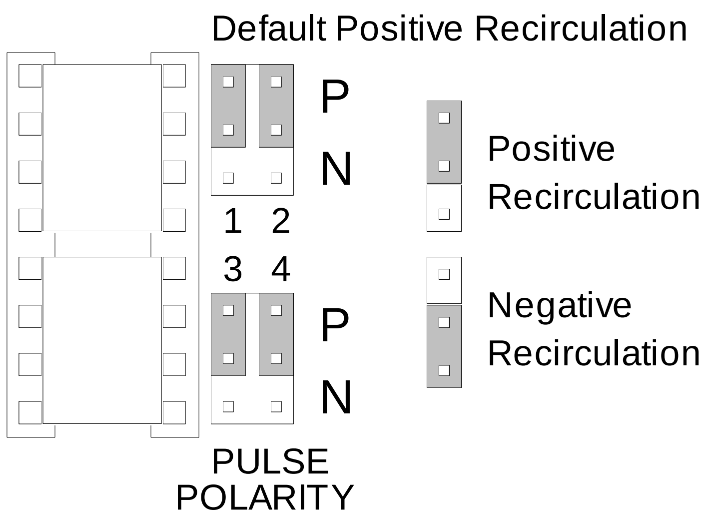
Magnetostrictive Transducer Interrogation Pulse Polarity

Jumpers P9-P12 are used to select the polarity of the transducer transmit Interrogation pulse. The motion controller is shipped with all four axes configured for positive interrogation pulses. Transducers can be ordered with either positive or negative interrogation pulses. Usually, the 1 foot transducers or shorter come with negative going interrogation pulses. **We recommend ordering only positive transmit pulses for consistency**.

NOTE: If the MMC 188/40 is configured for a positive interrogation pulse and is connected to a transducer requiring a negative interrogation pulse or vice versa, current limiting resistors on the MMC 188/40 will burn out. The current limiting resistors protect the transducer and are field replaceable.

Jumpers P9 - P12

P9	1-2	axis 1 positive pulse	P11	1-2	axis 3 positive pulse
P9	2-3	axis 1 negative pulse	P11	2-3	axis 3 negative pulse
P10	1-2	axis 2 positive pulse	P12	1-2	axis 4 positive pulse
P10	2-3	axis 2 negative pulse	P12	2-3	axis 4 negative pulse



Number of Recirculations

Recirculation is a term used with magnetostrictive transducers (Temposonics) to indicate how many times the transducer is interrogated for a single position measurement. If the number of recirculations is set to 1, only one interrogation pulse is issued, if set to four, then four interrogation pulses are issued.

The tradeoffs are measurement time, resolution, and maximum transducer length. The higher the number of recirculations, the higher the resolution, the longer the measurement time, and the shorter the maximum transducer length. The lower the number of recirculations, the lower the resolution, the shorter the measurement time, and the longer the maximum transducer length. See tables below.

$$\text{Counts Per Inch} = \frac{\text{Calibration Number (9.1 us Typical)}}{\text{36 ns per clock (27.75 MHz clk)}} \times \text{Recirculations}$$

$$\text{Maximum Transducer Length} = \frac{65535}{\text{Counts per Inch}}$$

Longest Measurement Time = Transducer Length (Inch) x Cal. Number x Recirculations

$$\text{Maximum Resolution} = \frac{1}{\text{Counts per Inch}}$$

Number of Recirculations	Maximum Transducer Length	Measurement Time @60"	Maximum Usable Resolution
1	260"	546 us	.004"
2	130"	1092 us	.002"
4	65"	2184 us**	.001"

$$\text{Counts per mm} = \frac{\text{Calibration Number (358.27 ns/mm typical)}}{\text{36 ns per clock (27.75 MHz clk)}} \times \text{Recirculations}$$

$$\text{Maximum Linear Transducer Length} = \frac{65535}{\text{Counts per mm}}$$

Longest Measurement Time = Transducer Length (mm) x Cal. Number x Recirculations

$$\text{Maximum Resolution} = \frac{1}{\text{Counts per mm}}$$

Number of Recirculations	Maximum Transducer Length	Measurement Time @ 1.5M	Maximum Usable Resolution
1	6586 mm	537 us	.100 mm
2	3293 mm	1075 us	.05 mm
4	1647 mm	2150 us**	.025 mm

**** NOTE: It is best if the longest measurement time does not exceed 2000 microseconds. The data used by the MMC 188/40 must be available within its two millisecond control loop for precise control. If more than 2000 microseconds are required to obtain a position measurement, the controller will slow down to a 4 millisecond loop time, and a slight bump will occur in the motion of the axes at the transition point.**

Jumpers P5-P8 (Default 4 Recirculations)

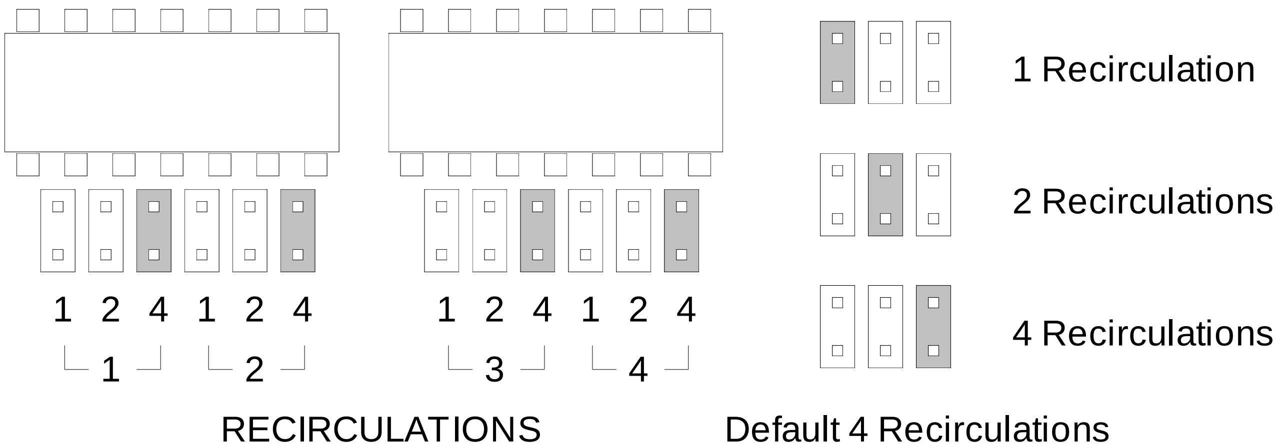
Jumper P5 for axis 1

Jumper P6 for axis 2

Jumper P7 for axis 3

Jumper P8 for axis 4

Jumper Pins	Recirculation Selected
1-2	1
3-4	2
5-6	4



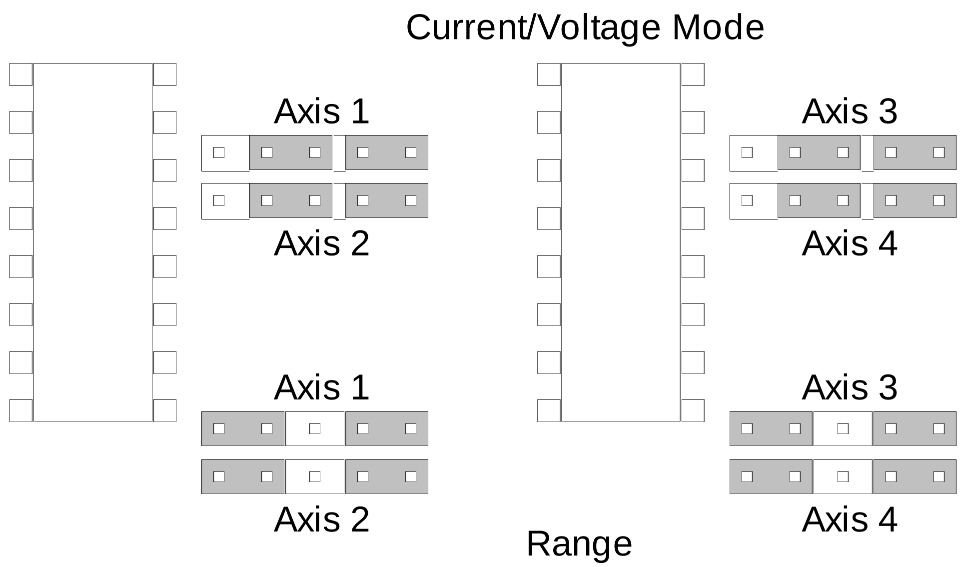
Drive Output Modes

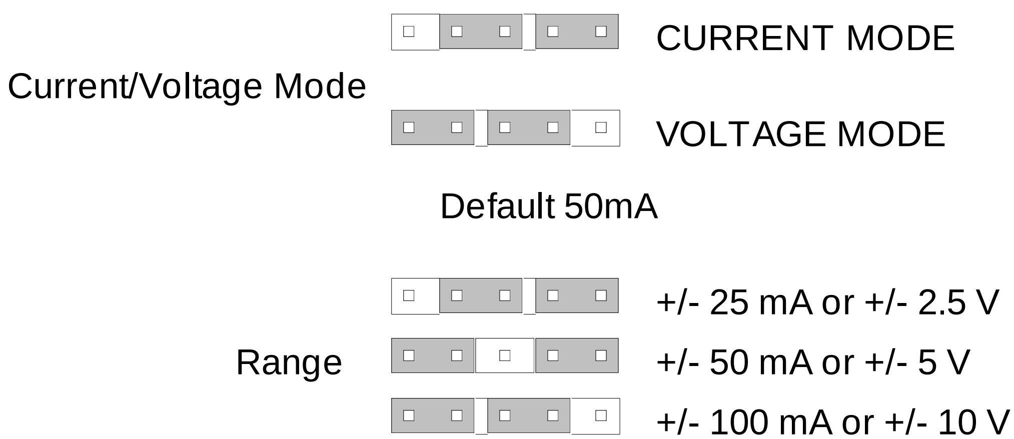
The MMC 188/40 can be configured for either current or voltage output at 3 levels. Jumpers P13-P14 and P17-P18 select the mode, and P15-P16 and P19-P20 select the level.

The levels available are:

Current mode (milliAmps)	Voltage mode (Volts)
±100	±10
±50	±5
±25	±2.5

See the following figure for selecting configurations.


Current/Voltage Mode
Axis 1
Axis 2
Axis 3
Axis 4
Axis 1
Axis 2
Axis 3
Axis 4
Range


Current/Voltage Mode
CURRENT MODE
VOLTAGE MODE
Default 50mA
Range
+/- 25 mA or +/- 2.5 V
+/- 50 mA or +/- 5 V
+/- 100 mA or +/- 10 V

COMMUNICATING WITH THE MMC 188/40

OURBUS Setup

Select eight 3XXXX registers for inputs and eight 4XXXX registers for output for each module.

The MMC 188/40 must be traffic copped as a B886, 8 register, bidirectional I/O module with binary format.

In the remainder of this chapter 3TTTT represents the base address for the input registers, and 4TTTT represents the base address for the output registers. These are the addresses at which the MMC 188/40 is traffic copped.

Memory Requirements

Unless the EEPROM option is installed, each axis requires 24 words of memory in the 984 Programmable Controller. This memory contains the initialization parameters for the axis. The blocks of 24 must be placed consecutively in memory for the axes used on each module. If only axes 1 through 3 are used, only 3 blocks need to be allocated. If axes 1, 3 and 4 are used (skipping axis 2), then all 4 blocks must be allocated. The parameters must be arranged in the following order:

0 NEW NULL
1 ESTOP MASK
2 HALT MASK
3 Reserved (Zero)
4 FEED FORWARD ADVANCE
5 NULL UPDATE
6 Reserved (Zero)
7 DITHER
8 HYSTERESIS
9 PROPORTIONAL STATIC GAIN
10 PROPORTIONAL EXTEND GAIN
11 PROPORTIONAL RETRACT GAIN
12 INTEGRAL GAIN
13 DIFFERENTIAL GAIN
14 EXTEND FEED FORWARD
15 RETRACT FEED FORWARD
16 SCALE
17 OFFSET
18 DIRECTION
19 MAX ERROR
20 AT COMMAND POSITION
21 NEAR COMMAND POSITION
22 EXTEND LIMIT
23 RETRACT LIMIT

If multiple MMC 188/40 modules are used, each module in the system must have its own block of memory allocated for parameter storage.

In addition to the parameter storage blocks, sufficient memory must be allocated for profile storage when EEPROM is not installed. Each profile requires 4 words of memory, and each board can use up to 16 profiles. Only enough memory must be allocated to hold the profiles used in all the modules in the system. The profiles are stored in the following order:

Profile 1 MODE

ACCELERATION

DECELERATION

MAX SPEED

Profile 2 MODE

ACCELERATION

DECELERATION

MAX SPEED

Profile 3 MODE

:

:

And so on.

Installing Custom Loadable Function Block FN10

To install the custom loadable function block follow these steps:

1 Load the Programmable Controller ladder logic program into the PC-AT.

2 Go to *Offline/Config*.

3 Go to *Loadable*.

4 Go to *Dir/Load*.

5 Enter a:\FN10.DAT (assuming the source diskette is in drive A).

6 Go to *Edit/Insert*.

7 Verify that the opcode (5F) does not conflict with an existing opcode.

8 Return to the main menu and download the program with the new configuration to the Programmable Controller.

In order to have access to the zoom function with the FN10 custom loadable when running Modsoft, copy the file dxfdt.sys from the source diskette to the \modsoft\runtime directory. Be sure to rename the dxfdt.sys file that was already in the \modsoft\runtime directory so it isn't lost.

For more information on custom loadables consult your software manuals.

Output Registers

Eight 16-bit registers are sent to the MMC 188/40 each time the I/O drop is accessed. The first register is always a command register, and its contents determines what the remaining 7 registers contain. They may contain either data or additional commands.

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.

Format A commands have a single command in register 1 and data in the remaining 7 registers. Format B commands alternate a command register with a data register.

Register Number	Format A	Format B
4TTTT + 0	Command	Command
4TTTT + 1	Data	Data
4TTTT + 2	Data	Command
4TTTT + 3	Data	Data
4TTTT + 4	Data	Command
4TTTT + 5	Data	Data
4TTTT + 6	Data	Command
4TTTT + 7	Data	Data

Each command in format B acts on one axis. Various command types described below can be sent to different axes simultaneously as long as they are format B commands.

Command registers are divided into 2 bytes. The Most Significant Byte (MSB - bits 1 - 8) specifies the readback mode; it is called the Readback byte. The Least Significant Byte (LSB - bits 9 - 16) specifies the instruction the MMC 188/40 is to follow; it is called the Instruction byte.

The two bytes in the command register are divided into 4-bit nibbles. 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 the two nibbles are Status Area Request (SAR) fields indicating what status information the MMC 188/40 is to return. (See Input Register section.)

COMMAND REGISTER

	READBACK								INSTRUCTION							
	SAR1				SAR0				TYPE				INDEX			
Bit #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Command Types

The Commands available for controlling the MMC 188/40 are:

Command Type	Binary Representation	Output Register Format	Description
0	0000	A	Single and synchronous moves
1	0001	B	Go using profile
2	0010	A	Set profiles
3	0011	A	Set parameters
4	0100	B	ASCII commands
5	0101	B	ASCII commands
6	0110		Reserved
7	0111		Reserved
8	1000		Reserved
9	1001		Reserved
A	1010	A	Get profiles
B	1011	A	Get parameters

C	1100	Reserved
D	1101	Reserved
E	1110	Reserved
F	1111	Reserved

In the following discussion, an **X** represents a bit whose value doesn't 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 0 - Synchronous and Single Axis Move (0XXX XXXX 0000 NNNN).

Format A - Single Command

The Synchronous and Single Axis Move commands use the Index field as a bit pattern for selecting the axes to move. Each bit in the index field corresponds to an axis. If only 1 bit is set in this field, the command is a "Single Axis Move" command. The purpose of this command is to allow changing position without using a previously defined profile.

If 2 or more bits are set in the index field, those axes will move synchronously. The synchronous commands allow changing the ACCELERATION, DECELERATION AND MAX SPEED fields for all selected axes. This command first sets the SyncA bit (bit 4) in the MODE field of the first selected axis. Next, the MODE field is copied to the MODE field of the other selected axes. This ensures that all the axes are in the same mode. The ACCELERATION, DECELERATION AND MAX SPEED used for that move are ratioed by the fraction of distance the axis must travel relative to the largest travel distance requested. Travel distance is the distance between the REQUESTED POSITION and the current TARGET POSITION.

NOTE: The mode used to specify the ACCELERATION and DECELERATION parameters is determined by the mode that the first axis specified was in during its previous move.

If the requested position for any axis selected is outside the travel limits then a parameter error bit will be set and none of the axes will move.

The Command Register format is shown below:

```
              BIT #
           |    | 111|1111
       1234|5678|9012|3456
       -------------------
HEX    SAR1|SAR0|CMND|INDX
VALUE  -------------------
XX00   0XXX|XXXX|0000|0000 - USE AS A GLOBAL DO-NOTHING COMMAND
XX01   0XXX|XXXX|0000|0001 - Go Axis 1        Format A - Single axis
XX02   0XXX|XXXX|0000|0010 - Go Axis 2        Format A - Single axis
XX04   0XXX|XXXX|0000|0100 - Go Axis 3        Format A - Single axis
XX08   0XXX|XXXX|0000|1000 - Go Axis 4        Format A - Single axis

XX03   0XXX|XXXX|0000|0011 - Go Axes 1,2      Format B - Multi Axes
XX07   0XXX|XXXX|0000|0111 - Go Axes 1,2,3    Format B - Multi Axes
XX0F   0XXX|XXXX|0000|1111 - Go Axes 1,2,3,4  Format B - Multi Axes
XX0B   0XXX|XXXX|0000|1011 - Go Axes 1,2,  4  Format B - Multi Axes
XX05   0XXX|XXXX|0000|0101 - Go Axis 1,  3    Format B - Multi Axes
XX0D   0XXX|XXXX|0000|1101 - Go Axis 1,  3,4  Format B - Multi Axes
XX09   0XXX|XXXX|0000|1001 - Go Axis 1,    4  Format B - Multi Axes
XX06   0XXX|XXXX|0000|0110 - Go Axes   2,3    Format B - Multi Axes
XX0E   0XXX|XXXX|0000|1110 - Go Axes   2,3,4  Format B - Multi Axes
XX0A   0XXX|XXXX|0000|1010 - Go Axes   2,  4  Format B - Multi Axes
XX0C   0XXX|XXXX|0000|1100 - Go Axes     3,4  Format B - Multi Axes
```

The content of the 8 output registers is defined as follows:

```
Format A - Single axis  4TTTT+0 Command (defined above)
                        4TTTT+1 MODE
                        4TTTT+2 ACCELERATION
                        4TTTT+3 DECELERATION
                        4TTTT+4 MAX SPEED
                        4TTTT+5 REQUESTED POSITION
                        4TTTT+6 Don't Care
                        4TTTT+7 Don't Care

Format B - Multi Axes   4TTTT+0 Command
                        4TTTT+1 ACCELERATION
                        4TTTT+2 DECELERATION
                        4TTTT+3 MAX SPEED
                        4TTTT+4 Axis 1 REQUESTED POSITION
                        4TTTT+5 Axis 2 REQUESTED POSITION
                        4TTTT+6 Axis 3 REQUESTED POSITION
                        4TTTT+7 Axis 4 REQUESTED POSITION
```

COMMAND TYPE 1 - Go Using Profile (0XXX XXXX 0001 NNNN).

Format B - 4 Commands.

The Go Using Profile command allows the 984 to tell the MMC 188/40 to move any number of axes independently or synchronously using pre-stored profiles. These commands are the most commonly used for simple applications. If the MODE register associated with the profile selected has a SyncA or SyncB bit set, the axes will be synchronized, otherwise they will move independently. The Go Using Profile command sets the Active bit and clears the At Command Position bit in the STATUS until the next time the drop is accessed.

The format of the Command Register is given below:

```
               BIT #
            |    | 111|1111
        1234|5678|9012|3456
        -------------------
HEX     SAR1|SAR0|CMND|INDX
VALUE   -------------------
XX10    0XXX|XXXX|0001|0000  - GO USING PROFILE  0
XX11    0XXX|XXXX|0001|0001  - GO USING PROFILE  1
XX12    0XXX|XXXX|0001|0010  - GO USING PROFILE  2
XX13    0XXX|XXXX|0001|0011  - GO USING PROFILE  3
XX14    0XXX|XXXX|0001|0100  - GO USING PROFILE  4
XX15    0XXX|XXXX|0001|0101  - GO USING PROFILE  5
XX16    0XXX|XXXX|0001|0110  - GO USING PROFILE  6
XX17    0XXX|XXXX|0001|0111  - GO USING PROFILE  7
XX18    0XXX|XXXX|0001|1000  - GO USING PROFILE  8
XX19    0XXX|XXXX|0001|1001  - GO USING PROFILE  9
XX1A    0XXX|XXXX|0001|1010  - GO USING PROFILE 10
XX1B    0XXX|XXXX|0001|1011  - GO USING PROFILE 11
XX1C    0XXX|XXXX|0001|1100  - GO USING PROFILE 12
XX1D    0XXX|XXXX|0001|1101  - GO USING PROFILE 13
XX1E    0XXX|XXXX|0001|1110  - GO USING PROFILE 14
XX1F    0XXX|XXXX|0001|1111  - GO USING PROFILE 15
```

The 8 output registers are defined as follows:

```
4TTTT+0  Axis 1 Command
4TTTT+1  Axis 1 REQUESTED POSITION
4TTTT+2  Axis 2 Command
4TTTT+3  Axis 2 REQUESTED POSITION
4TTTT+4  Axis 3 Command
4TTTT+5  Axis 3 REQUESTED POSITION
4TTTT+6  Axis 4 Command
4TTTT+7  Axis 4 REQUESTED POSITION
```

where the Command for all 4 axes is either one of the ones listed above or an 'N' command (0XXX XXXX 0100 1110 - XX4EH).

COMMAND TYPE 2 - Set Profiles (0XXX XXXX 0010 NNNN).

Format A - Single Command

These commands allow the programmer to change motion profiles stored in the MMC 188/40. Only 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.

```
               BIT #
            |    | 111|1111
        1234|5678|9012|3456
        -------------------
HEX     SAR1|SAR0|CMND|INDX
VALUE   -------------------
XX20    0XXX|XXXX|0010|0000  - SET PROFILE  0
XX21    0XXX|XXXX|0010|0001  - SET PROFILE  1
XX22    0XXX|XXXX|0010|0010  - SET PROFILE  2
XX23    0XXX|XXXX|0010|0011  - SET PROFILE  3
XX24    0XXX|XXXX|0010|0100  - SET PROFILE  4
XX25    0XXX|XXXX|0010|0101  - SET PROFILE  5
XX26    0XXX|XXXX|0010|0110  - SET PROFILE  6
XX27    0XXX|XXXX|0010|0111  - SET PROFILE  7
XX28    0XXX|XXXX|0010|1000  - SET PROFILE  8
XX29    0XXX|XXXX|0010|1001  - SET PROFILE  9
XX2A    0XXX|XXXX|0010|1010  - SET PROFILE 10
XX2B    0XXX|XXXX|0010|1011  - SET PROFILE 11
XX2C    0XXX|XXXX|0010|1100  - SET PROFILE 12
XX2D    0XXX|XXXX|0010|1101  - SET PROFILE 13
XX2E    0XXX|XXXX|0010|1110  - SET PROFILE 14
XX2F    0XXX|XXXX|0010|1111  - SET PROFILE 15
```

The 8 output registers are defined as follows:

```
4TTTT+0  Command
4TTTT+1  Mode
4TTTT+2  Accel
4TTTT+3  Decel
4TTTT+4  Speed
4TTTT+5  Reserved, (0)
4TTTT+6  Reserved, (0)
4TTTT+7  Reserved, (0)
```

COMMAND TYPE 3 - Set Parameters (0XXX XXXX 0011 NNNN).

Format A - Single Command.

These commands allow the 984 Programmable Controller to download new initialization parameters to the MMC 188/40. New parameters can be downloaded at any time, but they do not take effect until a 'P' command is issued (command type 5). Set Parameter commands are defined as follows:

```
              BIT #
         |    | 111|1111
        1234|5678|9012|3456
        -------------------
HEX     SAR1|SAR0|CMND|INDX
VALUE   -------------------  ; SET PARAMS AXIS 1 GROUP 1
XX30    0XXX|XXXX|0011|0000  4TTTT+0 XXXX|XXXX|0011|0000
                             4TTTT+1 Reserved
                             4TTTT+2 Reserved
                             4TTTT+3 Reserved
                             4TTTT+4 Reserved
                             4TTTT+5 NEW NULL
                             4TTTT+6 EMERGENCY STOP MASK
                             4TTTT+7 HALT MASK

                             ; SET PARAMS AXIS 1 GROUP 2
XX31    0XXX|XXXX|0011|0001  4TTTT+0 XXXX|XXXX|0011|0001
                             4TTTT+1 Reserved
                             4TTTT+2 FEED FORWARD ADVANCE
                             4TTTT+3 NULL UPDATE
                             4TTTT+4 Reserved
                             4TTTT+5 DITHER
                             4TTTT+6 HYSTERESIS
                             4TTTT+7 PRO STATIC GAIN

                             ; SET PARAMS AXIS 1 GROUP 3
XX32    0XXX|XXXX|0011|0010  4TTTT+0 XXXX|XXXX|0011|0010
                             4TTTT+1 PRO EXTEND GAIN
                             4TTTT+2 PRO RETRACT GAIN
                             4TTTT+3 INTEGRAL GAIN
                             4TTTT+4 DIFFERENTIAL GAIN
                             4TTTT+5 EXTEND FEED FORWARD
                             4TTTT+6 RETRACT FEED FORWARD
                             4TTTT+7 SCALE

                             ; SET PARAMS AXIS 1 GROUP 4
XX33    0XXX|XXXX|0011|0011  4TTTT+0 XXXX|XXXX|0011|0011
                             4TTTT+1 OFFSET
                             4TTTT+2 DIRECTION
                             4TTTT+3 MAX ERROR
                             4TTTT+4 AT COMMAND POSITION
                             4TTTT+5 NEAR COMMAND POSITION
                             4TTTT+6 EXTEND LIMIT
                             4TTTT+7 RETRACT LIMIT
```

Axis 2 parameters are set using the same format as above except the index varies from 4 to 7 (0100 to 0111) instead of 0 to 3. Axis 3 uses indeces 8 (1000) through B (1011), and axis 4 uses indeces C (1100) through F (1111).

COMMAND TYPES 4 and 5 - ASCII Commands (0XXX XXXX 010N NNNN).

Format B - 4 Commands.

These commands usually don't require parameters. The 'G' command is an exception. The MMC acknowledges these commands by setting the Active bit in the STATUS until the next time the drop is accessed.

```
                BIT #
        0000|0000|0111|1111
        1234|5678|9012|3456
        -------------------
HEX     SAR1|SAR0|CMND|INDX
VALUE   -------------------
XX40    0XXX|XXXX|0100|0000  - @   Reserved
XX41    0XXX|XXXX|0100|0001  - A   Reserved
XX42    0XXX|XXXX|0100|0010  - B   Reserved
XX43    0XXX|XXXX|0100|0011  - C   Reserved
XX44    0XXX|XXXX|0100|0100  - D   Reserved
XX45    0XXX|XXXX|0100|0101  - E   Reserved
XX46    0XXX|XXXX|0100|0110  - F   Feed Forward Adjust
XX47    0XXX|XXXX|0100|0111  - G   Go using last profile
XX48    0XXX|XXXX|0100|1000  - H   Halt Command
XX49    0XXX|XXXX|0100|1001  - I   Reserved
XX4A    0XXX|XXXX|0100|1010  - J   Reserved
XX4B    0XXX|XXXX|0100|1011  - K   Reserved
XX4C    0XXX|XXXX|0100|1100  - L   Reserved
XX4D    0XXX|XXXX|0100|1101  - M   Reserved
XX4E    0XXX|XXXX|0100|1110  - N   Single Axis Do Nothing
XX4F    0XXX|XXXX|0100|1111  - O   Open Loop

XX50    0XXX|XXXX|0101|0000  - P   Initialize Parameters Command
XX51    0XXX|XXXX|0101|0001  - Q   Reserved
XX52    0XXX|XXXX|0101|0010  - R   Restore null
XX53    0XXX|XXXX|0101|0011  - S   Save null
XX54    0XXX|XXXX|0101|0100  - T   Reserved
XX55    0XXX|XXXX|0101|0101  - U   Reserved
XX56    0XXX|XXXX|0101|0110  - V   Reserved
XX57    0XXX|XXXX|0101|0111  - W   Reserved
XX58    0XXX|XXXX|0101|1000  - X   Reserved
XX59    0XXX|XXXX|0101|1001  - Y   Retrieve Plot Information
XX5A    0XXX|XXXX|0101|1010  - Z   Reserved
XX5B    0XXX|XXXX|0101|1011  - [   Reserved
XX5C    0XXX|XXXX|0101|1100  - /   Reserved
XX5D    0XXX|XXXX|0101|1101  - ]   Reserved
XX5E    0XXX|XXXX|0101|1110  - ^   Reserved
XX5F    0XXX|XXXX|0101|1111  - _   Reserved
```

The 8 output registers are defined as follows:

```
4TTTT+0 Axis 0 Command
4TTTT+1 Axis 0 REQUESTED POSITION (for Go command)
4TTTT+2 Axis 1 Command
4TTTT+3 Axis 1 REQUESTED POSITION (for Go command)
4TTTT+4 Axis 2 Command
4TTTT+5 Axis 2 REQUESTED POSITION (for Go command)
4TTTT+6 Axis 3 Command
4TTTT+7 Axis 3 REQUESTED POSITION (for Go command)
```

'F' - Set Feed Forward (46H) (70)

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 was 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 set-up easier.

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

'G' - go to REQUESTED POSITION (47H) (71)

The programmable controller is responsible for making sure all initialization parameter words are valid when the 'G' or GO command is given. Normally, once the ACCELERATION, DECELERATION and MAXIMUM SPEED are set, only the REQUESTED POSITION needs to be changed. Once set, a 'G' put in the COMMAND word is used to get the axis to move. The MAXIMUM SPEED can be changed while the axis is moving. The motion controller module will ramp to the new speed at the rate specified by the ACCELERATION and DECELERATION parameters.

'H' - Halt (48H) (72)

The HALT command is used for an emergency 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, and the null update will be disabled. Jogging an axis is accomplished by alternating GO and HALT commands.

'N' - Single Axis Do Nothing (4EH) (78)

The Single Axis Do Nothing command is used when giving manual commands to an axis. The N is entered first, the Data for the axis is modified, and then the N is changed to another valid command. This is necessary because it is impossible to modify both the command and data for an axis manually before the first one is sent to the MMC 188/40.

'O' - Open Loop (4FH) (79)

CAUTION: Use this command with care. When operating open loop all safety features on the MMC 188/40 are disabled.

The Open Loop command allows the Programmable Controller to specify values for the analog output directly. The output has a range of -2048 to +2047 where 0 is null and 2047 is full extend drive.

'P' - set Parameters (50H) (80)

This command should be given as the first command after start-up and only when all of the axes are stopped. All initialization parameters are updated when a 'P' command is given. The minimum requirement of this command is to set the EXTEND and RETRACT LIMITS to their proper values (see Module Start-Up). When a 'P' command is given, the MMC 188/40 will wait for a few milliseconds to get the ACTUAL POSITION of the axis. This position is then copied into the TARGET and COMMAND POSITIONS.

If a 'P' command is given when the axis is in motion, only the following parameters will be updated:

ESOP MASK
HALT MASK
PROPORTIONAL STATIC GAIN
PROPORTIONAL EXTEND GAIN
PROPORTIONAL RETRACT GAIN
MAX ERROR
EXTEND FEED FORWARD
RETRACT FEED FORWARD
INTEGRAL GAIN
DIFFERENTIAL GAIN
AT COMMAND POSITION
NEAR COMMAND POSITION

Issuing a 'P' "on the fly" can be useful for systems with large changes in their dynamics during a move.

'R' - Restore Null (52H) (82)

The Restore Null command restores the last saved value of null. This value will be 2048 if no previous Save Null or NEW NULL was done.

'S' - Save Null (53H) (83)

The Save Null command saves the current value of the null so that it may be recalled later by a Restore Null command.

'Y' - Retrieve Graph Data (59H) (89)

The Retrieve Graph Data command moves graphics information stored on the MMC 188/40 to the Programmable Controller.

COMMAND TYPE A - Get Profiles (0XXX XXXX 1010 NNNN).

Format A - Single Command.

These Commands allow the programmer to retrieve profile data from the MMC188/40 that is set up with DCSMON. Notice that the returned command is the corresponding Set Profile command (command type 2). This allows copying the input registers to the output register and making any required changes to the parameters before they are sent back to the MMC 188/40.

```
4TTTT     0XXX|XXXX|1010|NNNN     UPLOAD PROFILE NNNN
     Returns:        3TTTT+0 XXXX|XXXX|0010|NNNN
                     3TTTT+1 MODE
                     3TTTT+2 ACCELERATION
                     3TTTT+3 DECELERATION
                     3TTTT+4 MAX SPEED
                     3TTTT+5 Ignore
                     3TTTT+6 Ignore
                     3TTTT+7 Ignore
```

NNNN represents a binary number from 0 (0000) through 16 (1111) indicating the profile number to be uploaded.

When using this command the SAR0 and SAR1 fields as well as the 7 data registers are ignored by the MMC 188/40.

COMMAND TYPE B - Get Parameters (0XXX XXXX 1011 NNNN).

Format A - Single Command.

These commands allows the programmer to retrieve the current state of the MMC 188/40 initialization parameters. Notice that the returned command is the corresponding Download Parameters command (command type 3). This allows copying the input registers to the output register and making any required changes to the parameters before they are sent back to the MMC 188/40. Reserved registers should not be modified.

```
               BIT #
            |    | 111|1111
        1234|5678|9012|3456
HEX     -------------------
VALUE   SAR1|SAR0|CMND|INDX

XXB0    0XXX|XXXX|1011|0000      Upload Parameters AXIS 1  GROUP 1
                     Returns 3TTTT+0 XXXX|XXXX|0011|0000
                             3TTTT+1 Ignore
                             3TTTT+2 Ignore
                             3TTTT+3 Ignore
                             3TTTT+4 Ignore
                             3TTTT+5 NEW NULL
                             3TTTT+6 EMERGENCY STOP MASK
                             3TTTT+7 HALT MASK

XXB1    0XXX|XXXX|1011|0001      Upload Parameters AXIS 1  GROUP 2
                     Returns 3TTTT+0 XXXX|XXXX|0011|0001
                             3TTTT+1 Ignore
                             3TTTT+2 FEED FORWARD ADVANCE
                             3TTTT+3 NULL UPDATE
                             3TTTT+4 Ignore
                             3TTTT+5 DITHER
                             3TTTT+6 HYSTERESIS
                             3TTTT+7 PRO STATIC GAIN

XXB2    0XXX|XXXX|1011|0010      Upload Parameters AXIS 1  GROUP 3
                     Returns 3TTTT+0 XXXX|XXXX|0011|0010
                             3TTTT+1 PRO EXTEND GAIN
                             3TTTT+2 PRO RETRACT GAIN
                             3TTTT+3 INTEGRAL GAIN
                             3TTTT+4 DIFFERENTIAL GAIN
                             3TTTT+5 EXTEND FEED FORWARD
                             3TTTT+6 RETRACT FEED FORWARD
                             3TTTT+7 SCALE
```

```
                BIT #
            |    | 111|1111
        1234|5678|9012|3456
HEX     -------------------
VALUE   SAR1|SAR0|CMND|INDX

XXB3    0XXX|XXXX|1011|0011      Upload Parameters AXIS 1  GROUP 4
                     Returns 3TTTT+0 XXXX|XXXX|0011|0011
                             3TTTT+1 OFFSET
                             3TTTT+2 DIRECTION
                             3TTTT+3 MAX ERROR
                             3TTTT+4 AT COMMAND POSITION
                             3TTTT+5 NEAR COMMAND POSITION
                             3TTTT+6 EXTEND LIMIT
                             3TTTT+7 RETRACT LIMIT
```

Axis 2 parameters are transfered using the same format as above except the index varies from 4 to 7 (0100 to 0111) instead of 0 to 3. Axis 3 uses indeces 8 (1000) through B (1011), and axis 4 uses indeces C (1100) through F (1111).

When using this command the SAR0 and SAR1 fields as well as the 7 data registers are ignored by the MMC 188/40.

COMMAND TYPE F - Diagnostics (0XXX XXXX 1111 XXXX)

Format A - Single Command.

The "Diagnostic" commands are for testing the MMC188/40 under safe conditions. The command echoes the output registers in the input registers. This tests the 984 to MMC188/40 communications.

```
4TTTT+0 = XXXX|XXXX|1111|XXXX  (XXFXH)  3TTTT+0 = Echo 4TTTT+0
4TTTT+1 =    don't care                 3TTTT+1 = Echo 4TTTT+1
4TTTT+2 =      "    "                   3TTTT+2 = Echo 4TTTT+2
4TTTT+3 =      "    "                   3TTTT+3 = Echo 4TTTT+3
4TTTT+4 =      "    "                   3TTTT+4 = Echo 4TTTT+4
4TTTT+5 =      "    "                   3TTTT+5 = Echo 4TTTT+5
4TTTT+6 =      "    "                   3TTTT+6 = Echo 4TTTT+6
4TTTT+7 =      "    "                   3TTTT+7 = Echo 4TTTT+7
```

When using this command the SAR0 and SAR1 fields are ignored by the MMC 188/40.

Input Registers

Status Information can be selectively returned to the 984 by using the Status Area Request (SAR) fields of the command words. There are two formats for the returned register fields. The first format uses each of the 4 command registers to request 2 status area registers for each axis as follows:

```
 OFFSET  AXIS          REGISTER DESCRIPTION
3TTTT+0    1      specified by COMMAND 1, SAR0
3TTTT+1    1      specified by COMMAND 1, SAR1
3TTTT+2    2      specified by COMMAND 2, SAR0
3TTTT+3    2      specified by COMMAND 2, SAR1
3TTTT+4    3      specified by COMMAND 3, SAR0
3TTTT+5    3      specified by COMMAND 3, SAR1
3TTTT+6    4      specified by COMMAND 4, SAR0
3TTTT+7    4      specified by COMMAND 4, SAR1
```

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	TARGET Speed
7	0111	DRIVE NULL

To get the STATUS and ACTUAL POSITION one would set SAR0 to 4 (0100) and SAR1 to 2 (0010). In the example below, SAR0 of each axis is requesting the status register, and SAR1 is requesting a different register from the status area.

```
        SAR1 SAR0 CMND INDX  (HEX)
4TTTT+0 0010|0100|0XXX|XXXX  (24XX)  3TTTT+0 AXIS 1 STATUS
                                     3TTTT+1 AXIS 1 ACTUAL POSITION
4TTTT+2 0011|0100|0XXX|XXXX  (34XX)  3TTTT+2 AXIS 2 STATUS
                                     3TTTT+3 AXIS 2 COUNTS
4TTTT+4 0000|0100|0XXX|XXXX  (04XX)  3TTTT+4 AXIS 3 STATUS
                                     3TTTT+5 AXIS 3 COMMAND POSITION
4TTTT+6 0111|0100|0XXX|XXXX  (74XX)  3TTTT+6 AXIS 4 STATUS
                                     3TTTT+7 AXIS 4 DRIVE NULL
```

The second format is specified by making Status Area Request 0 (SAR0) and Status Area Request 1 (SAR1) the same in the first COMMAND register (4TTTT+0). When SAR0 and SAR1 in the first COMMAND register are the same, the readback byte in the other registers is ignored.

```
HEX     SAR1|SAR0|CMND|INDX
00XX    0000|0000|0XXX|XXXX    RETURNS AXIS 1 STATUS AREA
11XX    0001|0001|0XXX|XXXX    RETURNS AXIS 2 STATUS AREA
22XX    0010|0010|0XXX|XXXX    RETURNS AXIS 3 STATUS AREA
33XX    0011|0011|0XXX|XXXX    RETURNS AXIS 4 STATUS AREA
```

```
 OFFSET  REGISTER DESCRIPTION
3TTTT+0  COMMAND POSITION
3TTTT+1  TARGET POSITION
3TTTT+2  ACTUAL POSITION
3TTTT+3  TRANSDUCER COUNTS
3TTTT+4  STATUS
3TTTT+5  DRIVE OUTPUT
3TTTT+6  TARGET SPEED
3TTTT+7  DRIVE NULL
```

Format A commands having only one command register (and 7 data registers) can retrieve all information for one axis or the same two words of information for 4 axes. To obtain all the information for one axis make SAR0 the same as SAR1 as described above. To obtain the same two words from all four axes enter the two SAR values desired in the first command register.

FN10 CUSTOM LOADABLE

The FN10 custom loadable is used to communicate the MMC 188/40. There are two files that must be installed on the Modsoft PC: FN10.DAT and DXFDT.SYS. The FN10.DAT file can be put into the \modsoft\programs directory, and the DXFDT.SYS file must be put in the \modsoft\runtime directory. The FN10.DAT is the actual custom loadable and the DXFDT.SYS is the DX zoom text.

To run FN10 in a Programmable Controller the loadable must be loaded into the Modsoft configuration. See the Modicon Modsoft Programmer User Manual for loading FN10; FN10 must be assigned an opcode which will not conflict with other loadables. Opcode 5F is usually free.

This chapter describes the sample program provided with the MMC 188/40 module and the subfunctions in the FN10 custom loadable. Each subfunction has its own DX zoom screen which is a window into the parameters and registers associated with each subfunction. Subfunction 1 gives a summary of all the other subfunctions available.

Input and output drops used by the FN10 custom loadable must reside in the same segment. This will eliminate a possible problem where status bits are set and cleared in the same scan by different segments. For example, the Active bit in the status word is set by the MMC 188/40 when a command has completed. The Active bit is cleared by the Programmable Controller on the next segment I/O input. If the input and output drops are not in the same segment as the Init Subfunction 32, the subfunction would never complete. Segment drops are assigned by the Modsoft segment scheduler. See the Modicon 984 Programmable Controller Systems Manual.

The FN10.DAT file on the DCSMON distribution disk contains all the subfunctions. The FN10A.DAT contains all subfunctions except 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, and 31. These are the indirect math and Move Block From Table subfunctions. If FN10A.DAT will be used, it must be renamed FN10.DAT. See FILELIST.TXT on the distribution disk for more information.

NOTE: To use a 984 E-series PLC with expanded 6 digit register addressing, you must use the FN10.EXE user loadable. You cannot use an FN10.DAT or FN10A.DAT with any PLC using 6 digit register addressing.

Functionally the FN10.EXE user loadable is identical to the FN10.DAT custom loadable. Use the CLSSDOC.TXT for user-loadable documentation, but note the following exceptions. The old custom loadable specified 4x and 3x register parameters using the whole number. The new user loadable specifies 4x and 3x registers using just the least significant digits. In other words the 4x and 3x most significant digit is understood. For example:

Old Specifier	New Specifier
30001	1
40180	180

So old PLC programs using the FN10.DAT custom loadable must convert the 4x and 3x parameters. See the new demo called MSDEMOE. Note - the indirect math subfunctions now uses only 4x registers as indirect pointers.

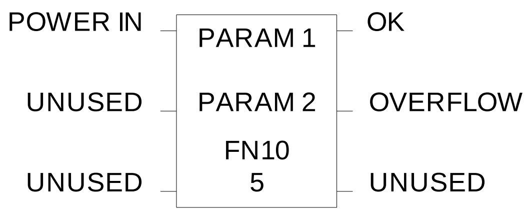
FN10.EXE also requires a Modicon support loadable called NSUP.EXE. NSUP.EXE must be configured as the first user loadable followed the FN10.EXE user loadable.

SUBFUNCTIONS

Subfunction Number			Description	Memory Usage PARAM 1	PARAM 2
Subfunction	5	DTOW:	Double To 16 Bit Word	3	1
Subfunction	6	WTOD:	16 Bit Word To Double	3	1
Subfunction	7	ADDI:	16 Bit Add Immediate	2	1
Subfunction	8	ADDX:	16 Bit Add Indirect	2	1
Subfunction	9	SADI:	Signed 16 Bit Add Immediate	2	1
Subfunction	10	SADX:	Signed 16 Bit Add Indirect	2	1
Subfunction	11	SUBI:	16 Bit Subtract Immediate	2	1
Subfunction	12	SUBX:	16 Bit Subtract Indirect	2	1
Subfunction	13	SSBI:	Signed 16 Bit Subtract Immediate	2	1
Subfunction	14	SSBX:	Signed 16 Bit Subtract Indirect	2	1
Subfunction	15	CMPI:	16 Bit Compare Immediate	2	1
Subfunction	16	CMPX:	16 Bit Compare Indirect	2	1
Subfunction	17	SCPI:	Signed 16 Bit Compare Immediate	2	1
Subfunction	18	SCPX:	Signed 16 Bit Compare Indirect	2	1
Subfunction	19	MULI:	Multiply Immediate	2	2
Subfunction	20	MULX:	Multiply Indirect	2	2
Subfunction	21	SMLI:	Signed Multiply Immediate	2	2
Subfunction	22	SMLX:	Signed Multiply Indirect	2	2
Subfunction	23	DIVI:	Divide Immediate	2	2
Subfunction	24	DIVX:	Divide Indirect	2	2
Subfunction	25	SDVI:	Signed Divide Immediate	2	2
Subfunction	26	SDVX:	Signed Divide Indirect	2	2
Subfunction	27	GPAR:	Get Parameters From MMC 188/40	8	multiple of 24 (96 max)
Subfunction	28	SPAR:	Set Parameters In MMC 188/40	15	multiple of 24 (96 max)
Subfunction	29	GPRO	Get Profile(S) From MMC 188/40	8	multiple of 4 (64 max)
Subfunction	30	SPRO:	Set Profile(S) In MMC 188/40	15	multiple of 4 (64 max)
Subfunction	31	MBFT:	Move Block From Table	7	Table size
Subfunction	32	INIT:	Initialize 4X Output Registers	4	8 (output registers)
Subfunction	33	MSTT:	Move Status To 0X Table	9	8

MATH SUBFUNCTIONS

SUBFUNCTION 5 - DOUBLE TO 16 BIT WORD



PARAM 1 comprises 3 consecutive 4X registers:

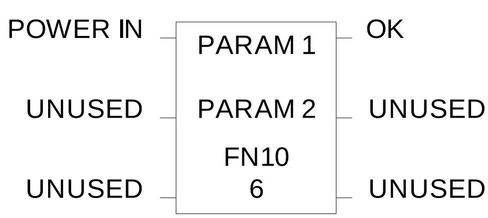
4XXXX	=	5	(subfunction number)
4XXXX + 1	=	Double Precision HIGH	
4XXXX + 2	=	Double Precision LOW	

PARAM 2 comprises 1 4X register:

4XXXX	=	16 bit word result

OK	=	operation performed successfully
OVERFLOW	=	double precision integer is greater than 65,535

SUBFUNCTION 6 - 16 BIT WORD TO DOUBLE



PARAM 1 comprises 3 consecutive 4X registers:

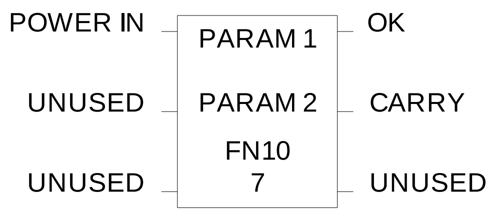
4XXXX	=	6	(subfunction number)
4XXXX + 1	=	Double Precision HIGH result	
4XXXX + 2	=	Double Precision LOW result	

PARAM 2 comprises 1 4X register:

4XXXX	=	16 bit word

OK	=	operation performed successfully

SUBFUNCTION 7 - 16 BIT ADD IMMEDIATE



PARAM 1 comprises 2 consecutive 4X registers:

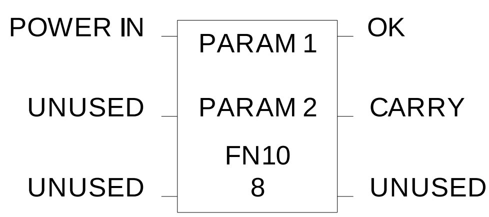
4XXXX	=	7	(subfunction number)
4XXXX + 1	=	16 bit word parameter	

PARAM 2 comprises 1 4X register:

4XXXX	=	16 bit word accumulator

OK	=	operation performed successfully (accumulator = accumulator + parameter)
CARRY	=	on when accumulator result would be greater than 65,535

SUBFUNCTION 8 - 16 BIT ADD INDIRECT



PARAM 1 comprises 2 consecutive 4X registers:

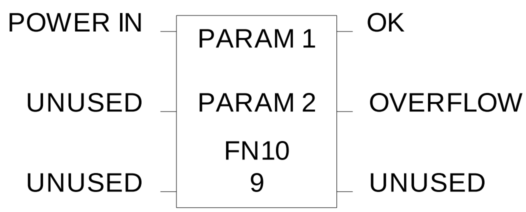
4XXXX	=	8	(subfunction number)
4XXXX + 1	=	3X or 4X address of 16 bit word parameter	

PARAM 2 comprises 1 4X register:

4XXXX	=	16 bit word accumulator

OK	=	operation performed successfully (accumulator = accumulator + parameter)
CARRY	=	on when accumulator result would be greater than 65,535

SUBFUNCTION 9 - SIGNED 16 BIT ADD IMMEDIATE



PARAM 1 comprises 2 consecutive 4X registers:

4XXXX = 9 (subfunction number)

4XXXX + 1 = signed 16 bit word parameter

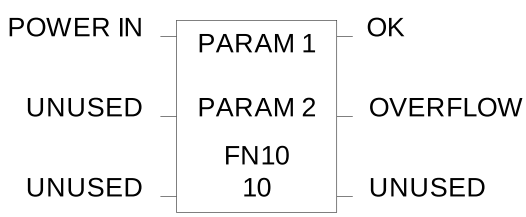
PARAM 2 comprises 1 4X register:

4XXXX = signed 16 bit word accumulator

OK = operation performed successfully (accumulator = accumulator + parameter)

CARRY = on when accumulator result would be greater than 32,767 or less than -32,768

SUBFUNCTION 10 - SIGNED 16 BIT ADD INDIRECT



PARAM 1 comprises 2 consecutive 4X registers:

4XXXX = 10 (subfunction number)

4XXXX + 1 = 3X or 4X address of 16 bit word parameter

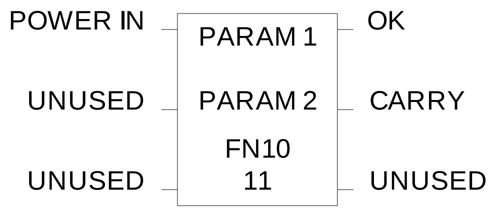
PARAM 2 comprises 1 4X register:

4XXXX = 16 bit word accumulator

OK = operation performed successfully (accumulator = accumulator + parameter)

CARRY = on when accumulator result would be greater than 32,767 or less than 32,768

SUBFUNCTION 11 - 16 BIT SUBTRACT IMMEDIATE



PARAM 1 comprises 2 consecutive 4X registers:

4XXXX = 11 (subfunction number)

4XXXX + 1 = 16 bit word parameter

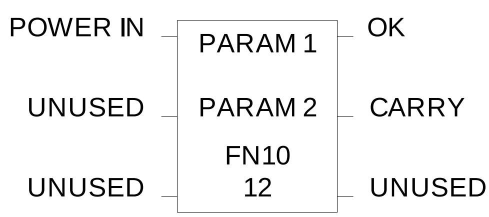
PARAM 2 comprises 1 4X register:

4XXXX = 16 bit word accumulator

OK = operation performed successfully (accumulator = accumulator - parameter)

CARRY = on when accumulator result would be less than 0

SUBFUNCTION 12 - 16 BIT SUBTRACT INDIRECT



PARAM 1 comprises 2 consecutive 4X registers:

4XXXX = 12 (subfunction number)

4XXXX + 1 = 3X or 4X address of 16 bit word parameter

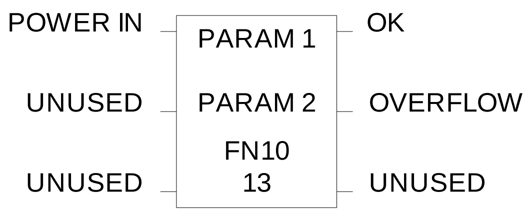
PARAM 2 comprises 1 4X register:

4XXXX = 16 bit word accumulator

OK = operation performed successfully (accumulator = accumulator - parameter)

CARRY = on when accumulator result would be less than 0

SUBFUNCTION 13 - SIGNED 16 BIT SUBTRACT IMMEDIATE



PARAM 1 comprises 2 consecutive 4X registers:

4XXXX = 13 (subfunction number)

4XXXX + 1 = signed 16 bit word parameter

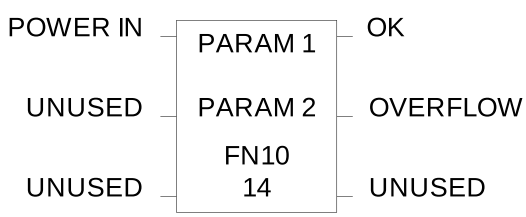
PARAM 2 comprises 1 4X register:

4XXXX = signed 16 bit word accumulator

OK = operation performed successfully (accumulator = accumulator - parameter)

CARRY = on when accumulator result would be greater than 32,767 or less than -32,768

SUBFUNCTION 14 - SIGNED 16 BIT SUBTRACT INDIRECT



PARAM 1 comprises 2 consecutive 4X registers:

4XXXX = 14 (subfunction number)

4XXXX + 1 = 3X or 4X address of 16 bit word parameter

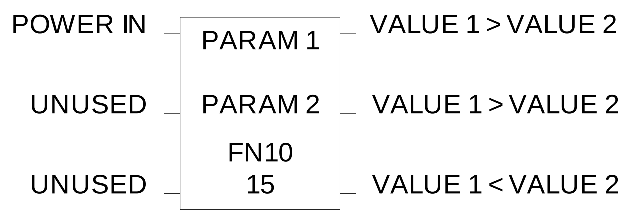
PARAM 2 comprises 1 4X register:

4XXXX = 16 bit word accumulator

OK = operation performed successfully (accumulator = accumulator - parameter)

CARRY = on when accumulator result would be greater than 32,767 or less than 32,768

SUBFUNCTION 15- 16 BIT COMPARE IMMEDIATE



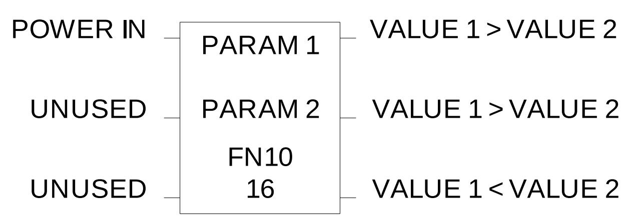
PARAM 1 comprises 2 consecutive 4X registers:

4XXXX	=	15	(subfunction number)
4XXXX + 1	=	16 bit word Value 1	

PARAM 2 comprises 1 4X register:

4XXXX	=	16 bit word Value 2

SUBFUNCTION 16 - 16 BIT COMPARE INDIRECT



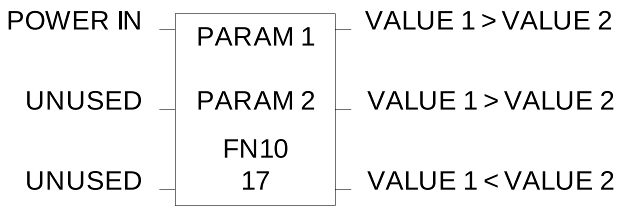
PARAM 1 comprises 2 consecutive 4X registers:

4XXXX	=	16	(subfunction number)
4XXXX + 1	=	3X or 4X address of 16 bit word Value 1	

PARAM 2 comprises 1 4X register:

4XXXX	=	16 bit word Value 2

SUBFUNCTION 17 - SIGNED 16 BIT COMPARE IMMEDIATE



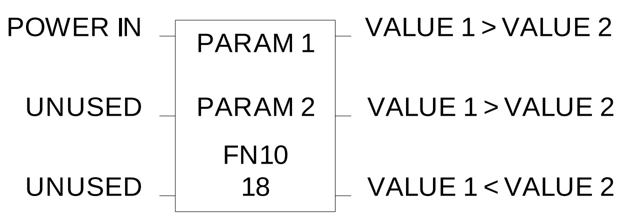
PARAM 1 comprises 2 consecutive 4X registers:

4XXXX	=	17	(subfunction number)
4XXXX + 1	=	signed 16 bit word Value 1	

PARAM 2 comprises 1 4X register:

4XXXX	=	signed 16 bit word Value 2

SUBFUNCTION 18 - SIGNED 16 BIT COMPARE INDIRECT



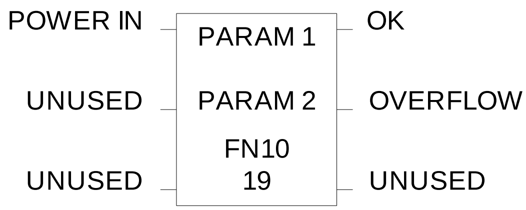
PARAM 1 comprises 2 consecutive 4X registers:

4XXXX	=	18	(subfunction number)
4XXXX + 1	=	3X or 4X address of 16 bit word Value 1	

PARAM 2 comprises 1 4X register:

4XXXX	=	16 bit word Value 2

SUBFUNCTION 19 - MULTIPLY IMMEDIATE



PARAM 1 comprises 2 consecutive 4X registers:

- 4XXXX = 19 (subfunction number)
- 4XXXX + 1 = 16 bit word parameter

PARAM 2 comprises 2 consecutive 4X register:

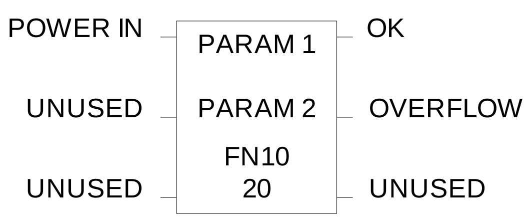
- 4XXXX = Accumulator LOW
- 4XXXX + 1 = Accumulator HIGH

OK = operation performed successfully

(accumulator HIGH : accumulator LOW = accumulator LOW x parameter)

OVERFLOW = on when accumulator result exceeds 16 bits

SUBFUNCTION 20 - MULTIPLY INDIRECT



PARAM 1 comprises 2 consecutive 4X registers:

- 4XXXX = 20 (subfunction number)
- 4XXXX + 1 = 3X or 4X address of 16 bit word parameter

PARAM 2 comprises 2 consecutive 4X register:

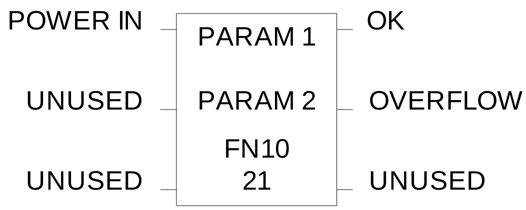
- 4XXXX = Accumulator LOW
- 4XXXX + 1 = Accumulator HIGH

OK = operation performed successfully

(accumulator HIGH : accumulator LOW = accumulator LOW x parameter)

OVERFLOW = on when accumulator result exceeds 16 bits

SUBFUNCTION 21 - SIGNED MULTIPLY IMMEDIATE



PARAM 1 comprises 2 consecutive 4X registers:

4XXXX = 21 (subfunction number)

4XXXX + 1 = 16 bit word parameter

PARAM 2 comprises 2 consecutive 4X register:

4XXXX = Accumulator LOW

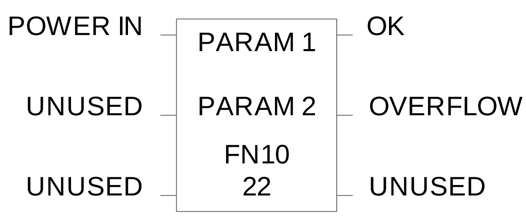
4XXXX + 1 = Accumulator HIGH

OK = operation performed successfully

(accumulator HIGH : accumulator LOW = accumulator LOW x parameter)

OVERFLOW = on when accumulator result exceeds 16 bits

SUBFUNCTION 22 - SIGNED MULTIPLY INDIRECT



PARAM 1 comprises 2 consecutive 4X registers:

4XXXX = 22 (subfunction number)

4XXXX + 1 = 3X or 4X address of 16 bit word parameter

PARAM 2 comprises 2 consecutive 4X register:

4XXXX = Accumulator LOW

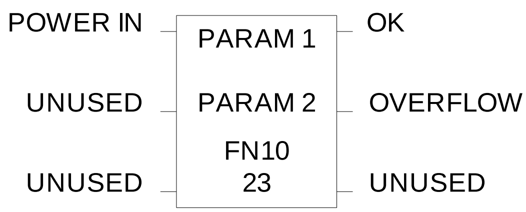
4XXXX + 1 = Accumulator HIGH

OK = operation performed successfully

(accumulator HIGH : accumulator LOW = accumulator LOW x parameter)

OVERFLOW = on when accumulator result exceeds 16 bits

SUBFUNCTION 23 - DIVIDE IMMEDIATE



PARAM 1 comprises 2 consecutive 4X registers:

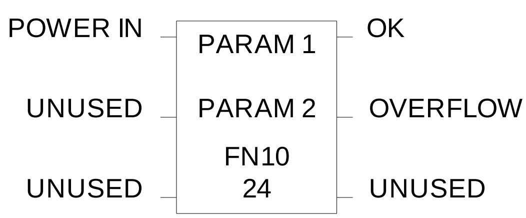
4XXXX	=	23	(subfunction number)
4XXXX + 1	=	16 bit word parameter	

PARAM 2 comprises 2 consecutive 4X register:

4XXXX	=	Accumulator LOW (Quotient)
4XXXX + 1	=	Accumulator HIGH (Remainder)

OK	=	operation performed successfully
		(accumulator = accumulator HIGH : accumulator LOW / parameter)
OVERFLOW	=	on when quotient exceeds 16 bits or divide by zero

SUBFUNCTION 24 - DIVIDE INDIRECT



PARAM 1 comprises 2 consecutive 4X registers:

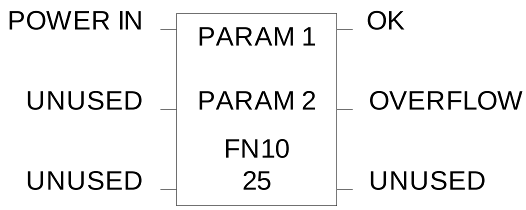
4XXXX	=	24	(subfunction number)
4XXXX + 1	=	3X or 4X address of 16 bit word parameter	

PARAM 2 comprises 2 consecutive 4X register:

4XXXX	=	Accumulator LOW (Quotient)
4XXXX + 1	=	Accumulator HIGH (Remainder)

OK	=	operation performed successfully
		(accumulator = accumulator HIGH : accumulator LOW / parameter)
OVERFLOW	=	on when accumulator result exceeds 16 bits or divide by zero

SUBFUNCTION 25 - SIGNED DIVIDE IMMEDIATE



PARAM 1 comprises 2 consecutive 4X registers:

4XXXX = 25 (subfunction number)

4XXXX + 1 = 16 bit word parameter

PARAM 2 comprises 2 consecutive 4X register:

4XXXX = Accumulator LOW (Quotient)

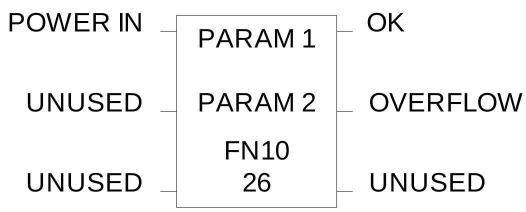
4XXXX + 1 = Accumulator HIGH (Remainder)

OK = operation performed successfully

(accumulator = accumulator HIGH : accumulator LOW / parameter)

OVERFLOW = on when accumulator result exceeds 16 bits or divide by zero

SUBFUNCTION 26 - SIGNED DIVIDE INDIRECT



PARAM 1 comprises 2 consecutive 4X registers:

4XXXX = 26 (subfunction number)

4XXXX + 1 = 3X or 4X address of 16 bit word parameter

PARAM 2 comprises 2 consecutive 4X register:

4XXXX = Accumulator LOW (Quotient)

4XXXX + 1 = Accumulator HIGH (Remainder)

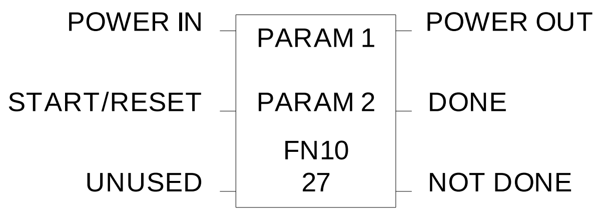
OK = operation performed successfully

(accumulator = accumulator HIGH : accumulator LOW / parameter)

OVERFLOW = on when accumulator result exceeds 16 bits or when divide by zero

MMC 188/40 SUBFUNCTIONS

SUBFUNCTION 27 - GET PARAMETERS FROM MMC 188/40



PARAM 1 comprises 8 consecutive 4X registers:

4XXXX	=	27	(subfunction number)
4XXXX + 1	=	Begin index (0 to 15)	
4XXXX + 2	=	End index (0 to 15)	
4XXXX + 3	=	First input register address (3XXXX)	
4XXXX + 4	=	First Output register address (4XXXX)	

4XXXX + 5 to 4XXXX + 7 are reserved registers and must not be used

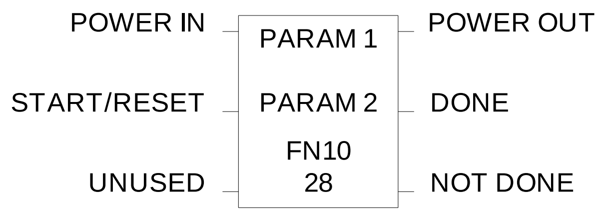
PARAM 2 is the list of initialization parameters for the MMC 188/40. There are 24 per axis.

POWER OUT	=	Power is passed through the subfunction block when all parameters are correct
DONE	=	Transfer complete
NOT DONE	=	The subfunction is in the process of transferring initialization parameters from the MMC 188/40 to memory.

This subfunction will read 1 to 16 parameter groups from the MMC 188/40. All groups can be transferred or a subset by specifying beginning group index and end group index. Indices must be specified with 0 to 15 notation. There are four groups per axis. For example, 0 and 3 in beginning and end index respectively will transfer the first four groups for the first axis.

Each group transferred takes one Programmable Controller scan. So it will take 16 scans to transfer all the parameters for all axes. The second input must be high for the whole transfer, once it reaches the last group to transfer, the transfer will stop. To restart the transfer set the second input low for at least one scan then high. When starting the Programmable Controller program the second input should be low for at least one scan, this will set internal counters to a known state.

SUBFUNCTION 28 - SET PARAMETERS IN MMC 188/40



PARAM 1 comprises 8 consecutive 4X registers:

4XXXX	=	28 (subfunction number)
4XXXX + 1	=	Begin index (0 to 15)
4XXXX + 2	=	End index (0 to 15)
4XXXX + 3	=	First input register address (3XXXX)
4XXXX + 4	=	First Output register address (4XXXX)

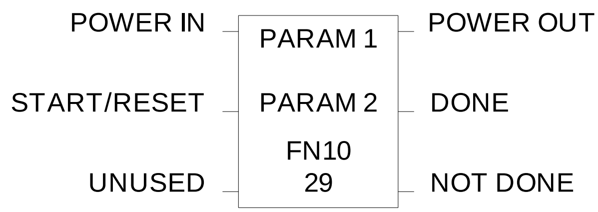
4XXXX + 5 to 4XXXX + 14 are reserved registers and must not be used

PARAM 2 is the list of initialization parameters for the MMC 188/40. There are 24 per axis.

POWER OUT	=	Power is passed through the subfunction block when all parameters are correct
DONE	=	Transfer complete
NOT DONE	=	The subfunction is in the process of transferring initialization parameters from memory to the MMC 188/40.

This subfunction will write 1 to 16 parameter groups to the MMC 188/40. See subfunction 27 for additional details.

SUBFUNCTION 29 - GET PROFILES FROM MMC 188/40



PARAM 1 comprises 8 consecutive 4X registers:

4XXXX	=	29 (subfunction number)
4XXXX + 1	=	Begin index (0 to 15)
4XXXX + 2	=	End index (0 to 15)
4XXXX + 3	=	First input register address (3XXXX)
4XXXX + 4	=	First Output register address (4XXXX)

4XXXX + 5 to 4XXXX + 7 are reserved registers and must not be used

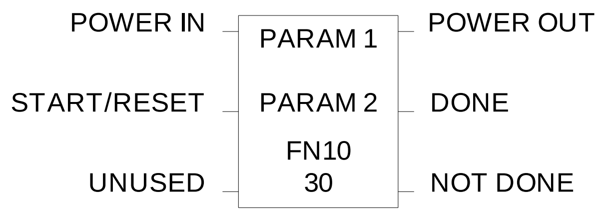
PARAM 2 is the list of profile commands for the MMC 188/40. There are a maximum of 16 trapezoidal profiles with 4 registers each.

POWER OUT	=	Power is passed through the subfunction block when all parameters are correct
DONE	=	Transfer complete
NOT DONE	=	The subfunction is in the process of transferring trapezoidal profiles from the MMC 188/40 to memory.

This subfunction will read 1 to 16 profiles from the MMC 188/40. All profiles can be transferred or a subset by specifying beginning profile index and end profile index. Indices must be specified with 0 to 15 notation. For example, 0 and 3 in beginning and end index respectively will transfer the first four profiles.

Each profile transferred takes one Programmable Controller scan. So it will take 16 scans to transfer all the profiles. The second input must be high for the whole transfer, once it reaches the last profile to transfer, the transfer will stop. To restart the transfer set the second input low for at least one scan then high. When starting the Programmable Controller program the second input should be low for at least one scan, this will set internal counters to a known state.

SUBFUNCTION 30 - SET PROFILES IN MMC 188/40



PARAM 1 comprises 8 consecutive 4X registers:

4XXXX	=	30	(subfunction number)
4XXXX + 1	=	Begin index (0 to 15)	
4XXXX + 2	=	End index (0 to 15)	
4XXXX + 3	=	First input register address (3XXXX)	
4XXXX + 4	=	First Output register address (4XXXX)	

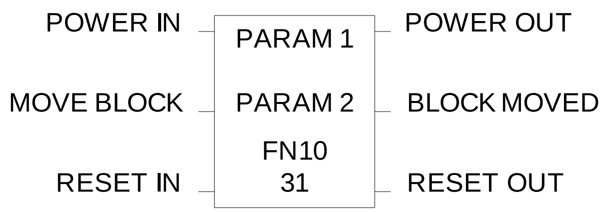
4XXXX + 5 to 4XXXX + 14 are reserved registers and must not be used

PARAM 2 is the list of trapezoidal profiles for the MMC 188/40. There are a maximum of 16 trapezoidal profiles with 4 registers each.

POWER OUT	=	Power is passed through the subfunction block when all parameters are correct
DONE	=	Transfer complete
NOT DONE	=	The subfunction is in the process of transferring trapezoidal profiles from memory to the MMC 188/40.

This subfunction will write 1 to 16 profiles to the MMC 188/40. See subfunction 29 for additional details.

SUBFUNCTION 31 - MOVE BLOCK FROM TABLE



PARAM 1 comprises 7 consecutive 4X registers:

4XXXX	=	31 (subfunction number)
4XXXX + 1	=	Block begin index (0 to N-1)
4XXXX + 2	=	Block end index (0 to N-1)
4XXXX + 3	=	Block size
4XXXX + 4	=	First Output register address (4XXXX)

4XXXX + 5 and 4XXXX + 6 are reserved registers and must not be used

PARAM 2 is the table of commands and requested positions to be transferred to the MMC 188/40.

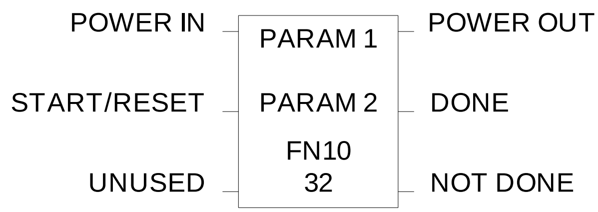
POWER OUT	=	Power is passed through the subfunction block when all parameters are correct
BLOCK MOVED	=	Transfer complete.
RESET OUT	=	Used for chaining subfunction 31 blocks together.

This subfunction will write 1 to N blocks of a table to the specified output register location. All blocks can be transferred by specifying beginning block index, end block index and block size in words. Indices must be specified with 0 to N-1 notation. For example, 0, 3 and 2 in beginning index, end index and block size respectfully will transfer the first four blocks of the table to the specified output register location. This transfer will take four scans, one for each high condition of the second input. All four blocks will be transferred to the same two locations of the specified output register.

Each block transferred takes one Programmable Controller scan. The second input must be high for each transfer, once it reaches the last block to transfer, the transfer will restart at the beginning block. When starting the Programmable Controller program the third input should be low for at least on scan, this will set internal counters to a known state.

This subfunction is useful for writing command/position words, from a table, to one axis or multiple axes on the MMC 188/40 board.

SUBFUNCTION 32 - INITIALIZE 4X OUTPUT REGISTERS



PARAM 1 comprises 4 consecutive 4X registers:

4XXXX	=	32 (subfunction number)
4XXXX + 1	=	First input register address (3XXXX)

4XXXX + 2 and 4XXXX + 3 are reserved registers and must not be used

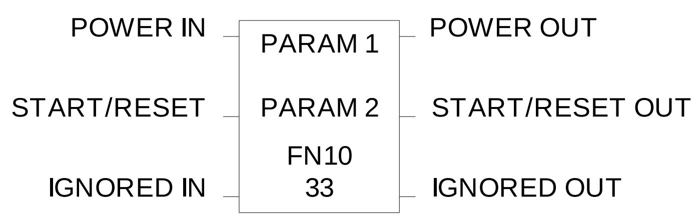
PARAM 2 comprises the 8 output registers (4XXXX).

POWER OUT	=	Power is passed through the subfunction block when all parameters are correct
DONE	=	Output registers initialized
NOT DONE	=	Initializing output registers

This subfunction will initialize the specified output registers to the current position of all four axes. For each axis it will issue a Halt command (2448 hex) followed by a Do Nothing command (244e hex), this will return the current status and position to the specified input registers. The subfunction will then copy the actual position for each axis to the specified output registers.

The second input must be high for the whole transfer. To restart the transfer set the second input low for at least one scan then high. When starting the Programmable Controller program the second input should be low for at least one scan, this sill set internal counters to a known state.

SUBFUNCTION 33 - MOVE STATUS TO 0X TABLE



PARAM 1 comprises 4 consecutive 8X registers:

4XXXX = 33 (subfunction number)

4XXXX + 1 = First input register address (3XXXX)

4XXXX + 2 = First output point address (0XXXX)

4XXXX + 3 and 4XXXX + 7 are reserved registers and must not be used

PARAM 2 comprises the 8 output registers (4XXXX).

POWER OUT = Power is passed through the subfunction block when all parameters are correct

DONE = Output registers initialized

NOT DONE = Initializing output registers

This subfunction will copy any MMC 188/40 status words from the input registers into the 0X discrete outputs for all four axes using 64 bits, coils or contacts. This transfer will take place every scan only for valid SR0/SR1 status requests. New SR0/SR1 status requests are activated when the Active bit is high in the status word of the input registers. SR0/SR1 status requests are made with MMC 188/40 commands, see mmc18840.txt for details. This subfunction can take the place of multiple Modsoft BLKM instructions and lots of "glue" logic.

The first output will be high whenever the Active bit is high for any axis. This lets the 984 know that the MMC 188/40 got the last command and that a new command can be given.

The second input must be high in order to copy status words into discrete outputs. Setting the second input low will clear all current SR0/SR1 status requests. When starting the Programmable Controller the second input should be low for at least on scan, this will set the subfunction to a known state.

The second output is the second input passed through. this allows the function blocks to be connected in a compact form with out a lot of "glue" logic.

The third input is ignored, but power is passed the third output.

DEMO PROGRAM

Included on the DCSMON diskette are 3 demo programs for the MMC 188/40:

MSDMO runs under Modsoft,

GSDEMO runs under Graysoft, and

TSDEMO runs under Taylor Software.

The demos consist of 3 networks. The first network initializes the module and moves axis 1 back and forth. The second network automatically adjusts the feed forward parameter on any axis when the lag or lead error becomes excessive. The third network moves axes 2,3 and 4 in a coordinated manner.

NETWORK 1 - Initializes the MMC 188/40 and moves axis 1.

NOTE: The 4 blocks in the upper left hand corner of the network need to be repeated for each module in the system.

Contact 01024 is open during the first Programmable Controller scan. It is closed at the end of the scan and remains closed until the program is restarted. This contact provides the low-to-high transition needed to start the module initialization process.

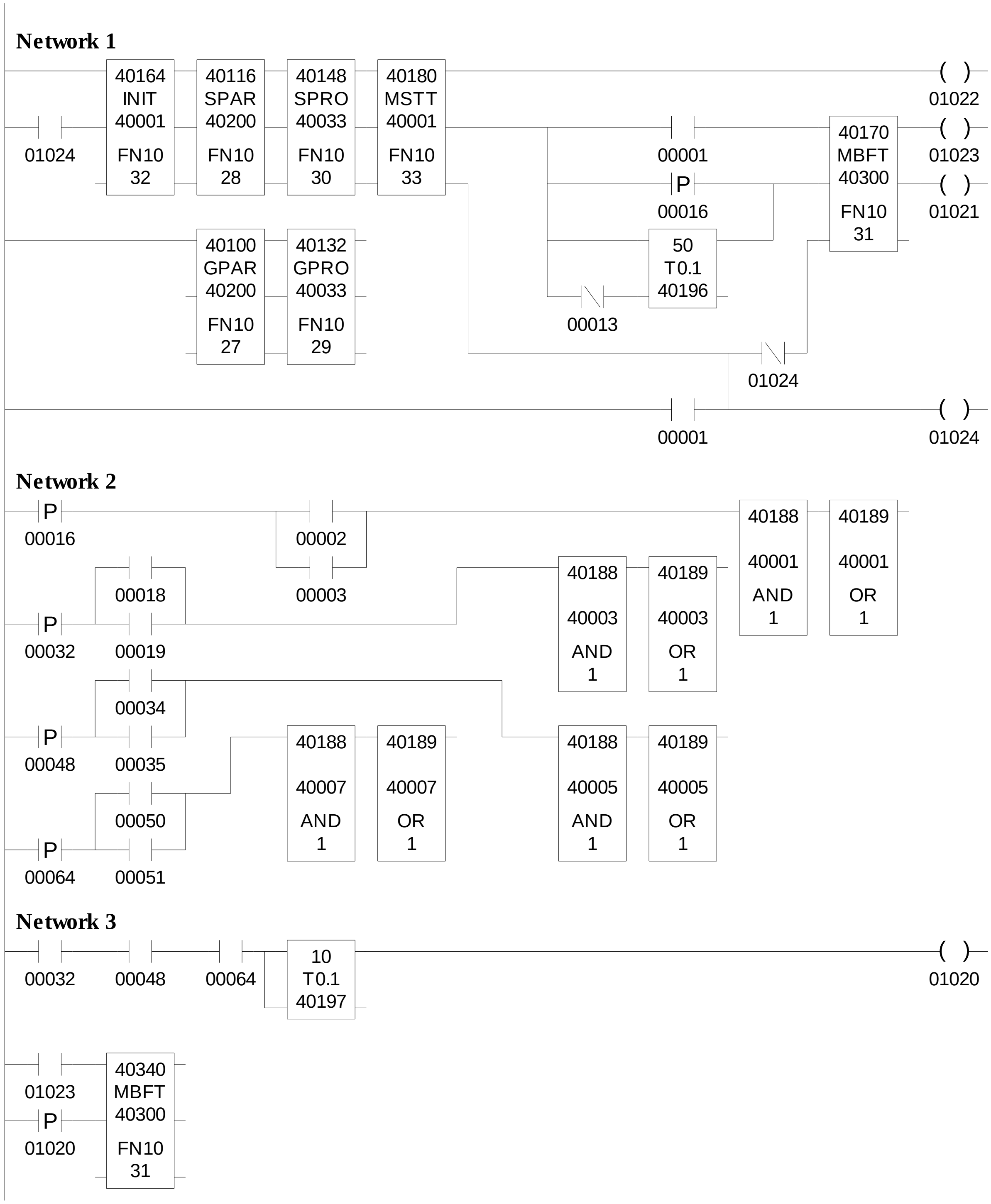
FN10 #32 (Initialize 4X Output Registers) is activated on the transition of its second input, and it initializes the MMC 188/40 to a known state so the next blocks can communicate properly.

FN10 #28 (Set Parameters in MMC 188/40) is activated by the output from FN10 #32. FN10 #28 transfers the initialization parameters from the programmable controller memory to the MMC 188/40 and then issues a "P" command to all axes. At this point the module is ready to accept motion commands.

FN10 #30 (Set Profiles in MMC 188/40) is activated by the middle node on FN10 #28 and transfers profiles from the Programmable Controller memory to the MMC 188/40. Note that profiles contain a mode word, acceleration, deceleration and max speed parameters, but no position information. Requested positions are specified when motion commands are given.

FN10 #33 (Move Status to 0X Table) moves 64 bits of status information from the input register to discrete I/O points. It also passes power from FN10 #30 through its middle node.

FN10 #31 (Move Block From Table) copies a block of data to the MMC 188/40 each time its middle node is activated.


Network 1
40164 INIT 40001 FN10 32
40116 SPAR 40200 FN10 28
40148 SPRO 40033 FN10 30
40180 MSTT 40001 FN10 33
01022
01024
00001
40170 MBFT 40300 FN10 31
01023
P
00016
01021
40100 GPAR 40200 FN10 27
40132 GPRO 40033 FN10 29
50 T0.1 40196
00013
01024
00001
01024
Network 2
P
00016
00002
00003
40188 40001 AND 1
40189 40001 OR 1
00018
P
00032
00019
40188 40003 AND 1
40189 40003 OR 1
00034
P
00048
00035
00050
P
00064
00051
40188 40007 AND 1
40189 40007 OR 1
40188 40005 AND 1
40189 40005 OR 1
Network 3
00032
00048
00064
10 T0.1 40197
01020
01023
P
01020
40340 MBFT 40300 FN10 31

START-UP

CAUTION: Do not connect the drive outputs to the drives until the limits have been computed and MMC 188/40 properly initialized by the 984 Programmable Controller with these limits.

CAUTION: Great care must be taken to avoid accidents when starting the motion controller module for the first time. The most common accident is a runaway. This is where the motion controller tries to go to a position beyond the physical limits of the axis.

When the MMC 188/40 motion controller module is first turned on, default values are used for the initialization parameters. These values are there only to give the motion controller something to work with so the present position can be maintained. The parameters need to be changed to work optimally in your system. The 984 programmable controller is responsible for initializing the parameters when the MMC 188/40 is powered up by sending the correct initialization parameters to the module and issuing a 'P' command. The following procedure will help in finding the optimal values for the initialization parameters.

TIP: Use Delta's DCSMON setup program to help determine initial parameters and operation. Also, Delta's SSS/10 Servo System Simulator can provide a simple way to test your program before connecting the module to a real system.

First, disconnect the drive output from the MMC 188/40 (15 pin connector), and establish a means of moving the axis under manual control. Set up DCSMON to communicate with the module. Make sure the transducers are connected and powered up.

Initialization Parameters

If you want to get up and running fast in order to play around with the system, you can proceed to the following section - Moving the Axis. Continue with this section now or return to it later to select the SCALE, DIRECTION and OFFSET parameters.

DIRECTION Parameter

Decide whether the ACTUAL POSITION will increase or decrease as the axis extends. (Remember - extending is the direction that returns increasing TRANSDUCER COUNTS.) The DIRECTION parameter should be 0 if the ACTUAL POSITION will increase as the axis extends. This is the default value. The DIRECTION parameter should be set to -1 (0FFFFH) if the ACTUAL POSITION decreases when the axis extends. The Programmable Controller must now issue a Set Parameter ('P') command.

SCALE Parameter

The next step is to compute the SCALE factor that converts TRANSDUCER COUNTS to Position Units. The SCALE is computed by:

$$\text{SCALE} = \frac{32768}{\text{Position Unit} \times \text{Transducer Cal. Number} \times 27.75 \times \text{Number of Recirculations}}$$

Also P0 and P1 commands can be used in DCSMON to Calculate DIRECTION, SCALE and OFFSET.

OFFSET Parameter

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 LIMITS by the same amount. Use the following equation to adjust the current ACTUAL POSITION to be something else:

OFFSET = OLD OFFSET + DESIRED POSITION - ACTUAL POSITION

Both the EXTEND and RETRACT limits must be adjusted in the same way as the OFFSET, and another '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, manually move the axis close to the physical extend limit - be sure to leave some space for safety. (Remember: extending is the direction that returns increasing TRANSDUCER COUNTS.) 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 a 'P' command. (If you are using DCSMON, be sure to read the parameters into the Programmable Controller's memory.

Powering Up

Turn off the power to the MMC 188/40 and connect the drive output. Turn the power back on, and issue a 'P' command. Make several short slow moves by setting ACCELERATION, DECELERATION and MAX SPEED, entering the REQUESTED POSITION, and issuing a GO 'G' command.

NOTE: Here, DCSMON is extremely useful, and we recommend using it to troubleshoot and tune the system. Refer to the DCSMON manual to obtain plots of moves.

At this point the HALT MASK should still be set to zero and any errors on the axis will cause it to stop. If the axis indicator light flashes red after a 'G' command, check the STATUS word to determine what caused the error. Refer to the Tuning section below for suggestions on how to correct the errors.

Increase the speed and accelerations slowly while making long moves. Use DCSMON to make plots of the moves, and look for lead, lag and overshoot or oscillations. Eventually the moves will cause an error on the axis.

There are three errors to be concerned with at this time. If an over drive error occurs, there is not enough drive capacity to drive the axis at the requested speed. Should this occur, reduce the MAXIMUM SPEED. If a lag error occurs, the appropriate feed forward must be increased. If a lead error occurs, the appropriate feed forward must be decreased. If the lag or lead error cannot be eliminated, try increasing the FEED FORWARD ADVANCE parameter. In case it still doesn't solve the problem, the acceleration and deceleration ramps are too steep for the response of the system and the ramp distance should be made longer. After the problem causing the error has been fixed, keep moving the axis back and forth until you reach the desired speed. Should the system seem a little sloppy, try decreasing the MAXIMUM POSITION ERROR and adjusting one of the system gains (STATIC, EXTEND and RETRACT) until the position can be moved without getting an error. Remember, the parameters are not updated in the MMC 188/40 until the 'P' command is issued.

Jogging The Axis

Jogging the axis can be accomplished by setting the REQUESTED POSITION to the EXTEND LIMIT or RETRACT LIMIT and using the 'G' and 'H' commands repeatedly. This causes the motion controller to GO and HALT.

Using DCSMON

Use DELTA's DCSMON to help tune your system. See the DCSMON manual for information.

Tuning

NOTE: Use DCSMON to plot moves while tuning the system

NOTE: You will probably need to mask off several bits in the HALT MASK while tuning the system to avoid stopping every time an error occurs.

0 Don't use the INTEGRAL or DIFFERENTIAL GAINS or FEED FORWARD ADVANCE until you have finished with tips 1 through 10. Leave them at zero.

1 If the axis oscillates when it shouldn't be moving, reduce the STATIC GAIN.

2 If the axis oscillates when it is extending, reduce the EXTEND GAIN.

3 If the axis oscillates when it is retracting, reduce the RETRACT GAIN.

4 Gains should be increased until the axis starts to oscillate then follow tips 1 through 3.

5 If the axis gets a lead error and no lag error while extending, reduce the EXTEND FEED FORWARD.

6 If the axis gets a lag error and no lead error while extending, increase the EXTEND FEED FORWARD.

7 If the axis gets a lead error and no lag error while retracting, reduce the RETRACT FEED FORWARD.

8 If the axis gets a lag error and no lead error while retracting, increase the RETRACT FEED FORWARD.

9 If the axis gets both a lag error and a lead error, check for oscillations (tips 2 & 3). If it is not oscillating, increase the MAX ERROR until you get only one error (tips 5 - 8). Also check the FEED FORWARD ADVANCE (tip 14).

10 Once the axis moves smoothly use the "F" command (automatic tuning).

11 If the axis hunts around the set point, increase the HYSTERESIS slowly until the hunting stops or the axis starts to oscillate. If oscillations occur, reduce the HYSTERESIS.

12 The NULL UPDATE should not be used until the HYSTERESIS is adjusted. Try to keep the NULL UPDATE interval value higher than 1000. Systems with fast response times require larger NULL UPDATE intervals.

13 If the axis gets no lag or lead errors, reduce the MAX ERROR until errors start to occur then either tune the axis some more (tips 5 - 8), or increase the MAX ERROR a little.

14 The FEED FORWARD ADVANCE must be the same for all synchronized axis. Increasing the FEED FORWARD ADVANCE will decrease the chance of getting a lag error when ramping up or getting a lead error while ramping down. If the FEED FORWARD ADVANCE is too high, the axis will lead when ramping up and lag when ramping down.

15 The INTEGRAL GAIN is useful only when the feed forward terms can't be adjusted correctly. This may occur if the upper limit of the feed forward is reached or the axis has a non linear response to drive output. If the axis requires a different feed forward to move 10 inches a second than when moving 20 inches a second, the axis is non linear and the integrator can help compensate. In these cases a little integral gain will do the job. The better the feed forwards are adjusted, the lower integral gain.

Another problem occurs on moves of a long duration and when the feed forward can't be adjusted high enough. This usually only happens on systems that move less than 2 inches or 50 mm per second. If the integrator must be used, increase by units of 50. Sometimes the integrator gain will get to 500 or more.

16 The DIFFERENTIAL GAIN is of little use on systems that respond quickly. It is most useful on systems with slow response times that oscillate at 5 Hz or less. It adds a small boost while accelerating and helps stop while decelerating. Increase this number by ones. The DIFFERENTIAL GAIN should not get over 10; high values will cause oscillations.

The appropriate value for EXTEND FEED FORWARD can be found by making a move with the axis using a MAXIMUM SPEED of 10,000. The amount of output drive required to maintain MAXIMUM SPEED should be used as the Extend Feed Forward parameter. If, after the parameter has been set, lag errors occur, the feed forward is too small or the system response is too slow. If lead errors occur, the feed forward is too big or the system response is too slow. The 'F' command used after an extend move on the axis will automatically adjust the EXTEND FEED FORWARD parameter.

Lag errors can occur if the feed forward is too low or the feed forward advance is set too high. In the first case, the feed forward should be decreased. In the second case, the feed forward advance should be increased. Lead errors can occur if the feed forward is too high or the axis response is too slow to respond to the output while ramping down. In the first case, the feed forward should be decreased. In the second case, the feed forward advance should be increased.

Saving Parameters

After the system is set up and tuned using DCSMON, 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 being used need to be saved, and only the profiles used need to be saved.

984 PROGRAMMING HINTS

The Programmable Controller is responsible for storing the initialization parameters used by the MMC 188/40 and making sure the motion controller is properly initialized.

CAUTION: The Programmable Controller is also responsible for issuing an 'H' or Halt command when drive power is off. The Save Null and Restore Null commands can also be used - save the null before power is turned off, and restore the null when the power is turned back on.

The MMC 188/40 motion controller provides a STATUS word for each axis. Should an error bit be set in the STATUS word, the Programmable Controller is responsible for shutting down the axis drive power. It must have a watch dog timer that will shut down the drives if a time-out occurs.

Write to the MMC 188/40 only once per segment, otherwise the last data will overwrite the previous data.

Error Handling

The MMC 188/40 reports errors within two milliseconds after one is detected. This is done by setting a bit in the appropriate STATUS word and flashing the appropriate Front Panel LED. The Programmable Controller is responsible for checking errors using the STATUS bits. It is up to the Programmable Controller to determine what should be done if an error is detected.

The system must have the capability to shut down the axis drive power using a normally open output which is activated to close 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 be deactivated and the axes will stop. If an error occurs on the MMC 188/40, the contact can be deactivated, thus stopping the axes. Usually the Programmable Controller will not take so drastic a step until it has determined that all control is lost. An 'H' or HALT command for the axis with an error can take care of most error situations.

When two or more axes are moving in a coordinated manner and one axis starts moving slower than it should, it is probably best to issue a HALT command to all the axes and stop operation until the problem with the faulty axis has been solved.

It is also possible to use ESTOP and HALT on error detection by clearing the appropriate bits in the HALT MASK and the ESTOP MASK corresponding to the error bits in the STATUS word. See ESTOP MASK, HALT MASK and STATUS.

COMMON SYSTEM PROBLEMS

Problems and Solutions

Ladder program can not access parameters or operate module.

1) OURBUS not configured properly - Active light red.

Traffic cop the module as a B886 8 register bidirectional module with binary format.

2) Make sure the Programmable Controller is accessing the correct I/O registers.

Axis indicator constant red.

This is an indication the transducer is not responding to the module. Every two milliseconds the module interrogates the transducer's position. If a return response is not seen, the indicator will be a constant red. Check the following:

a. Polarity of the interrogation pulse (see hardware section).
b. External transducer power supply to make sure ±15 volts is present
c. Transducer power from the module to make sure ±15 volts is present
d. External transducer power supply to make sure +5 volts is present

If the +12 volt output is used, check its voltage also. If not present, make sure +15 volt is attached.

If the transducer power provided by the module has failed, check the 4.7 and 15 ohm resistors on the module. These resistors act as fuses and may be open if a problem with the transducer or external wiring occurred. Change the 4.7 ohm 5% resistor(s) (R13-R16) if a problem exists with the -15 volts. Change the 15 ohm 5% resistors (R9-R12) if a problem exists with the +15 volts.

During a move, the ACTUAL POSITION is erratic.

Electrical noise or a defective transducer is usually the cause of this problem. Monitor STATUS word bit 6 to determine if the module is detecting a transducer error. To reduce electrical noise check the following:

a. Make sure the transducer wiring is separated from all other wiring.
b. Add a termination resistor as close to the transducer (220 ohm for Temposonics) as possible.
c. Terminate the shield at module end, or transducer end or both.
d. Install a pair of Delta Computer Systems, Inc. AMP 10 RS-422 converter cards. If a TEMPO II transducer is used, the RPM option along with one AMP 10 card is recommended. Refer to the enclosed wiring diagram for more information.

During a move, the drive comes to a halt without any apparent reason.

When the module detects a transducer error an emergency stop is made. See ACTUAL POSITION is erratic in section above for more information. Other reasons for halting are:

Lag error, if enabled	Lead error, if enabled
Overdrive error, if enabled	Stopped, if enabled
Position Overflow, if enabled	Parameter Error, if enabled

Transducer counts field not indicating transducer location.

See AXIS indicator constant red section above.

Transducer counts field changes but output drive does not work

See drive comes to a halt without any apparent reason section above.

The System is unresponsive and hard to tune.

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

1) Is there a hose installed between the hydraulic valve and the cylinder? - the hose acts like an accumulator and the fluid goes to fill the hose instead of moving the cylinder.
2) Does the valve have overlap? - Overlap in hydraulic valves causes a significant deadband and slows their response. Some amplifier cards for proportional valves have deadband eliminator circuits - these make tuning easier.
3) If you have a servo motor and a ball screw, is there any backlash? - Backlash produces a deadband when the axis changes direction, and the controller will tend to oscillate in the deadband.

Field Module Repair

The following list of common problems and solutions is provided for those interested in repairing their module. Any repairs made other than replacing the transducer protection resistors will void the warranty.

1) If the transducer count is not being updated, check 4.7 ohm 5% resistor(s) (R62-R65) if a problem exists with the -15 volts. Check the 15 ohm 5% resistors (R53-R56) if a problem exists with the +15 volts.
2). If the interrogation pulse fails, change pulse drivers U18 and U19 (SN75158P)

WARRANTY

The MMC 188/40 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 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 MMC 188/40 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 Returns

When returning the MMC 188/40 for repair, please contact Delta prior to shipment for an RMA number. Returned modules should be packaged in static protection material and the RMA number clearly marked on the outside of package. Please include a short note explaining the problem. Send module to:

Delta Computer Systems, Inc. **Phone: 360-254-8688**
11719 NE 95th St., Suite D **Fax: 360-254-5435**
Vancouver, Washington 98682-2444

Return For Refund

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 items can be returned for re-stocking.

GLOSSARY OF TERMS

Actual Position

The scaled measured axis position expressed in Position Units.

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.

Dither

A small square wave added to the output drive to overcome static friction.

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.

Overlap

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

P.I.D.

Control industry term used to indicate 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 by using SCALE and COUNTS.

Recirculation

Is the number of times a magnetostrictive transducer is interrogated for a measurement. More recirculations provide higher resolution at a cost of increased measurement time and reduced sensor length.

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 be used as a means of 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 position to which the motion controller was told to move.

Target

A calculated position at which an axis is supposed to be at any given time.

Transducer

A general term that refers to a device that converts a given physical distance to an electrical signal.

Transducer Counts

See Counts.